Breast Cancer Classification using Machine Learning

Nur Sultan VASI and Teoman ÜNAL

Abstract—Machine learning (ML) methods are of great importance when applied interdisciplinarily. As in many fields, ML methods also provide cost and time savings in medical applications. In this study, we investigated the effectiveness of various machine learning techniques for breast cancer classification. We attempted to use various machine learning methods such as K-Nearest Neighbors (KNN), Naive Bayes, Decision Trees, and Support Vector Machines (SVM) both independently and in conjunction with dimensionality reduction techniques such as Principal Component Analysis (PCA) and Neighborhood Component Analysis (NCA). Our analysis includes comparing the results obtained when all features are used as input, and examining the impact of PCA and NCA on classification performance.

When several ML models were analyzed, KNN achieved a best training score of 0.99, a test score of 0.99 and a train score of 1.0. Other machine learning methods also demonstrated successful performance.

Index Terms— Outlier detection, standardization, Machine-learning methods, principal component analysis, neighborhood components Analysis, breast cancer classification.

I. INTRODUCTION

Breast cancer remains one of the most prevalent and concerning health issues globally, affecting millions of women every year. Early detection and accurate classification of breast cancer are paramount for effective treatment and patient outcomes. In recent years, machine learning techniques have emerged as powerful tools for assisting in the diagnosis and classification of breast cancer based on various clinical and imaging data.

The features used in this study are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. Additionally, this dataset contains information about whether the data in breast cancer classification is benign or malignant.

In this study, we aimed to evaluate the effectiveness of different machine learning models in classifying breast cancer tumors. Specifically, we investigated the performance of four popular machine learning algorithms: k-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Decision Tree, and Naive Bayes. Additionally, dimensionality reduction techniques such as Principal Component Analysis (PCA) and Neighborhood Components Analysis (NCA) were applied to measure whether success rates would increase.

Overall, this study contributes to the field of machine learning applications in healthcare, particularly in research focused on breast cancer classification. The findings underscore the importance of using machine learning techniques to accurately classify breast cancer types and initiate timely treatment. This has the potential to improve patient outcomes and increase survival rates.

II. METHOD

A. System Overview

In this study, we trained the dataset using the methods and parameters we employed in the dataset. As another option, in the second step, after PCA and NCA analysis, we trained the dataset. In the second step, 70% of the dataset is selected for training, and ML methods are constructed with these data. In the third step,

the remaining 30% of the data is used to evaluate ML methods, and in the final step, different evaluation methods are compared. This is demonstrated for single-layer Cross-Validation (CV) for one dataset. 10-fold CV is computed for each ML method for both datasets.

B. Datasets

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.
n the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].
Attribute Information:
1) ID number
2) Diagnosis (M = malignant, B = benign)
3-32)
Ten real-valued features are computed for each cell nucleus:
a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation")
The mean, standard error and "worst" or largest (mean of the three
largest values) of these features were computed for each image,
resulting in 30 features. For instance, field 3 is Mean Radius, field
13 is Radius SE, field 23 is Worst Radius.

All feature values are recorded with four significant digits, and there are no missing attribute values. Class distribution: 357 benign (non-cancerous), 212 malignant (cancerous). This represents the distribution of diagnosed cells in the dataset.

C. Data Preprocessing, Principal Component Analysis and Neighborhood Components Analysis

Data preprocessing is one of the most critical stages in ML studies. For instance, feature selection helps us choose features with high discriminatory power. In this study, correlation was examined. When examining correlation, a threshold value of >= 0.75 was applied for feature selection. Features with this threshold value or higher may not yield meaningful results for classification on their own. Features observed to have a threshold value equal to or above it are radius_worst, perimeter_worst, concave_points_mean, and concave_points_worst.

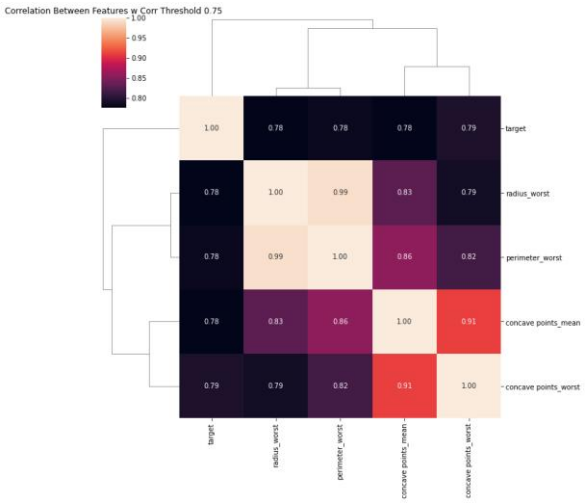

Fig. 1. Correlation matrix when threshold = 0.75.

We plotted a box plot to compare the distributions of features among classes in the dataset, as seen in Figure 2. In this graph, we observed that the feature values were widely spread across different ranges. Therefore, standardization was applied as shown in Figure 3. However, before applying this, we decided to perform outlier detection. Upon examining the graphs between features, we observed positive skewness. Hence, we decided to conduct outlier detection using the Local

Outlier Factor (LOF) method, as shown in Figure 4 where the outlier-selected feature is displayed. The outlier data found by this method was removed from the dataset.

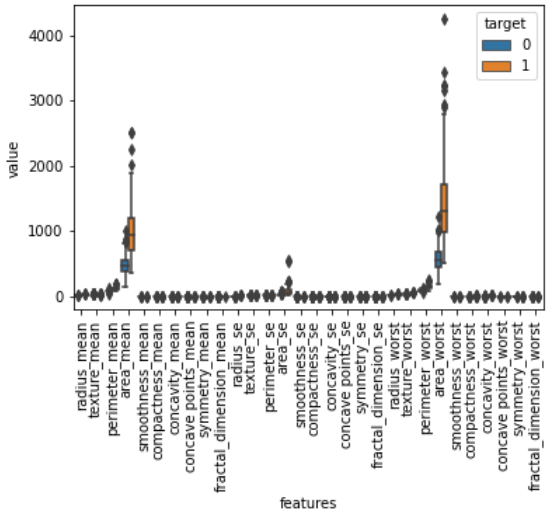

Fig. 2. A boxplot to compare the distributions of features across classes in the dataset.

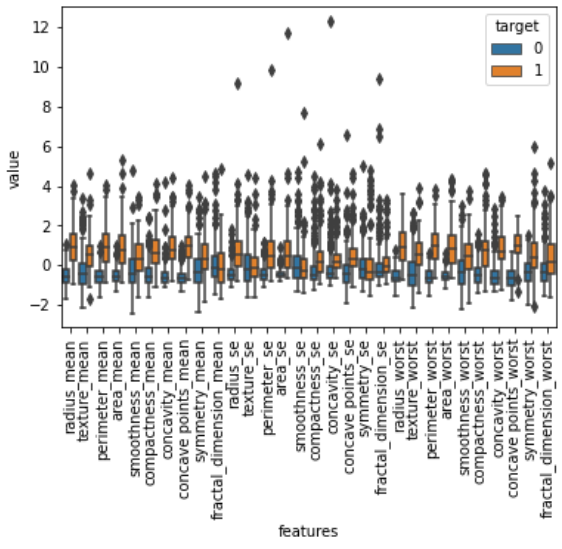

Fig. 3. A box plot to compare the distribution of features in the data set between classes after standardization.

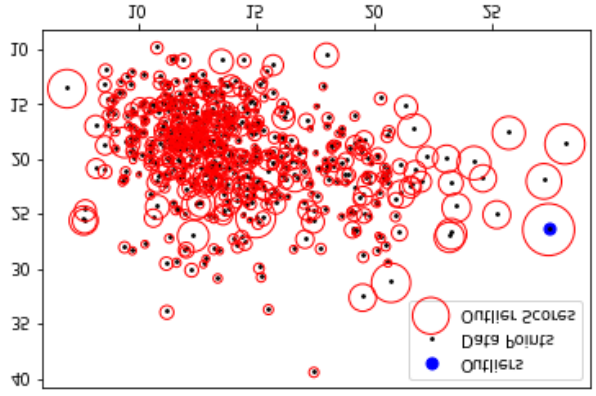

Fig. 4. Viewing features after outlier detection.

Principal Component Analysis (PCA) and Neighborhood Components Analysis (NCA) were applied to reduce the dimensionality of the feature space. PCA selects components to maximize the variance of the data, working under the assumption of linear structure and unimodal Gaussian distribution. NCA projects the data into a low-dimensional space to optimize classification accuracy, aiming to preserve neighborhood structure and utilizing gradient-based optimization techniques. In both the NCA and PCA approaches, the features were reduced to only 2 dimensions.

D. Usage of Machine-Learning Methods

Various types of machine learning methods were applied. These ML methods are k-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Decision Tree, and Naive Bayes algorithms.

For kNN, the value of k was determined by testing only odd numbers from 1 to 31. The kNN parameters are as follows: 'n_neighbors' includes odd numbers from 1 to 31 (e.g., 1, 3, 5, ... 29), 'weights' can be 'uniform' or 'distance', and 'metric' can be 'euclidean' or 'manhattan'. The SVM parameters are: 'C' values of 0.1, 1, 10, and 100; 'kernel' options of 'linear', 'poly', 'rbf', and 'sigmoid'; and 'gamma' options of 'scale' and 'auto'. For Decision Tree, the parameters are: 'criterion' can be 'gini' or 'entropy'; 'max_depth' values of 10, 20, 30, 40, 50, and None; 'min_samples_split' values of 2, 5, and 10; and 'min_samples_leaf' values of 1, 2, and 4. Additionally, Naive Bayes algorithms were applied.

E. Evaluation

The implementation of a Cross-Validation (CV) scheme has a significant impact on evaluating an ML method. In this method, the dataset is divided into 10 equal parts, and each part is sequentially used as the test set while the remaining 9 parts are used as the training set. This way, the model's generalization ability is evaluated more reliably, and problems like overfitting are minimized. Ten-fold cross-validation is completed with each part serving

as the test set, and the average performance of the 10-fold cross-validation is calculated.

III. RESULTS

In this study, several runs were conducted. The average performance of 10-fold cross-validation (CV) was calculated for each ML method by applying each evaluation. Here, we will examine the results in two sections.

A. Comparison of Results Using All Features as Input

In this analysis, four different types of standard ML methods were tested on the dataset. All features were provided as input for each ML method. 10-fold CV was applied, and the results contain the average performance of 10-fold CV when the parameters of ML methods were adjusted as explained in the Methods section. Table I presents the best training score, test score, train score, mean CV score, the best parameters among the results, and confusion matrices, respectively, for all methods. Figure 5 represents a histogram graph of these results. When examining the results from here, Decision Tree has the best training score, but it did not perform as well as other algorithms in terms of other metrics.

TABLE I: Training and testing results of features on the data set with different methods and parameters

Model	Best Training Score	Test Score	Train Score	Mean CV Score	Parameters (Best)	CM Test	CM Train
KNN	0.978	0.959	0.977	0.965	{'metric': 'manhattan', 'n_neighbors': 5, 'weights': 'uniform'}	[[106, 3], [4, 58]]	[[247, 1], [8, 141]]
SVC	0.975	0.977	0.992	0.970	{'C': 1, 'gamma': 'scale', 'kernel': 'linear'}	[[108, 1], [3, 59]]	[[248, 0], [3, 146]]
Decision Tree	0.987	0.942	0.987	0.938	{'criterion': 'entropy', 'max_depth': 30, 'min_samples_leaf': 2, 'min_samples_split': 5}	[[105, 4], [6, 56]]	[[248, 0], [5, 144]]
Naive Bayes	0.940	0.918	0.940	0.933	-	[[102, 7], [7, 55]]	[[238, 10], [14, 135]]

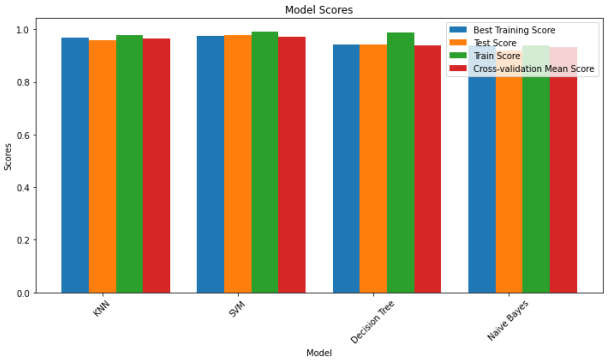

Fig. 5. Histogram chart of the best training and testing results of features in different methods

B. Comparison of PCA and NCA

PCA and NCA are commonly used when there are not enough samples to represent the feature space. Here, we applied PCA and NCA approaches to reduce thirty-two dimensions to two dimensions. The results cover the average of 10-fold cross-validation. Table II displays the results for each method and its parameters trained after PCA. Figure 6, on the other hand, represents a histogram graph of these results. When examining the results, SVM stands out with the highest best training score, but other algorithms also show very good performance. After PCA, the best training score has slightly decreased.

TABLE II: Training and testing results of the features on the data set with different methods and parameters after applying PCA

Model	Best Train Score	Test Score	Train Score	Best Parameters	CV Best Train Score	Mean CV Score	CV Best Parameters
KNN	0.9369	0.924	0.9369	{'n_neighbors': 9, 'weights': 'uniform'}	0.9369	0.9277	{'n_neighbors': 9, 'weights': 'uniform'}
SVM	0.9496	0.947	0.9496	{'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}	0.9496	0.9278	{'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
Decision Tree	0.9344	0.930	0.9344	{'max_depth': 3}	0.9294	0.9125	{'max_depth': 3}
Naive Bayes	0.9192	0.895	0.9192	No parameters	0.9192	0.9192	-

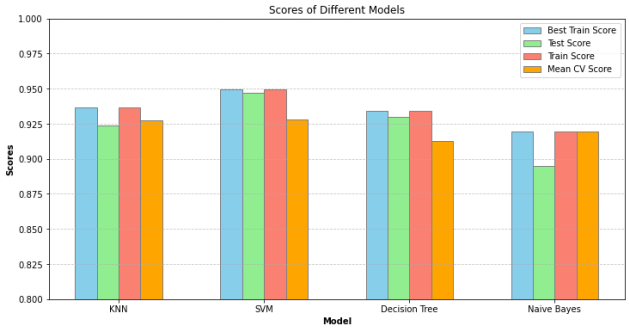

Fig. 6. Histogram chart of the best training and testing results of features in different methods after applying PCA

The data was transformed into a lower-dimensional space using the Neighborhood Components Analysis (NCA) method, followed by training the K-Nearest Neighbors (KNN) model. The best results obtained from the trained dataset were observed here.

This model has achieved high accuracy on both the training and test datasets. The best score obtained on the training set is 98.7%, indicating a strong fit to the training data, while the accuracy score for the test set is approximately 99.4%. In the confusion matrices, both datasets show high values for true positives and true negatives, with zero values for false positives and false negatives, indicating effective classification for both classes. Precision, sensitivity, and specificity values are measured as excellent for both datasets. The model's F1 score is high, and the false positive rate is zero, indicating accurate prediction of the negative class. With a high Matthews correlation coefficient, the model demonstrates balanced performance. All these metrics indicate that the model performs well on both training and test datasets, making it a reliable classification model overall.

TABLE III: Best parameter results in knn method of features after applying NCA

	Test Set	Train Set
Accuracy Score	0.994	1.0
Best Training Score	-	0.987
Confusion Matrix (CM)	[[108, 1], [0, 62]]	[[248, 0], [0, 149]]
True Positive (TP)	108	248
False Negative (FN)	1	0
False Positive (FP)	0	0
True Negative (TN)	62	149
Precision	1.0	1.0
Sensitivity	0.9917	1.0
Specificity	1.0	1.0
F1 Score	0.9958	1.0

The False Negative value in the test set is 1. This indicates that the model misclassified 1 positive example as negative in the test dataset. This situation can be observed in Figure 4 in detail. In other words, it classified a benign cancer as malignant. This suggests that the model may miss some positive examples and there are areas for improvement.

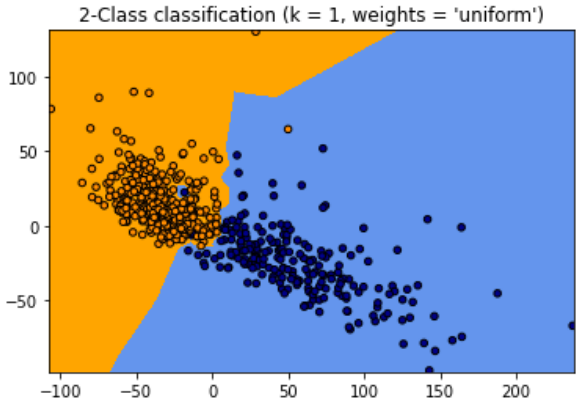

Fig. 7. Best parameter results in knn method of features after applying NCA

IV. CONCLUSION AND FUTURE WORK

In this study, the dataset was analyzed for breast cancer classification by applying various ML methods. We achieved quite positive results through the application and comparison of different techniques. Each technique used provided useful outcomes. Traditional classification approaches were employed to make decisions about breast cancer classification based on the given features. Performance improvement was also achieved through outlier detection and standardization methods before using these approaches.

In future studies of these works, we believe that by adding different datasets and methods, performance improvement or obtaining a model with better performance can be achieved.

REFERENCES

1. https://www.kaggle.com/code/mmenendezg/breast-cancer-classification-random-forest.
2. https://www.kaggle.com/code/janaasharf/machine-learning-breast-cancerr