BIRZEIT UNIVERSITY

Faculty of Engineering & Technology Electrical & Computer Engineering Department

Machine Learning and Data Science - ENCS5341

Assignment 3

KNN, Logistic Regression, SVM, Kernel Methods, and Ensemble Methods (Boosting and Bagging)

Prepared by: QOSSAY RIDA 1211553

MAI BEITNOBA 1210260

Instructor: Ismail Khater

Date: 12/26/202

Table of Contents

Table of Figures

List of Tables

side note: I asked Dr. Yazan about the page limit, and he confirmed that it doesn't include the cover page and tables.

Introduction

This report explores the application of several machine learning algorithms to classify breast cancer as either malignant or benign using the Breast Cancer dataset from scikit-learn. The objective is to evaluate these algorithms in terms of their predictive accuracy, and ability to generalize. By analyzing their performance across a variety of configurations, we aim to identify the most suitable method for this classification task.

Data Set Description

The Breast Cancer dataset from scikit-learn contains data derived from digitized images of breast tissue samples. The dataset includes 569 instances, each represented by 30 numeric features and a target variable. The target variable, labeled as "target," indicates whether the tumor is malignant (0) or benign (1). The features summarize specific measurements of the cell nuclei, such as the mean, standard error, and worst (largest) values of radius, symmetry, and fractal dimension. The dataset is complete, with no missing values, making it ready for analysis without additional preprocessing.

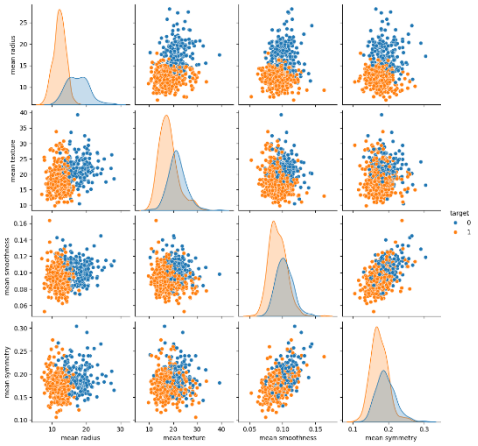

Figure 1:Data Visualization

Figure 1 shows how selected features (mean radius, mean texture, mean smoothness, and mean symmetry) are distributed and how they relate to each other. The diagonal plots display density curves for each feature, highlighting differences in their distributions between malignant and benign cases. The scatter plots in the off-diagonal cells reveal the relationships between pairs of features and how well they separate the two classes.

Algorithms Explored

K-Nearest Neighbors (KNN)

A non-parametric algorithm that classifies a data point based on the majority class among its nearest neighbors. The choice of distance metric and the number of neighbors (K) significantly affect its performance such that A smaller value of K can lead to overfitting, while a larger value smoothens decision boundaries but might introduce underfitting.

Logistic Regression

A statistical model that uses a logistic function to model the probability of a binary dependent variable. It assumes a linear relationship between the independent variables and the log-odds of the dependent variable. Regularization techniques like L1 (Lasso) and L2 (Ridge) help prevent overfitting, especially when dealing with high-dimensional datasets.

Support Vector Machines (SVM)

A supervised learning model that finds the optimal hyperplane to separate data points of different classes by maximizing the margin. The use of kernels, such as linear, polynomial, and radial basis function (RBF), enables SVM to handle both linear and non-linear data. SVM is particularly effective in high-dimensional spaces.

Ensemble Methods

- **Boosting:** An ensemble method that combines multiple weak learners to create a strong learner. AdaBoost, a common boosting algorithm, iteratively adjusts the weights of misclassified samples to focus the learning process on challenging cases. Boosting is effective in reducing bias and has been successfully applied to problems like fraud detection and sentiment analysis.
- **Bagging:** Another ensemble method that builds multiple models on different subsets of the data and combines their outputs to improve stability and accuracy. Random Forest, a popular bagging-based algorithm, grows multiple decision trees and averages their predictions to reduce variance and overfitting. Bagging methods are well-suited for tasks like anomaly detection and feature selection.

Methodology

The dataset was divided into training and testing sets using an 80-20 split to ensure unbiased evaluation of the models. The models were trained on the training data and evaluated on the testing data.

K-Nearest Neighbors (KNN)

For the KNN classifier, experiments were conducted with three distance metrics: Euclidean, Manhattan, and Cosine. Cross-validation was used to identify the optimal number of neighbors (k) for each metric. Values of k ranging from 1 to 40 were tested, and the k that yielded the highest cross-validation accuracy was selected. The final KNN models were trained using the optimal k and corresponding distance metric.

Logistic Regression

Logistic regression models were trained with two types of regularization: L1 (Lasso) and L2 (Ridge). The liblinear solver was used for L1 regularization, while saga was employed for L2, ensuring compatibility and efficiency. Each model was trained on the training data, and predictions were generated for the test set.

Support Vector Machines (SVM)

The SVM classifier was tested with three kernel functions: linear, polynomial (degree 3), and radial basis function (RBF). Each kernel was applied to train the SVM models on the training data. Since SVMs do not directly output probabilities, the (probability=True) parameter was used to enable probability estimation. Test set predictions and class probabilities were then used for performance evaluation.

Ensemble Methods

Two ensemble methods, AdaBoost and Random Forest, were implemented to evaluate their performance. AdaBoost was trained with 100 estimators, focusing on sequentially improving weak learners. Random Forest, a bagging technique, was also trained with 100 estimators to leverage ensemble diversity and reduce overfitting. Both models were trained on the training data, and predictions were made on the test set.

The performance of all models was assessed using the following evaluation metrics: **accuracy**, **precision**, **recall**, **F1-score**, and **ROC-AUC**.

Results, Analysis, and Comparison

K-Nearest Neighbors (KNN)

Table 1:KNN Results

	Best K	Cross-Validation Accuracy	Test Accuracy	Precision	Recall	F1-Score	ROC-AUC
Euclidean	9	0.9275	0.9561	0.9459	0.9859	0.9655	0.9956
Manhattan	6	0.9407	0.9561	0.9459	0.9859	0.9655	0.9964
Cosine	11	0.9209	0.9474	0.9333	0.9859	0.9589	0.9843

Using the Euclidean distance, the best value of k was 9. The cross-validation accuracy was 0.9275, and the test accuracy reached 0.9561. The model achieved a precision of 0.9459, a recall of 0.9859, and an F1-score of 0.9655. The ROC curve for Euclidean distance showed a steep rise near the origin, reflecting a strong true positive rate at low false positive rates, with an excellent ROC-AUC score of 0.9956. These results indicate that Euclidean distance is highly effective for this dataset, providing accurate predictions with minimal overlap between positive and negative classes.

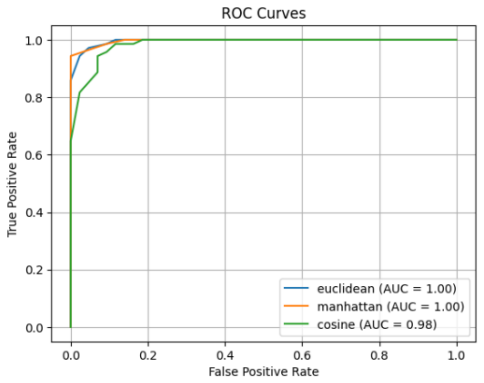

Figure 2:KNN ROC Curves

For the Manhattan distance, the optimal k was 6. The cross-validation accuracy was slightly higher at 0.9407, and the test accuracy remained consistent at 0.9561. The precision, recall, and F1-score matched those of Euclidean, while the ROC-AUC score was marginally higher at 0.9964. The ROC curve exhibited similar characteristics to Euclidean, reinforcing the Manhattan metric's robustness. This slight edge in AUC suggests that Manhattan may better handle subtle patterns in the dataset due to its sensitivity to feature scales.

The Cosine distance, with an optimal k of 11, demonstrated slightly lower performance. The cross-validation accuracy was 0.9209, and the test accuracy dropped to 0.9474. Precision was 0.9333, recall was 0.9859, and the F1-score was 0.9589. The ROC-AUC score of 0.9843 was still strong but lower than those of the other metrics. The ROC curve for Cosine distance showed

slightly less precision in distinguishing between classes, particularly at intermediate thresholds. This result suggests that Cosine similarity, which measures directional similarity, may be less suited to this dataset compared to distance-based metrics.

The choice of distance metric significantly impacts the classification performance of KNN. Euclidean and Manhattan distances achieved comparable results, with Manhattan showing a slight edge in cross-validation accuracy and ROC-AUC. Cosine distance, while still effective, lagged slightly behind the other two metrics in terms of accuracy and F1-score. These findings highlight the importance of selecting an appropriate distance metric that aligns with the underlying structure of the data.

The best value of k also varied across metrics, with k=9 for Euclidean, k=6 for Manhattan, and k=11 for Cosine. The choice of k represents a critical balance between overfitting and underfitting. Smaller values of k can make the model overly sensitive to noise in the training data, while larger values may smooth out important patterns and lead to underfitting. For this dataset, k=6 with Manhattan distance provided the best overall performance, making it the most suitable configuration.

Logistic Regression

Table 2:Logistic Regression results

	Accuracy	Precision	Recall	F1-Score	ROC-AUC
L1	0.9561	0.9459	0.9859	0.9655	0.9977
L2	0.9737	0.9595	1.000	0.9793	0.9951

Logistic Regression was evaluated with two regularization techniques, L1 (Lasso) and L2 (Ridge). Both regularizations achieved strong performance across all metrics, showcasing the model's robustness.

For L1 regularization, the accuracy was 0.9561, precision was 0.9459, recall was 0.9859, and the F1-score reached 0.9655. The ROC-AUC score was exceptionally high at 0.9977, indicating excellent discriminatory ability between positive and negative classes. The steep ROC curve with minimal deviation highlights the effectiveness of the L1-regularized model in maintaining predictive precision across thresholds.

For L2 regularization, the performance was even stronger, with an accuracy of 0.9737, precision of 0.9595, recall of 1.0000, and an F1-score of 0.9793. The ROC-AUC score was slightly lower at 0.9951, but still reflected near-perfect classification capability. The higher recall and F1-score of L2 regularization suggest better handling of true

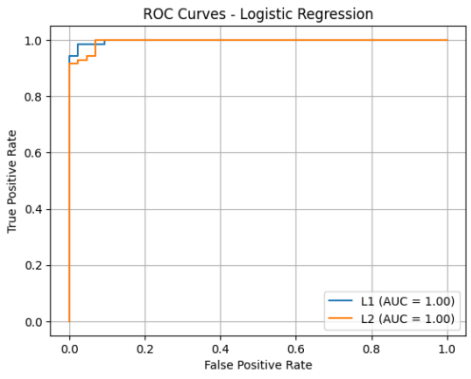

Figure 3:Logistic Regression ROC Curve

positives and overall predictive balance. The ROC curve for L2 closely mirrors that of L1, indicating consistent performance across both regularizations.

When comparing Logistic Regression and KNN, several key differences emerge. Logistic Regression demonstrated higher overall accuracy, particularly with L2 regularization, where it achieved 0.9737 compared to the KNN's best accuracy of 0.9561. Precision and recall metrics also favored Logistic Regression, with L2 achieving a recall of 1.0000, compared to 0.9859 for KNN. This suggests that Logistic Regression is more effective at identifying all true positive cases while maintaining fewer false negatives.

In terms of the ROC-AUC metric, both methods performed exceptionally well, with Logistic Regression (L1: 0.9977, L2: 0.9951) slightly surpassing KNN (Euclidean: 0.9956, Manhattan: 0.9964). The smoother, steep ROC curves for Logistic Regression emphasize its strong ability to separate classes, particularly when paired with L1 regularization.

Another distinction lies in model simplicity and computational efficiency. Logistic Regression offers a straightforward interpretation of coefficients and faster execution, making it a more efficient choice for high-dimensional datasets. KNN, while effective, can become computationally expensive as the dataset size increases due to the need to calculate distances for all data points during predictions.

Support Vector Machines (SVM)

Table 3:SVM Results

	Accuracy	Precision	Recall	F1-Score	ROC-AUC
LINEAR	0.9561	0.9459	0.9859	0.9655	0.9957
POLY	0.9474	0.9221	1.0000	0.9595	0.9915
RBF	0.9474	0.9221	1.0000	0.9595	0.9934

The Linear kernel achieved an accuracy of 0.9561, a precision of 0.9459, a recall of 0.9859, and an F1-score of 0.9655. The ROC-AUC score was 0.9957, and its ROC curve exhibited a steep rise near the origin, indicating a strong true positive rate with minimal false positives. This near-perfect AUC reflects the kernel's ability to draw effective linear decision boundaries, aligning well with the dataset's characteristics.

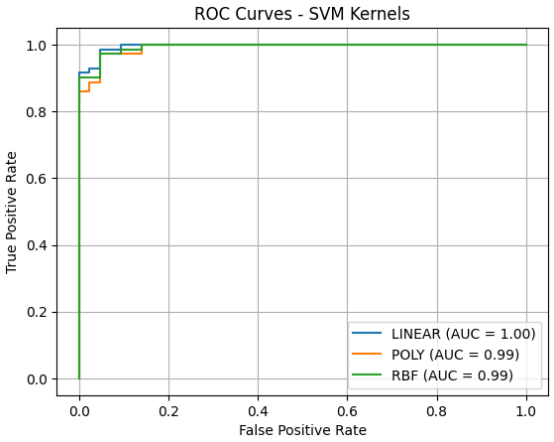

Figure 4:SVM ROC Curve

The Polynomial kernel achieved slightly lower accuracy at 0.9474, paired with a recall of 1.0000 and a precision of 0.9221, resulting in an F1-score of 0.9595. Its ROC-AUC

score of 0.9915, while still excellent, was marginally lower than the Linear kernel. The ROC curve for the Polynomial kernel showed strong performance but deviated slightly at intermediate thresholds, suggesting that while it captures all positive cases, it struggles with precision, possibly due to overfitting on non-linear patterns.

The RBF kernel achieved the same accuracy, precision, recall, and F1-score as the Polynomial kernel but with a slightly higher ROC-AUC score of 0.9934. Its ROC curve was comparable to the Polynomial kernel but displayed better separability at certain thresholds. This indicates that the RBF kernel may generalize slightly better than the Polynomial kernel, particularly for complex, non-linear relationships in the data.

The choice of the kernel significantly impacts the model's performance. The Linear kernel emerged as the most effective, achieving the highest accuracy and ROC-AUC while maintaining a strong balance between precision and recall. This simplicity in performance highlights that the dataset likely has a linear decision boundary, making complex kernels unnecessary.

Both the Polynomial and RBF kernels achieved similar performance but with slight trade-offs. While their perfect recall indicates they captured all positive cases, their slightly lower precision and accuracy suggest susceptibility to overfitting or difficulty generalizing across the entire dataset. The RBF kernel's slightly higher ROC-AUC compared to the Polynomial kernel reflects its ability to better adapt to the dataset's characteristics.

Ensemble Methods

Table 4:Ensemble Methods Results

	Accuracy	Precision	Recall	F1-Score	ROC-AUC
AdaBoost	0.9737	0.9722	0.9859	0.9790	0.9954
Random Forest	0.9649	0.9589	0.9859	0.9722	0.9953

The performance of ensemble methods, specifically AdaBoost and Random Forest, was evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics. Both methods demonstrated exceptional classification capabilities, with slight differences in their results. AdaBoost achieved an accuracy of 0.9737, surpassing Random Forest's accuracy of 0.9649. It also showed higher precision at 0.9722 and a slightly better F1-score of 0.9790 compared to Random Forest's 0.9722. The recall for both methods was identical at 0.9859, demonstrating their ability to correctly identify positive cases. The ROC-AUC

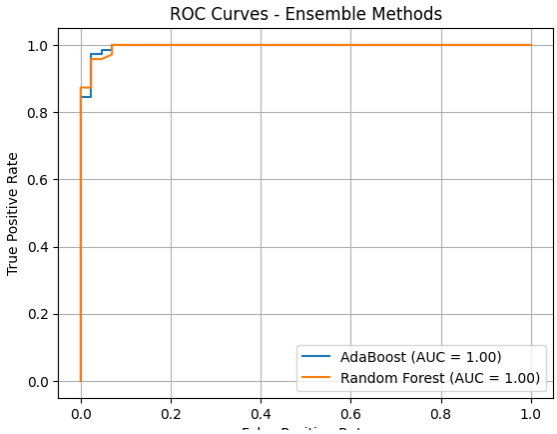

Figure 5:Ensemble Methods ROC Curve

scores for both AdaBoost and Random Forest were nearly perfect, at 0.9954 and 0.9953, respectively. The ROC curves for both models showed steep rises near the origin, indicative of high true positive rates at low false positive rates. The marginally better performance of AdaBoost can be attributed to its iterative learning process, which focuses on correcting misclassified samples, allowing it to adapt more effectively to complex patterns.

When comparing ensemble methods like AdaBoost and Random Forest to individual models such as KNN, Logistic Regression, and SVM, the ensembles consistently came out on top across various evaluation metrics. Among the individual models, Logistic Regression with L2 regularization came closest to matching AdaBoost's performance. It achieved an impressive accuracy of 0.9737, an F1-score of 0.9793, and a perfect recall of 1.0000. Although its precision (0.9595) and ROC-AUC score (0.9951) were slightly below AdaBoost's, it still delivered excellent results.

SVM with a linear kernel also performed well, with an accuracy of 0.9561, an F1-score of 0.9655, and a ROC-AUC of 0.9957. However, its performance varied with other kernels, where accuracy and F1-scores dipped slightly. KNN, tested with Manhattan, Euclidean, and Cosine distance metrics, showed decent results but fell short of the ensemble methods. The Manhattan metric performed best, with a test accuracy of 0.9561 and a ROC-AUC of 0.9964, but its F1-score (0.9655) and precision (0.9459) couldn't match AdaBoost or Logistic Regression.

The strength of ensemble methods lies in their ability to combine predictions effectively. Random Forest, using bagging, reduced overfitting, while AdaBoost refined predictions by focusing on harder-to-classify samples. This adaptability allowed them to handle overlapping classes and subtle patterns more effectively than individual models. Overall, AdaBoost emerged as the best performer, offering a balance of high accuracy, precision, and adaptability.

Conclusion

This assignment focused on applying machine learning algorithms to classify breast cancer as malignant or benign using the Breast Cancer dataset from scikit-learn. Among the models tested, AdaBoost emerged as the best performer due to its adaptability and ability to handle challenging samples effectively. Logistic Regression with L2 regularization also stood out as a strong competitor, delivering reliable and consistent results.

Ensemble methods, particularly AdaBoost and Random Forest, proved superior to individual models like KNN, Logistic Regression, and SVM. SVM with a linear kernel delivered strong results, but its performance varied with different kernels, highlighting its sensitivity to kernel selection. KNN with the Manhattan distance metric and Logistic Regression performed well individually but lacked the flexibility and robustness of ensemble approaches.

This underscores the importance of selecting the right model and tuning hyperparameters to balance accuracy and computational efficiency. Ensemble methods demonstrated clear advantages, making them highly suitable for complex classification tasks.