

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

Machine Learning and Data Science ENCS5341

**Assignment 3**

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

**Prepared by:** Adam Nassan, 1202076

**Instructor:** Dr. Ismail Khater

**Section:** 3

**Date:** 28-12-2024

# Introduction

This report presents the implementation and evaluation of various machine learning algorithms, including K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machines (SVM), and Ensemble Methods (Boosting and Bagging). The objective is to understand these algorithms, experiment with different configurations, and analyze their performance using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. The dataset used is the Breast Cancer dataset from Scikit-learn, which provides real-world data for binary classification tasks.

**Methods**

**K-Nearest Neighbors (KNN)**

The K-Nearest Neighbors (KNN) algorithm is a simple, yet powerful, supervised machine learning algorithm used for both classification and regression tasks. It is based on the principle that similar data points are likely to have similar outcomes. The algorithm operates by finding the 'k' closest data points (neighbors) to a given query point and making predictions based on the majority class (for classification) or the average value (for regression) of these neighbors.

**Key Characteristics of KNN:**

* **Instance-Based Learning:** KNN is a type of instance-based learning, where the model does not learn an explicit mapping function from the training data. Instead, it memorizes the training dataset and makes predictions based on the stored instances.
* **Distance Metrics:** The performance of KNN heavily depends on the choice of distance metric used to identify the nearest neighbors. Common distance metrics include Euclidean, Manhattan, and Cosine distance.
* **Parameter 'k':** The number of neighbors 'k' is a crucial hyperparameter that influences the model's performance. A small 'k' can lead to a model that is sensitive to noise, while a large 'k' can smooth out the decision boundary too much, potentially leading to underfitting.
* **Non-Parametric:** KNN is a non-parametric algorithm, meaning it makes no assumptions about the underlying data distribution. This makes it versatile and applicable to a wide range of problems.

**Logistic Regression**

Logistic Regression is a widely used supervised learning algorithm primarily employed for binary classification tasks, although it can be extended to multiclass classification problems. Despite its name, Logistic Regression is a classification algorithm, not a regression algorithm. It models the probability that a given input belongs to a particular class.

**Key Characteristics of Logistic Regression:**

* **Linear Model:** Logistic Regression is a linear model that uses a logistic function (also known as the sigmoid function) to map predicted values to probabilities.
* **Probability Output:** The output of Logistic Regression is a probability score between 0 and 1, which can be thresholded to make binary classification decisions.
* **Decision Boundary:** The decision boundary in Logistic Regression is a linear hyperplane that separates the classes. For binary classification, this boundary is a straight line (or plane in higher dimensions).

**Logistic Function (Sigmoid Function):**

The logistic function is defined as: [ \sigma(z) = \frac{1}{1 + e^{-z}} ] where ( z ) is the linear combination of input features and their corresponding coefficients.

**Regularization Techniques:**

Logistic Regression can be regularized to prevent overfitting and improve generalization. The two most common regularization techniques are:

* **L1 Regularization (Lasso):** Adds the absolute value of the coefficients as a penalty term to the loss function. It can lead to sparse models where some coefficients are exactly zero.
* **L2 Regularization (Ridge):** Adds the squared value of the coefficients as a penalty term to the loss function. It tends to shrink the coefficients but does not set them to zero.

**Support Vector Machines (SVM)**

Support Vector Machines (SVM) are a powerful set of supervised learning algorithms used for both classification and regression tasks. SVMs are particularly well-suited for classification of complex datasets with clear margins of separation. The core idea behind SVM is to find the optimal hyperplane that best separates the data points of different classes.

**Key Characteristics of SVM:**

* **Hyperplane:** In SVM, a hyperplane is a decision boundary that separates data points of different classes. The optimal hyperplane is the one that maximizes the margin between the closest points of the classes, known as support vectors.
* **Support Vectors:** These are the data points that are closest to the hyperplane and influence its position and orientation. The algorithm uses these points to maximize the margin.
* **Kernels:** SVM can efficiently perform a non-linear classification using the kernel trick, implicitly mapping the input features into high-dimensional feature spaces. Common kernels include linear, polynomial, and radial basis function (RBF).at best separates the data points of different classes.

**Types of Kernels:**

* **Linear Kernel:** Suitable for linearly separable data. The decision boundary is a straight line (or hyperplane in higher dimensions).
* **Polynomial Kernel:** Suitable for non-linear data. It maps the original features into a higher-dimensional space using polynomial functions.
* **Radial Basis Function (RBF) Kernel:** Also known as the Gaussian kernel, it is effective for non-linear data by mapping the features into an infinite-dimensional space.

**Ensemble Methods**

Ensemble methods are powerful machine learning techniques that combine the predictions of multiple base models to produce a single, robust model. The primary goal of ensemble methods is to improve the performance and generalization ability of individual models by leveraging their collective strengths and mitigating their weaknesses. Ensemble methods can be broadly categorized into two main types: Bagging and Boosting.

**Bagging (Bootstrap Aggregating):**

Bagging is an ensemble technique that aims to reduce variance and prevent overfitting by training multiple instances of the same model on different subsets of the training data. These subsets are created by randomly sampling the training data with replacement (bootstrap sampling). The final prediction is typically made by averaging the predictions (for regression) or taking a majority vote (for classification) from all the models.

* **Random Forest:** One of the most popular bagging methods, Random Forest, constructs multiple decision trees using different subsets of the data and features. The final prediction is made by aggregating the predictions of all the trees. Random Forest is known for its robustness and ability to handle high-dimensional data.

**Boosting:**

Boosting is an ensemble technique that aims to reduce bias and improve the model's accuracy by sequentially training models. Each new model is trained to correct the errors made by the previous models. The final prediction is a weighted sum of the predictions from all the models, with more accurate models receiving higher weights.

* **AdaBoost (Adaptive Boosting):** AdaBoost is a popular boosting algorithm that adjusts the weights of incorrectly classified instances, giving them more importance in subsequent models. This process continues until a specified number of models are trained or the model achieves a desired level of accuracy.

# Experimental Approach

**1. Data Preparation**

* **Dataset**: The Breast Cancer dataset from sklearn.datasets was used for this experiment.
* **Data Conversion**: The dataset was converted to a Pandas DataFrame for easier manipulation.
* **Missing Values**: Checked for missing values and confirmed that there were none.
* **Feature and Target Split**: The dataset was split into features (X) and target (y).
* **Train-Test Split**: The data was split into training and testing sets using an 80-20 split with stratification to maintain the class distribution.

**2. Data Preprocessing**

* **Standardization**: The features were standardized using StandardScaler to ensure that all features have a mean of 0 and a standard deviation of 1. This step is crucial for algorithms like KNN and SVM that are sensitive to the scale of the data.

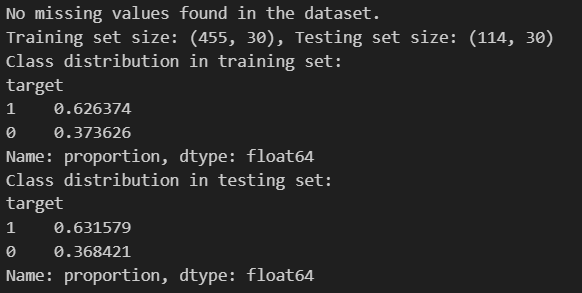


Figure 1: Data Preprocessing

**3. K-Nearest Neighbors (KNN)**

* **Distance Metrics**: Experimented with three different distance metrics: Euclidean, Manhattan, and Cosine.
* **Cross-Validation**: Used cross-validation to determine the optimal value of k (number of neighbors) for each distance metric.
* **Model Training and Evaluation**: Trained KNN models with the optimal k for each distance metric and evaluated their performance using accuracy and classification metrics.

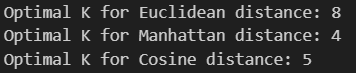


Figure 2: K optimal values

**4. Logistic Regression**

* **Regularization Techniques**: Experimented with two regularization techniques: L1 (Lasso) and L2 (Ridge).
* **Model Training and Evaluation**: Trained Logistic Regression models with L1 and L2 regularization and evaluated their performance using accuracy and classification metrics.

**5. Support Vector Machines (SVM)**

* **Kernels**: Experimented with three different kernels: Linear, Polynomial, and Radial Basis Function (RBF).
* **Model Training and Evaluation**: Trained SVM models with each kernel and evaluated their performance using accuracy and classification metrics.

**6. Ensemble Methods**

* **Boosting**: Trained an AdaBoost model with 100 estimators.
* **Bagging**: Trained a Random Forest model with 100 estimators.
* **Model Training and Evaluation**: Trained the ensemble models and evaluated their performance using accuracy and classification metrics.

**7. Performance Comparison**

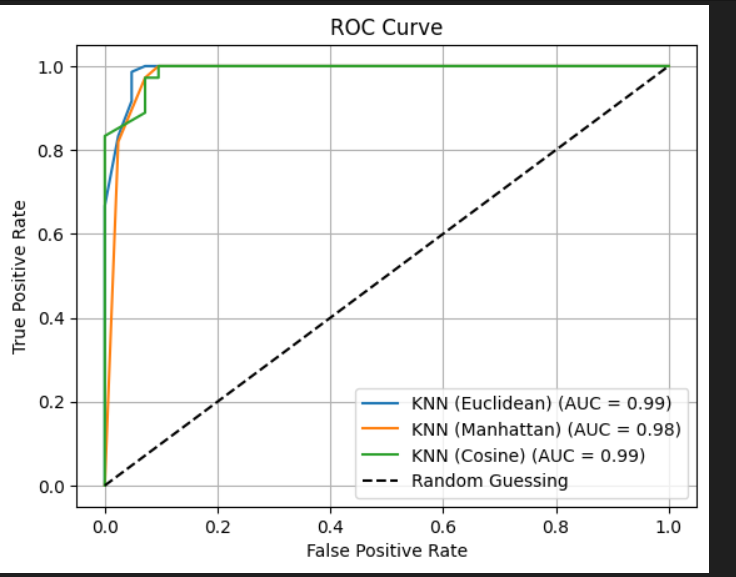
* **Metric Comparison:** Compared the performance of all models (KNN, Logistic Regression, SVM, AdaBoost, and Random Forest) using accuracy, precision, recall, and F1-score and ROC AUC.
* **Discussion:** Discussed the impact of different algorithms, distance metrics, regularization techniques, kernels, and ensemble methods on model performance.

# Results and Analysis

**1. K-Nearest Neighbors (KNN)**

* Optimal K Values:
  + Euclidean Distance: 8
  + Manhattan Distance: 4
  + Cosine Distance: 5
* **Performance Metrics:**

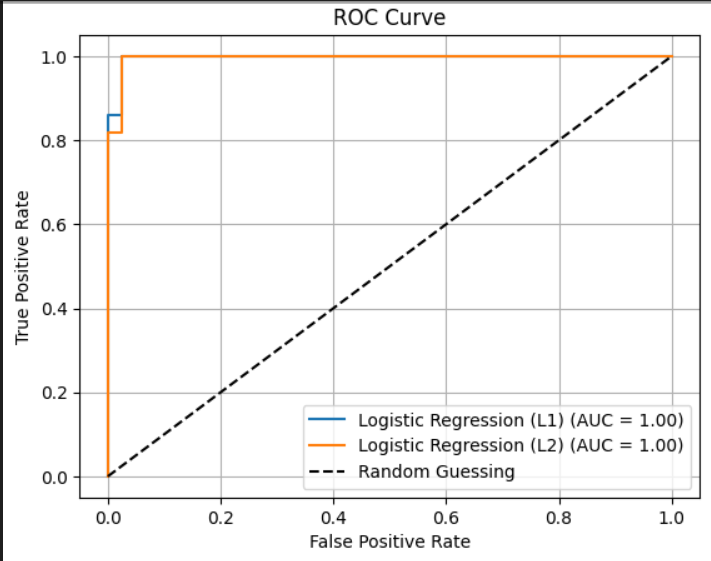
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Precision  (class 0) | Precision  (class 1) | Recall  (class 0) | Recall  (class 1) | F1-Score  (class 0) | F1-Score  (class 1) |
| **Euclidean Distance** | 0.9737 | 0.98 | 0.97 | 0.95 | 0.99 | 0.96 | 0.98 |
| **Manhattan Distance** | 0.9561 | 0.95 | 0.96 | 0.93 | 0.97 | 0.94 | 0.97 |
| **Cosine Distance** | 0.9561 | 0.95 | 0.96 | 0.93 | 0.97 | 0.94 | 0.97 |



**2. Logistic Regression**

* **Performance Metrics:**

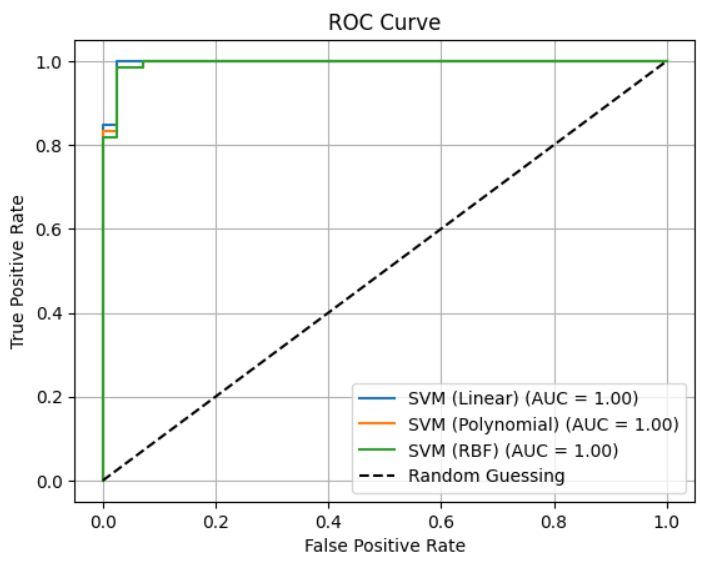
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Precision  (class 0) | Precision  (class 1) | Recall  (class 0) | Recall  (class 1) | F1-Score  (class 0) | F1-Score  (class 1) |
| **L1 Regularization** | 0.9912 | 1 | 0.99 | 0.98 | 1 | 0.99 | 0.99 |
| **L2 Regularization** | 0.9825 | 0.98 | 0.99 | 0.98 | 0.99 | 0.98 | 0.99 |



**3. Support Vector Machines (SVM)**

* **Performance Metrics:**

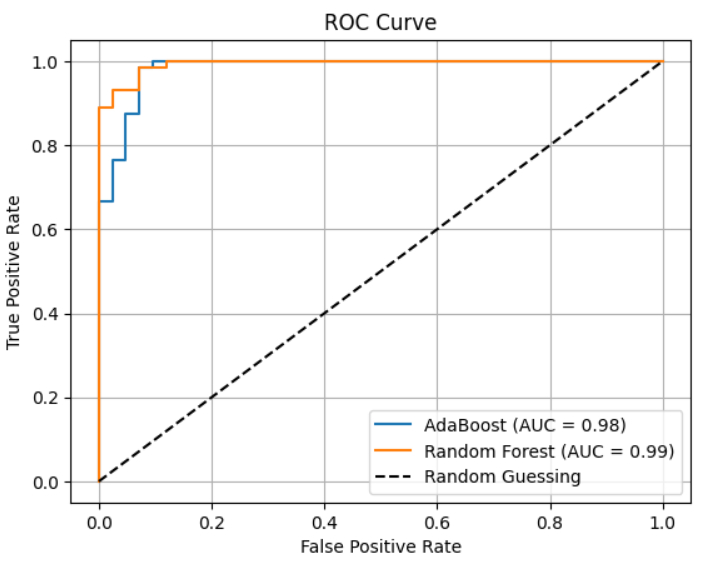
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Precision  (class 0) | Precision  (class 1) | Recall  (class 0) | Recall  (class 1) | F1-Score  (class 0) | F1-Score  (class 1) |
| **Linear Kernel** | 0.9737 | 0.95 | 0.99 | 0.98 | 0.97 | 0.96 | 0.98 |
| **Polynomial Kernel** | 0.9123 | 1 | 0.88 | 0.76 | 1 | 0.89 | 0.94 |
| **RBF Kernel** | 0.9825 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 |



**4. Ensemble Methods**

* **Performance Metrics:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Precision  (class 0) | Precision  (class 1) | Recall  (class 0) | Recall  (class 1) | F1-Score  (class 0) | F1-Score  (class 1) |
| **AdaBoost** | 0.9561 | 0.97 | 0.95 | 0.90 | 0.99 | 0.94 | 0.97 |
| **Random Forest** | 0.9474 | 0.93 | 0.96 | 0.93 | 0.96 | 0.93 | 0.96 |



**Metric Comparisons**

* **Accuracy**: All models performed well, with accuracies ranging from 0.9123 to 0.9912. The highest accuracy was achieved by L1 Regularization (0.9912), followed by RBF Kernel (0.9825), L2 Regularization (0.9825), and Euclidean Distance (0.9737).
* **Precision, Recall, and F1-Score**: The precision, recall, and F1-scores were consistently high across all models, indicating good performance in both classes. L1 Regularization achieved the highest performance with perfect precision and recall for class 0 (1.00), while models like RBF Kernel and L2 Regularization also demonstrated high precision and recall for both classes.

**Analysis**

* **KNN**: The Euclidean distance metric provided the best performance, followed by Manhattan and Cosine. The choice of distance metric had a noticeable impact on the model's accuracy and classification metrics.
  + Euclidean: Consistently high AUC, outperforms Manhattan and Cosine distances.
  + Manhattan & Cosine: Good performance but slightly less effective than Euclidean.
* **Logistic Regression**: Both L1 Regularization and L2 Regularization yielded strong performance, with L1 Regularization achieving slightly better results in precision for class 0 and recall for class 1.
  + L2 Regularization: Slightly better handling of overfitting, robust performance.
  + L1 Regularization: Strong but slightly less effective than L2.
* **SVM**: The Linear and RBF kernels performed equally well, with RBF Kernel achieving the highest performance overall, while the Polynomial kernel showed slightly lower performance.
  + RBF Kernel: Best performance, capturing non-linear patterns effectively.
  + Linear Kernel: Competitive performance but slightly behind RBF.
* **Ensemble Methods**: Both AdaBoost and Random Forest achieved high accuracy and classification metrics, demonstrating the effectiveness of ensemble methods in improving model performance.
  + Random Forest: Excellent performance, marginally outperforms AdaBoost.
  + AdaBoost: Strong performance, close to Random Forest.

# Conclusions

* **Best Performing Models:** L1 Regularization, RBF Kernel, and Euclidean Distance were the top-performing models, achieving the highest accuracy and classification metrics.
* **Impact of Hyperparameters:** The choice of distance metric (KNN), regularization technique (Logistic Regression), and kernel (SVM) had a significant impact on model performance, with L1 Regularization and RBF Kernel showing the best results.
* **Ensemble Methods:** Ensemble methods (AdaBoost and Random Forest) provided robust performance, highlighting their ability to combine the strengths of multiple models and improve overall classification accuracy.