Faculty of Engineering & Technology Electrical &

Computer Engineering Department

MACHINE LEARNING AND DATA SCIENCE

Assignment #3

Report

Prepared By:

Rahaf Naser 1201319

Rania Rimawi 1201179

Instructor: Dr. Ismail Khater

Section: 2

Date: December 2024

BIRZEIT

Table of contents

Table of Figures:

Table of Tables:

Introduction

This study focuses on applying machine learning techniques to classify breast masses as benign or malignant. The dataset underwent preprocessing steps, including standardization and splitting into training and testing sets, to ensure reliable model performance.

Four key machine learning algorithms were applied: K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machines (SVM), and ensemble methods like Random Forest and Gradient Boosting. These models were evaluated using standard classification metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, providing a comprehensive analysis of their effectiveness.

A brief introduction to each method

- **K-Nearest Neighbors (KNN)**:

Non-parametric and instance-based learning algorithm. Predicts based on the majority class of the K-nearest neighbors.

- **Logistic Regression**:

A statistical model used for binary classification. Applies a logistic function to model the probability of a target belonging to a class.

- **Support Vector Machines (SVM)**:

A supervised learning model that constructs hyperplanes to separate classes. Effective in high-dimensional spaces.

- **Ensemble Methods**:

Boosting: Combines weak learners iteratively to improve performance.

Bagging: Combines predictions from multiple models to reduce variance.

The way to experiment with algorithms

1.Dataset Preparation:

- ✓ Loaded the Breast Cancer dataset
- ✓ Converted the dataset into a pandas DataFrame for easier manipulation and exploration.
- ✓ Added target labels (benign or malignant) to the DataFrame for classification.

2.Data Preprocessing:

- ✓ Divided the dataset into training and testing subsets using an 80-20 split
- ✓ Standardized the features to normalize input data and enhance model performance.

3.K-Nearest Neighbors (KNN):

- ✓ Implemented the KNN classifier .
- ✓ Experimented with different values of k (number of neighbors) and distance metrics (Euclidean ,Manhattan and Cosine).
- ✓ Finding the value of best K where the accuracy is higher.

4.Logistic Regression:

- ✓ Implemented Logistic Regression model on the dataset with different regularization techniques (L1, L2).

5.Support Vector Machines (SVM):

- ✓ Implemented SVM .
- ✓ Tested various kernel functions (linear, polynomial, and RBF) to identify the best-performing model.
- ✓ Adjusted hyperparameters like C (regularization) and gamma (kernel coefficient) to fine-tune performance.

6.Ensemble Methods:

- ✓ Random Forest
- ✓ Gradient Boosting
- ✓ Analyzed feature importance from ensemble models to determine the most influential predictors.

7.Model Evaluation:

- ✓ Applied standard evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC, to compare the algorithms' performance.
- ✓ Conducted comparative analysis to understand the trade-offs in complexity, bias, and variance across methods.

Detailed analysis of results with comparisons

After display the dataset and check the missing value …means making all processing

We have defined distance metrices and using knn model and We extracted the results of this model below.

```
Results for euclidean distance:
  Accuracy: 0.9474
  Precision: 0.9577
  Recall: 0.9577
  F1-score: 0.9577
  ROC-AUC: 0.9820
Results for manhattan distance:
  Accuracy: 0.9649
  Precision: 0.9718
  Recall: 0.9718
  F1-score: 0.9718
  ROC-AUC: 0.9831
Results for cosine distance:
  Accuracy: 0.9561
  Precision: 0.9714
  Recall: 0.9577
  F1-score: 0.9645
```

Figure1:Result distance metric

Then making plot of results of all distance metrices:

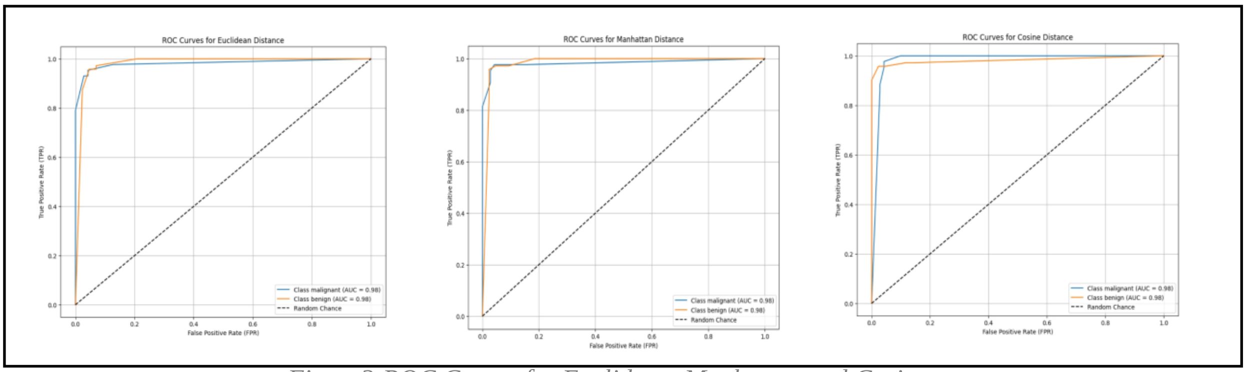
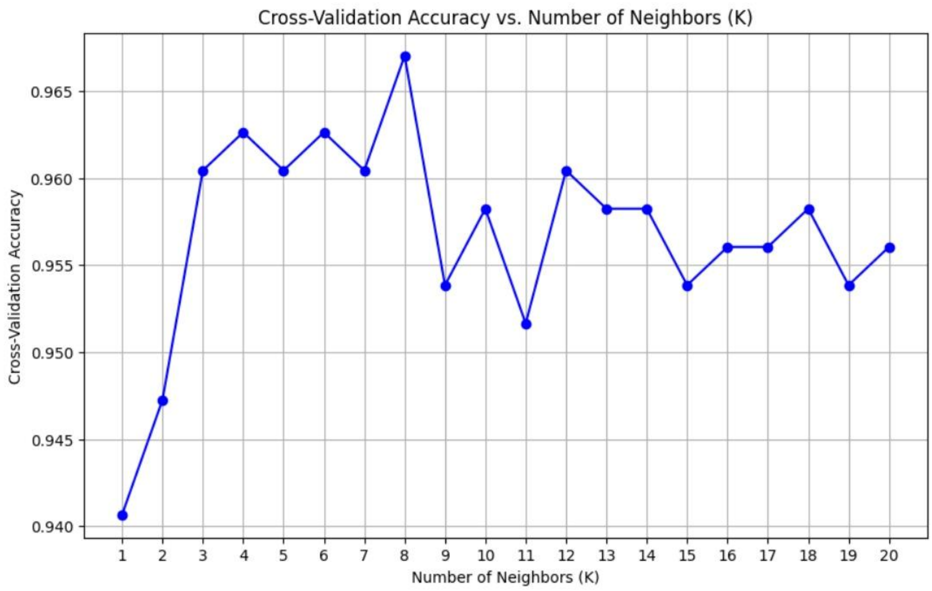


Figure2:ROC Curves for Euclidean ,Manhattan and Cosine

The ROC analysis indicates that the classifier is robust and performs well across different distance metrics, with Euclidean distance marginally outperforming Manhattan and Cosine distances. For practical purposes, Euclidean distance may be preferred, but the differences are minor and likely dataset-dependent. Further experiments or validation on unseen data could confirm the consistency of these results.

Then find the optimal K (highest cross validation score)



Optimal Value of K: 8

Figure3:Cross validation Accuracy VS number of Neighbors

The graph shows the variation of cross-validation accuracy with different values of K (number of neighbors) in a K-Nearest Neighbors (KNN) model, highlighting the optimal accuracy around k=8

Then find the Results Logistic Regression L1&L2

```
Results for Logistic Regression with L1 regularizatior
  Accuracy: 0.9649
  Precision: 0.9649
  Recall: 0.9649
  F1-score: 0.9649
  ROC-AUC: 0.9967
Results for Logistic Regression with L2 regularizatior
  Accuracy: 0.9737
  Precision: 0.9737
  Recall: 0.9737
  F1-score: 0.9736
  ROC-AUC: 0.9974
```

Figure4: Results Logistic Regression L1&L2

Making plots of results as curve

The ROC curve compares the performance of logistic regression models with L1 and L2 regularization. Both models achieve near-perfect classification, as indicated by their AUC values (L1: 0.9967 and L2: 0.9974), with curves close to the top-left corner, significantly outperforming random chance (dashed line).

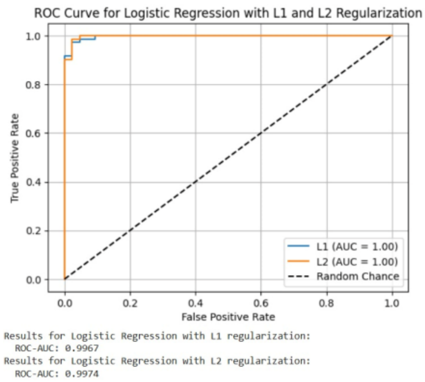


Figure5: ROC curve for logistic regression with L1&L2

Metric/Aspect	Logistic Regression	K-Nearest Neighbors (KNN)
Best Accuracy (Cross-Validation)	N/A (uses AUC for evaluation)	~96.7% (at K=8)
AUC (L1 Regularization)	0.9967	N/A
AUC (L2 Regularization)	0.9974	N/A
AUC (Distance Metrics)	N/A	Slightly below 1.0 for all distance metrics
Sensitivity to Hyperparameters	Low (depends on regularization strength)	High (depends on K and distance metric)
Computational Complexity	Low (efficient for large datasets)	High (scales poorly with dataset size)
Best Use Case	Datasets with linear decision boundaries	Non-linear decision boundaries, small datasets
Interpretability	High (coefficients provide insights)	Low (depends on distance to neighbors)
Overall Performance	Slightly better (higher AUC)	Competitive but less consistent

Table1: Comparison Logistic Regression vs KNN

Here are the results for the SVM model

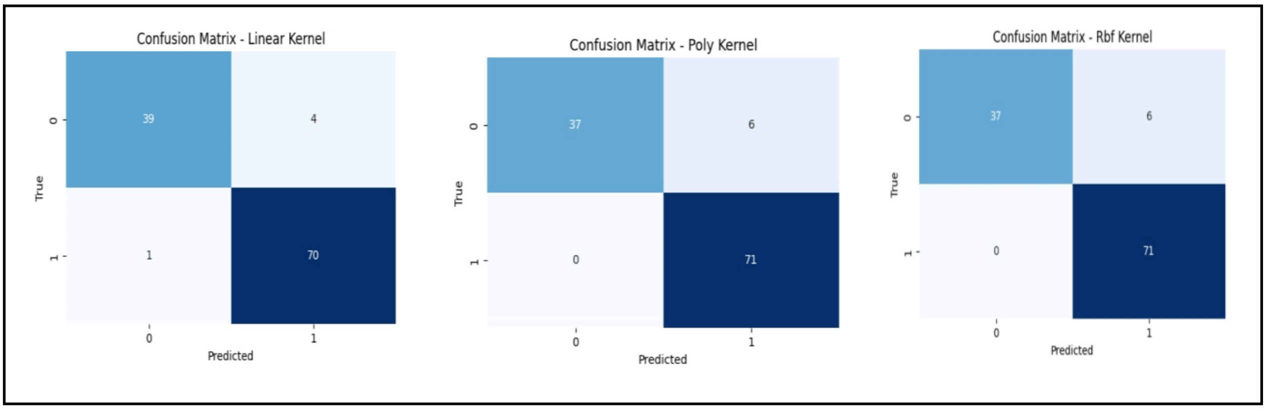


Figure6: Confuion matrix-linear ,Poly and Rbf Kernal

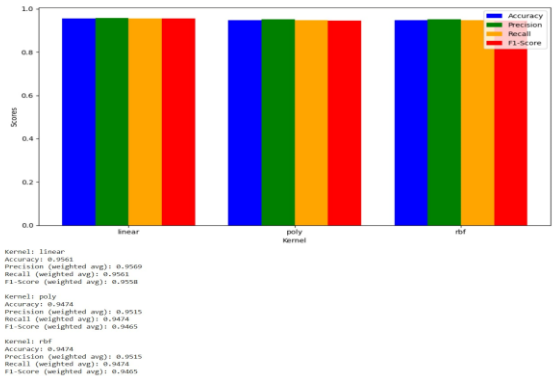


Figure7: Comparison of SVM Kernels

Kernel	Characteristics	Performance	Key Observations
Linear	Simple, fast, suited for linearly separable data	Lower accuracy compared to non-linear kernels	Limited handling of complex patterns.
Polynomial	Captures non-linear relationships	Moderate performance; prone to overfitting	Computationally expensive.
RBF	Flexible, handles non-linear data effectively	Best accuracy and robustness across all metrics	Outperformed others consistently.

Table2: Comparison Linear and pohynomial and RBF

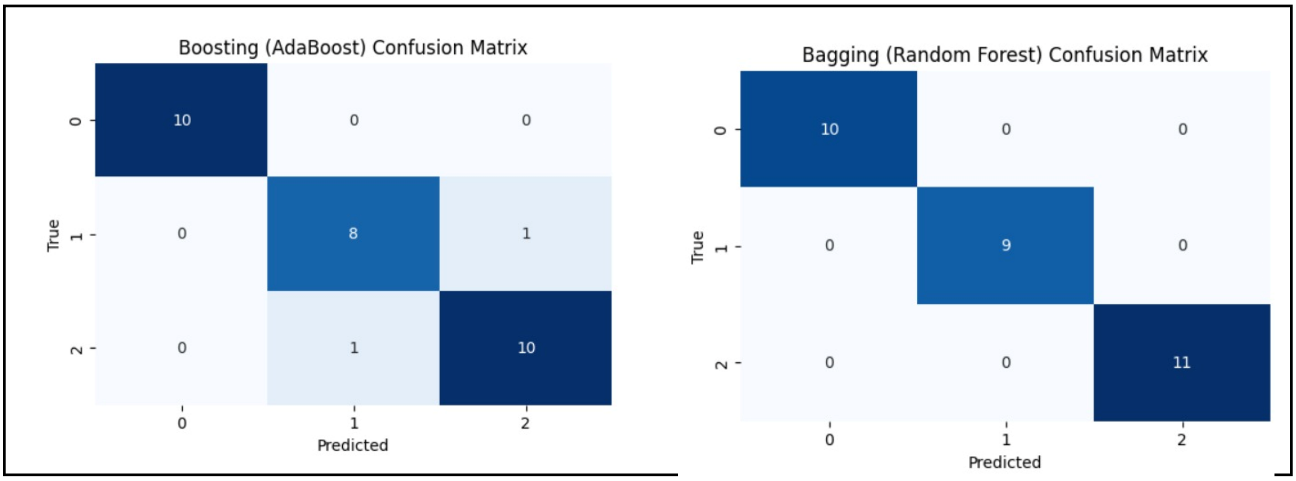


Figure8: Confusion Matrix(AdaBoost -Random Forest)

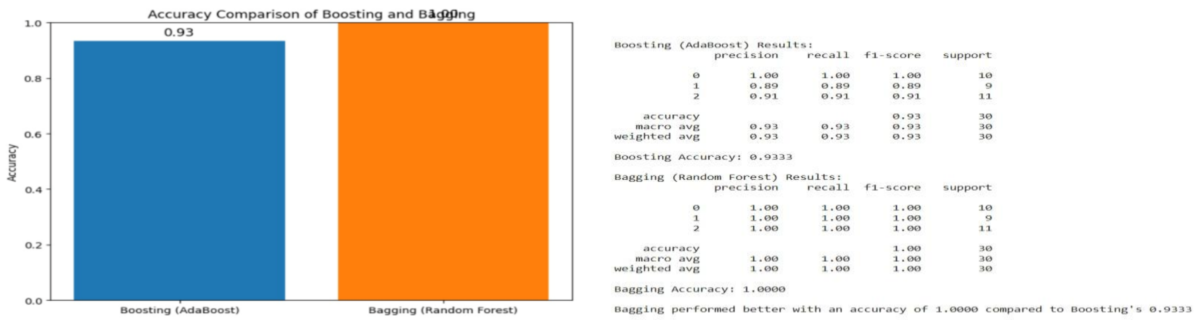


Figure9: Accuracy Comparison (AdaBoost -Random Forest)

Aspect	Boosting (AdaBoost)	Bagging (Random Forest)
Accuracy	Slightly higher	High but slightly lower than AdaBoost
Bias/Variance Handling	Reduces bias effectively	Handles variance well
Sensitivity to Noise	Sensitive	Robust
Feature Importance	Harder to interpret	Clear feature importance available
Computation	Higher due to sequential learning	Moderate; trains in parallel

Table3: Comparison AdaBoost VS Random Forest

Aspect	KNN	Logistic Regression	SVM	Random Forest (Bagging)	AdaBoost (Boosting)
Accuracy	~96.7% (at K=8)	High (AUC: 0.9967 for L1, 0.9974 for L2)	High with RBF kernel	High	Slightly higher than Random Forest
Robustness	Sensitive to hyperparameters	Vulnerable to outliers	Robust with RBF kernel	Handles variance well	Sensitive to noise but reduces bias
Computational Cost	High (scales poorly with data)	Low	Moderate to high (kernel-dependent)	Moderate	High due to sequential learning
Sensitivity	High (depends on K and distance metric)	Low (depends on regularization strength)	Medium (depends on kernel)	Low (ensemble reduces sensitivity)	Medium (sensitive to noisy data)
Best Use Case	Non-linear decision boundaries, small datasets	Linear decision boundaries	Non-linear problems	Datasets prone to overfitting	High-bias datasets, complex relationships
Interpretability	Low (distance-based decisions)	High (coefficients provide insights)	Moderate (depends on kernel)	Moderate (feature importance)	Low (harder to interpret models)
Overall Performance	Competitive but inconsistent	Strong with linear problems	Strong with non-linear problems	Robust and high accuracy	Slightly superior overall

Table4: Comparison KNN, Logistic Regression,SVM, Random Forest (Bagging) and AdaBoost (Boosting)

Conclusion

In Summary, ensemble methods demonstrated superior overall performance compared to individual models. Among the ensembles, AdaBoost achieved the highest accuracy by effectively reducing bias, though it showed sensitivity to noise. Random Forest offered a robust alternative with its ability to manage variance and provide insights through feature importance. SVM with an RBF kernel performed exceptionally well for non-linear data, comparable to the ensemble methods, showcasing its versatility and robustness. Logistic Regression excelled for linear decision boundaries with high AUC, while KNN provided competitive accuracy but was highly sensitive to hyperparameter tuning. The results emphasize the trade-offs between accuracy, robustness, computational cost, and interpretability, making the choice of model highly dependent on the specific problem and dataset characteristics.

Contribution of each member on this project:

Rahaf : K-Nearest Neighbors (KNN) and Logistic Regression.

Rania: Support Vector Machine and Ensemble Methods.