جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

Machine Learning and Data Science - ENCS5341

Assignment #3

Classification, Logistic regression, SVM, and Ensemble

Prepared by:

- Ali Shaikh Qasem ID: 1212171
- Abdalrahman Juber ID: 1211769

Instructors: Dr. Ismail Khater, Dr. Yazan Abu Farha.

Sections: 1,3.

Date: 25/12/2024

Birzeit University

2024-2025

Contents

List of Figures

List of Tables

1 Brief Dataset Description

In this assignment, we used the **Breast Cancer dataset** which performs diagnosis classification for patient records. It contains **30** different features like radius_mean, texture_mean, concavity_mean and many other features. The dataset has a binary class label with two possible values**: M (malignant) and B (benign).** The dataset has **567** different samples.

Pre-processing steps:

The dataset was clean; however, we performed the following steps to ensure consistency and reliable performance of the model:

- Normalizing features:

All features were normalized to the range (0-1), to avoid the dominance of high ranges features.

- Encoding output label:

The output label was encoded as shown M = 1, B = 0 to ensure compatibility with classification algorithms.

The dataset was splitted as shown: 80% training and 20% testing.

2 K-Nearest Neighbors (KNN)

2.1 Introduction

K-Nearest Neighbors (KNN) is a simple supervised machine learning algorithm used for classification and it's unlike the regression algorithm a non-parametric algorithm which means it makes no assumptions about the data (no weights). KNN works with k-nearest data (neighbors) based on distance metrics (e.g., Euclidean, Manhattan, and Cosine distance) which of the three of them was used.

2.2 Approach of the experiment

For this method the following systematic approach was followed:

Using APIs:

Building KNN: We import KNeighborsClassifier from sklearn.

Building cross-validation: We import cross_val_score from sklearn.

And we used Euclidean distance with accuracy metric. Setting cv to 5.

2.3 Analyze the results and discuss

2.3.1 How do different distance metrics affect classification performance?

Table 2-1: Different distance metric performance

	Euclidean	**Manhattan**	**Cosine**
Accuracy	0.965	0.965	0.904
Precision	0.953	0.9535	0.8333
Recall	0.9535	0.9535	0.9302
F1-score	0.954	0.955	0.879
ROC-AUC	0.963	0.963	0.909

As we can see the results for Euclidean and Manhattan are more accurate for our data set. Meanwhile, the cosine is less accurate, likely it focuses on the angular relationship between vectors, which might not capture the critical patterns in the dataset.

What is the best value of K for your dataset, and why?

We used cross-validation to obtain the best value of k we obtained the value 19.

Cross-validation maintains a value of k without overfitting.

ROC curve:

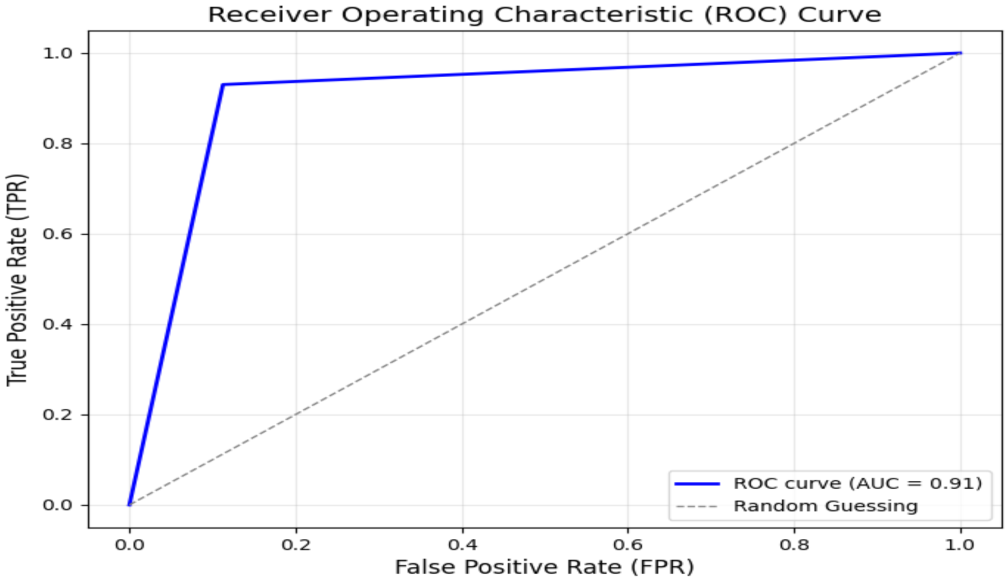


Figure 2-1: ROC curve for cosine distance

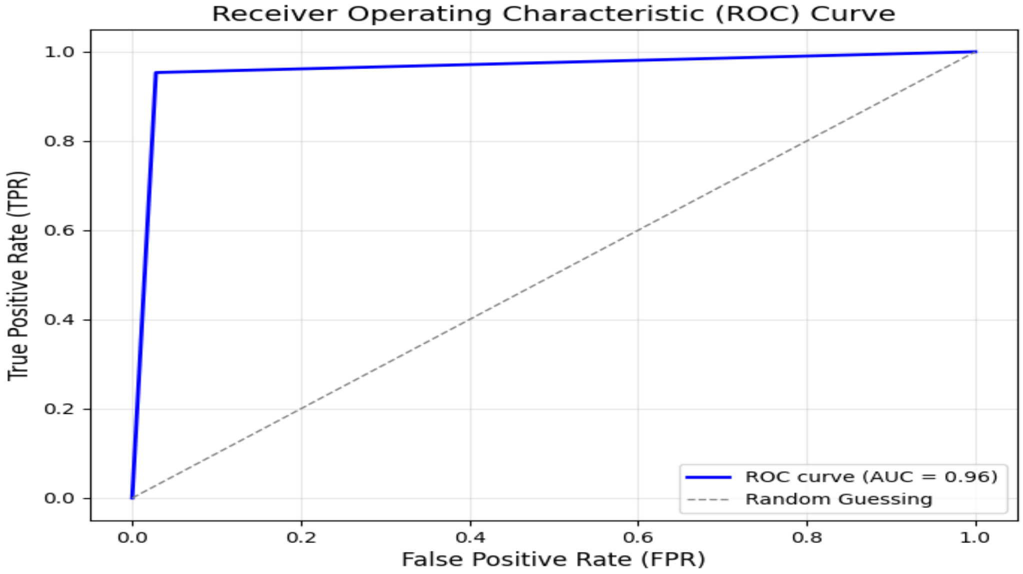


Figure 2-2: ROC curve for Euclidean distance

For both fig1 and fig 2 we can see that the area for fig 2 is bigger which means better.

3 Logistic Regression

3.1 Introduction

Logistic Regression is an algorithm used for binary classification problems. It uses a linear combination of input features with corresponding weights like normal regression, but it passes them through a sigmoid function to predict probabilities from 0 to 1 for each class label. The model is trained by maximizing the log-liklihood function using gradient ascent method.

3.2 Approach of the Experiment

For Logistic Regression, we used LogisticRegression function using sklearn.linear_model API, the API was used to fit the model using training set and predict values using testing set.

Our experiment was performed using three types of Logistic Regression: Normal, L1 regularization, and L2 regularization.

3.3 Results Analysis and Discussion

The following table summarizes the results after applying each logistic regression classifier to the testing set:

	Normal	**L1 Regularization**	**L2 Regularization**
Accuracy	0.982	0.956	0.982
Precision	1.0	0.952	1.0
Recall	0.953	0.930	0.953
F1-score	0.976	0.941	0.976
ROC-AUC	0.977	0.951	0.977

Table 3-1: performance metrics for logistic regression

From results above we see that the normal logistic regression and L2 regularization produced the best performance across all metrics, while L1 regularization showed a slight trade-off in performance, particularly in recall and precision.

3.4 Comparing Logistic regression with KNN

Logistic regression generally performs better than the KNN classifier, especially in terms of accuracy, precision, F1-score, and ROC-AUC. KNN with Euclidean and Manhattan metrics performs similarly but is less precise and has lower overall performance compared to logistic regression. KNN with the Cosine metric has the weakest results across most metrics.

ROC curve for KNN:

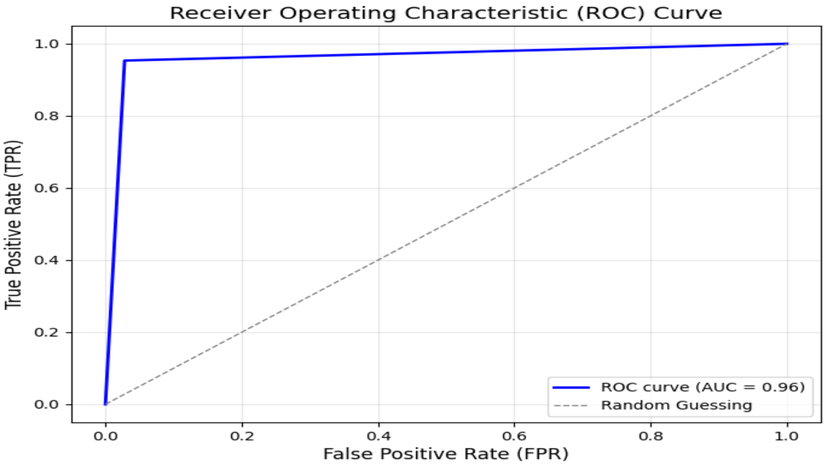
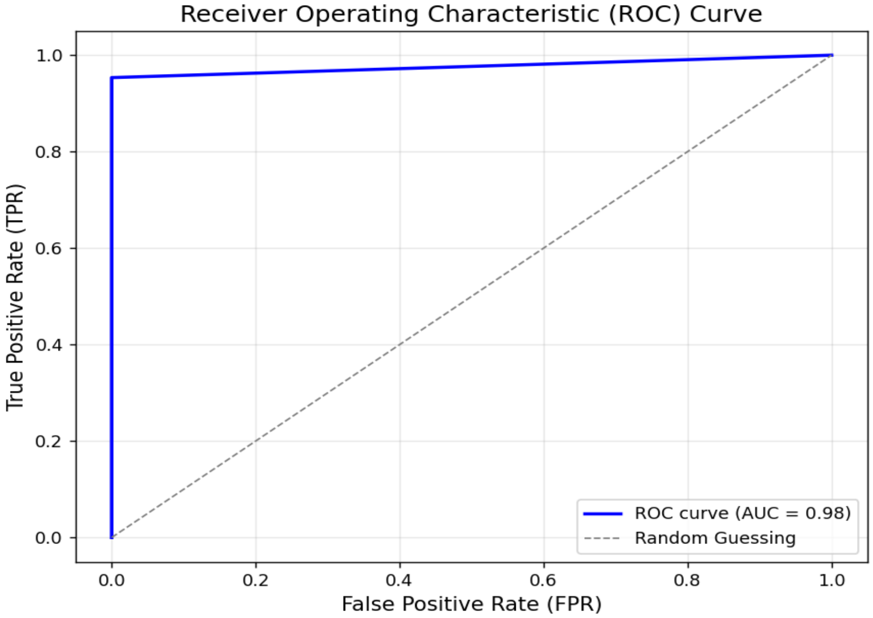


Figure 3-1: ROC curve for Euclidean distance KNN

ROC curve for Logistic regression with L2 regularization:



Its clear that the logistic regression performs better since the area under the ROC curve is clearly larger than the KNN ROC curve.

4 Support Vector Machines (SVM)

4.1 Introduction

Support Vector Machines (SVM) is a supervised learning algorithm used mostly for classification. The idea of SVM is to find a hyperplane that best separates data points from different classes. SVM can handle linear and non-linear problems using kernels, such as polynomial, radial basis function (RBF) by transforming data into higher dimensions.

4.2 Approach of the experiment

For this method the following systematic approach was followed:

Using APIs:

Building SVM: We import SVC from sklearn.

Building kernels: same as SVC.

We used poly kernels with degree equals 3.

4.3 Analyze the results and discuss

4.3.1 Compare the performance of the kernels using classification metrics.

Table 4-1: performance metrics for SVM

	Linear	Poly (degree = 3)	RBF
Accuracy	0.983	0.983	0.974
Precision	1.0	1.0	0.975
Recall	0.954	0.954	0.954
F1-score	0.98	0.976	0.965
ROC-AUC	0.977	0.978	0.970

The linear kernel performs the best in terms of overall accuracy and has very high precision, recall, F1-score, and ROC-AUC, suggesting that it is a good choice for this dataset.

The polynomial (degree = 3) kernel shows similar results to the linear kernel but has a slightly lower F1-score, meaning that it may introduce more complexity without significantly improving the results.

The RBF kernel has slightly lower values across all metrics compared to the linear and polynomial kernels, indicating that it may not be as well-suited for this dataset in terms of generalization and balance between precision and recall.

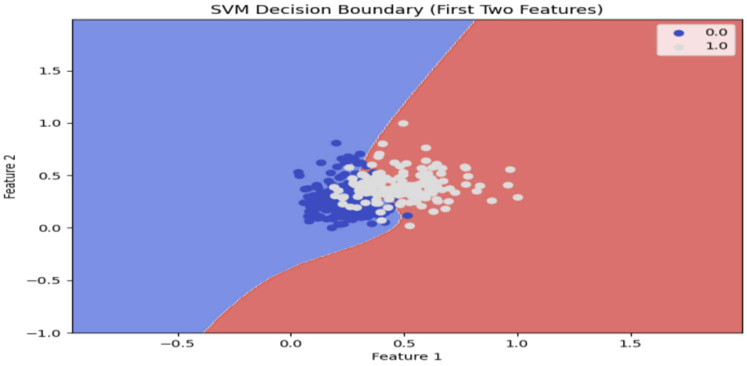


Figure 4: SVM with poly kernels

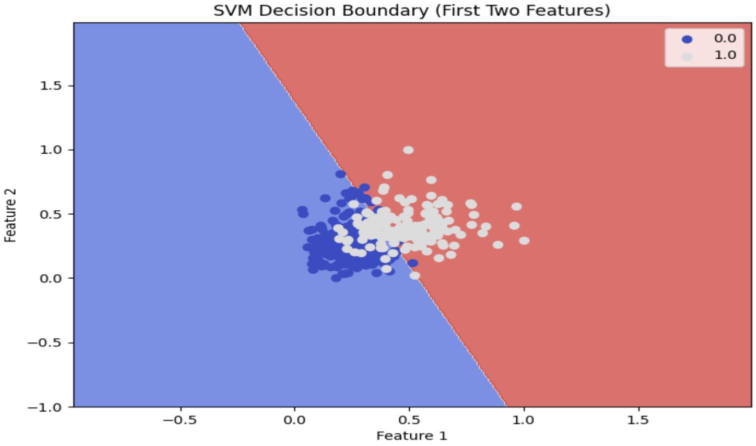


Figure 5: SVM with linear kernels

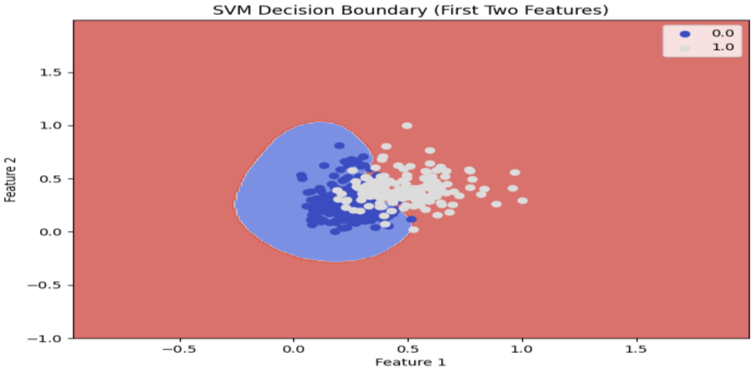


Figure 6: SVM with RBF kernels

5 Ensemble Methods

5.1 Introduction

Ensemble methods is a technique that combines multiple models to improve performance. It contains two main concepts: Bagging and Boosting. Bagging trains models on different data subsets and averages their predictions like Random Forest classifier. Boosting trains models sequentially, with correcting the previous model's errors like AdaBoost and other techniques.

5.2 Approach of the Experiment

We used the API sklearn.ensemble to train the classifiers .

For **Boosting,** we used AdaBoost classifier.

For **Bagging,** we used Random Forest classifier.

For both techniques we used **100** weak learners.

5.3 Results Analysis and Discussion

The following table summarizes the results after applying each logistic regression classifier to the testing set:

	Boosting	**Bagging**
Accuracy	0.973	0.964
Precision	0.976	0.976
Recall	0.953	0.930
F1-score	0.964	0.952
ROC-AUC	0.970	0.958

Table 5-1: performance results for ensemble methods

It's clear from results above that Boosting performs better than Bagging in all metrices. This can be legal since our dataset is small, thus, boosting focuses on hard-to-classify instances, maximizing the use of available data by giving higher importance to challenging cases. While Bagging may suffers overlapping between different subsets which leads to less improvement.

5.4 Comparison between Ensemble methods and previous Classifiers

SVM with Linear and Polynomial kernels performs best overall, with boosting close behind in terms of balanced performance. Bagging and KNN are less effective, particularly in recall and F1-score, making boosting a strong contender for this dataset.

6 Conclusion

This assignment evaluated various classification techniques on the Breast Cancer dataset. After preprocessing the data, we applied K-Nearest Neighbors (KNN) with different distance metrics, finding that Euclidean and Manhattan metrics performed best, while Cosine showed lower accuracy. Logistic Regression performed well across all variants, with normal and L2 regularization achieving the highest scores in accuracy, precision, and F1-score. Support Vector Machines (SVM) with the Linear kernel outperformed the Polynomial and RBF kernels, making it the most suitable choice. Finally, Ensemble Methods (AdaBoost and Random Forest) improved performance, with Boosting providing the best results. Overall, Logistic Regression and SVM (Linear kernel) were the most effective, with ensemble methods offering valuable performance enhancements.