Brandenburgische Technische Universität Cottbus-Senftenberg
Institute für Infomatik
Chair of Database and Information Systems

Brandenburgische
Technische Universität
Cottbus - Senftenberg

# Practical Training – Task 2
# Implementation and Application of Classification Methods

Module: Foundations of Data Mining

Group 08

1 Shaik Fharook
MatrikelNr.: 5014962

2 Yalla Pavani Bhuvaneshwari
MatrikelNr.: 5011106

*Date of Submission: 28-Jan-2025*

Supervisors:
1 Prof. Dr.-Ing. habil. Ingo Schmitt
2 Alexander Stahl, M. Sc.

# 1. Introduction

This report provides an implementation and comparative analysis of three classification methods applied to real-world datasets. The study compares a custom implementation of the K-Nearest Neighbors (KNN) algorithm with two established classifiers from the scikit-learn library: KNeighborsClassifier and Gaussian Naïve Bayes. The analysis evaluates both the accuracy and computational efficiency of these methods across various classification scenarios.

# 2. Experiment Setup

## 2.1. Datasets

- **Breast Cancer Wisconsin (Diagnostic) Dataset:** This binary classification dataset comprises 569 samples and 30 features, sourced from the UCI Machine Learning Repository. Each sample represents a set of measures derived from cell nuclei images, including mean radius, texture, and perimeter. The objective is to classify tumors as benign (Class 0) or malignant (Class 1) based on these cell image characteristics.
- **Wine Quality Dataset:** This multiclass classification dataset contains 4898 samples of both red and white wines, along with 11 features, also sourced from the UCI Machine Learning Repository. The features include physiochemical properties such as alcohol content, pH, and residual sugar. The goal is to classify wine quality into six classes, ranging from 3 (lowest quality) to 8 (highest quality)

## 2.2. Classifiers

- **CustomKNNClassifier:** This classifier implements the K-Nearest Neighbors algorithm, utilizing Euclidean distance to calculate similarity and a majority vote among the k-nearest neighbors to predict labels. The computation of distances occurs pairwise, resulting in high computational complexity for larger datasets.
- **KNeighborsClassifier:** This classifier, part of the scikit-learn library, optimizes neighbor searches using data structures such as KDTree or BallTree. It employs the same majority voting principle but operates significantly faster due to its optimized implementation.
- **Gaussian Naïve Bayes:** This probabilistic classifier is based on Bayes' theorem and assumes that all features are conditionally independent, following a Gaussian distribution for continuous features. Its simplicity enhances efficiency, although it may be less effective for datasets exhibiting complex interdependencies among features.

## 2.3. Data Preprocessing

Before training, both datasets were standardized to have a mean of 0 and a standard division of 1 to ensure that all features have similar scales. This step prevents any dominant influence in distance-based calculations for KNN and probabilistic calculations for Naïve Bayes. The datasets were divided into training and testing sets using stratified sampling to maintain the

class distribution in both splits. For each dataset, 80% of data is allocated for training, and the remaining 20% for testing.

### 2.4. Hyperparameter Tuning

- **Gaussian Naïve Bayes**: This model relies on the assumption that features are conditionally independent, and it is applied with its default parameters for both datasets. Hyperparameter tuning was not performed for Gaussian Naïve Bayes, as it works well with its default settings.
- **KNN Classifiers (CustomKNNClassifier or KNeighborsClassifier)**: For both KNN-based models, the default value of *k=5* was first experimented with, and the Euclidean distance metric is used for distance calculations. Afterward, a hyperparameter search was performed for *k* values ranging from *1 to 50* to find the optimal k for both classifiers. The optimal *k* was determined based on the accuracy achieved on the test set.

### 2.5. Evaluation Metrics:

The evaluation of model performance utilized the following metrics:

- **Test Accuracy:** The proportion of correctly classified instances.
- **Recall:** The ability of the classifier to correctly identify positive instances (useful in imbalanced datasets).
- **Precision:** The proportion of positive predictions that were actually correct.
- **Training Time:** The time taken to fit the model on the training data.
- **Prediction Time:** The time taken to make predictions on the test data.

## 3. Presentation of Results

The table below summarizes the performance of the classifiers on both datasets.

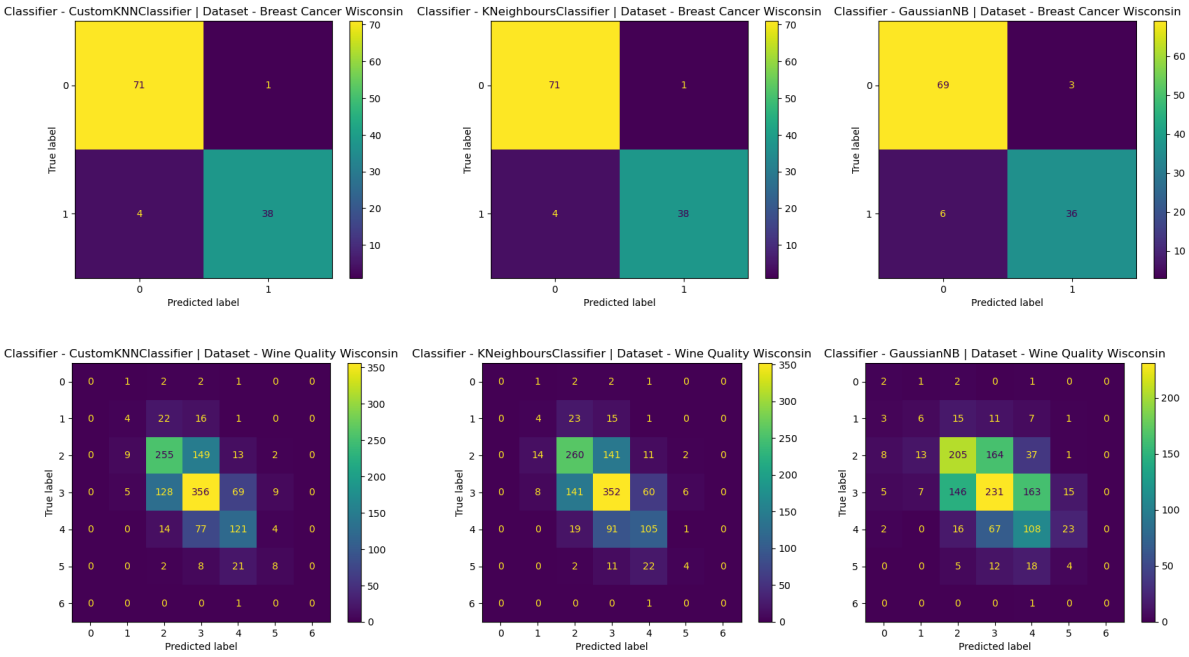
| Dataset | Classifier | Test Accuracy | Recall | Precision | Training Time | Prediction Time |
|---|---|---|---|---|---|---|
| Breast Cancer Wisconsin Diagnostic | CustomKNNClassifier | 0.9561 | 0.9862 | 0.9466 | 1.9073 | 0.1332 |
| | KNeighborsClassifier | 0.9561 | 0.9862 | 0.9466 | 0.0006 | 0.0371 |
| | Gaussian Naïve Bayes | 0.9210 | 0.9583 | 0.9200 | 0.0017 | 0.0003 |
| Wine Quality Dataset | CustomKNNClassifier | 0.5723 | 0.5723 | 0.5652 | 5.9604 | 15.9903 |
| | KNeighborsClassifier | 0.5577 | 0.5577 | 0.5486 | 0.0018 | 0.0422 |
| | Gaussian Naïve Bayes | 0.4277 | 0.4277 | 0.4461 | 0.0013 | 0.0004 |

### 3.1. Confusion Matrices

The confusion matrices for these experiments were saved as CSV files, as following:

- g008_d2_c1.csv: CustomKNNClassifier on Breast Cancer Wisconsin dataset.
- g008_d2_c2.csv: KNeighborClassifier on Breast Cancer Wisconsin dataset.
- g008_d2_c3.csv: Gaussian Naïve Bayes on Breast Cancer Wisconsin dataset.
- g008_d3_c1.csv: CustomKNNClassifier on Wine Quality Dataset.

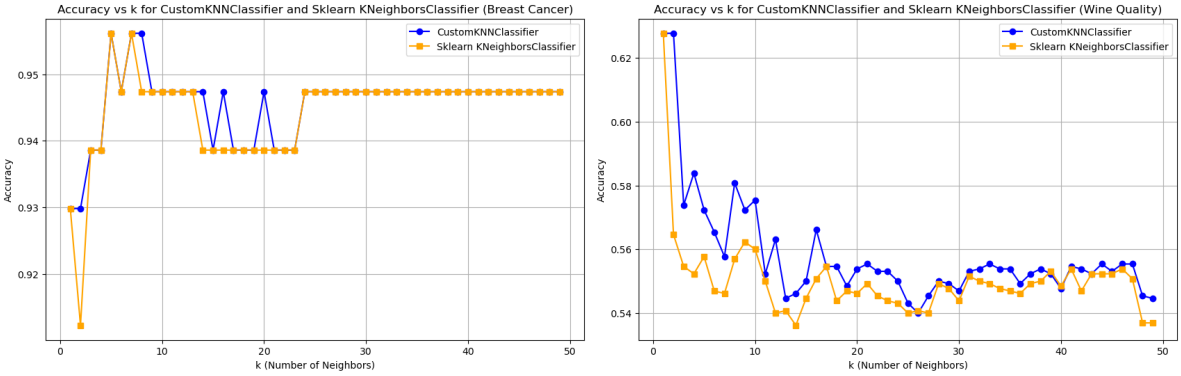
- g008_d3_c2.csv: KNeighborClassifier on Wine Quality Dataset.
- g008_d3_c3.csv: Gaussian Naïve Bayes on Wine Quality Dataset.




## 3.2. Optimal *k* Analysis

Through hyperparameter tuning, it is found that the optimal value of k for both CustomKNNClassifier and KNeighborsClassifier are as follows:

- **Breast Cancer Wisconsin Diagnostic Dataset:** *k=5,* with the highest accuracy of 0.9561. This k value balanced underfitting and overfitting, resulting in a well-generalized model.
- **Wine Quality Dataset:** *k = 1,* with the highest accuracy of 0.6277. A smaller k proved more effective due to the noise and complexity inherent in the dataset, where localized decision-making (considering fewer neighbors) enhanced performance.




# 4. Discussion

For the Breast Cancer Wisconsin Diagnostic dataset, both CustomKNNClassifier and KNeighborsClassifier demonstrated exceptional performance, yielding nearly identical results.

The high accuracy, recall, and precision indicate that these models are well-suited for binary classification problems characterized by clearly separable classes. The KNN Classifiers, particularly with an optimal k of 5, outperformed Gaussian Naïve Bayes in scenarios where feature correlations were strong. Despite its computational speed, Naïve Bayes struggled to maintain the same accuracy level due to its stringent assumptions regarding feature independence.

The performance on the Wine Quality dataset was notably lower, especially for Gaussian Naïve Bayes, which encountered challenges due to feature correlation issues. Although KNN-based models outperformed Naïve Bayes, the overall results remained relatively poor. This observation suggests that the complexity of the Wine Quality dataset necessitates a more sophisticated approach, potentially utilizing advanced ensemble methods or neural networks to achieve higher accuracy.

The training times for all classifiers were generally rapid, particularly for Gaussian Naïve Bayes. However, KNN-based models exhibited slower prediction times, especially with the Wine Quality dataset, due to the larger sample and feature sizes. While the training time for KNN models remained relatively low, the prediction time escalated significantly as the dataset increased in size. This trade-off merits consideration when employing KNN for large datasets.

## 5. Conclusion

This study established that KNN-based classifiers (CustomKNNClassifier and KNeighborsClassifier) are highly effective for binary classification tasks, as evidenced by their excellent performance on the Breast Cancer dataset. The optimal value of k (5) achieved a balance between underfitting and overfitting, resulting in high accuracy, precision, and recall. However, the performance of KNN Classifiers proved suboptimal for more complex datasets, such as Wine Quality, indicating a need for more localized decision-making, as demonstrated by the suboptimal k value of 1 for this dataset.

While Gaussian Naïve Bayes exhibits computational efficiency, it struggles in scenarios characterized by significant feature dependencies, particularly within the Wine Quality dataset. These findings underscore the trade-offs between computational efficiency and model performance, emphasizing the necessity for algorithm selection to align with the complexity and structure of the dataset.

## 6. References


- Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1993). Breast Cancer Wisconsin (Diagnostic) [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.
- Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Wine Quality [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C56S3T.
- Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.
- The complete codebase, model implementation, hyperparameter tuning, and evaluation, is available on GitHub: fharookshaik/BTU_FDM