Breast Cancer Classification GUI Application

Abdelrahman Islam

Student ID: 2305152

Date: May 2025

Abstract

This project presents a desktop application developed using Python and PyQt5 for classifying breast cancer tumors as benign or malignant. Leveraging the UCI Breast Cancer Wisconsin dataset, the application integrates data preprocessing, fuzzy feature engineering, decision tree classification, and multiple evaluation visualizations. The GUI facilitates intuitive interaction, enabling users to explore data, visualize patterns, train models, and assess performance metrics.

1. Introduction

Breast cancer remains a leading cause of mortality among women worldwide. Early detection and accurate diagnosis are crucial for effective treatment and improved survival rates. This project aims to develop a user-friendly application that assists in the classification of breast cancer tumors, providing healthcare professionals with a tool to support diagnostic decisions.

2. Objectives

- Develop a GUI application for breast cancer classification.
- Implement data preprocessing techniques, including scaling and fuzzy feature engineering.
- Train a Decision Tree Classifier with hyperparameter tuning.
- Provide visualizations for data exploration and model evaluation.
- Facilitate user interaction through an intuitive interface.

3. Methodology

3.1. Dataset

The application utilizes the Breast Cancer Wisconsin (Diagnostic) dataset from the UCI Machine Learning Repository. The dataset comprises 569 instances with 30 numerical features. Each instance is labeled as either malignant (M) or benign (B).

3.2. Data Preprocessing

- Loading Data
- Label Encoding
- Splitting Data
- Scaling using Min-Max normalization

3.3. Fuzzy Feature Engineering

Fuzzy logic is applied to selected features (radius1, area1, texture1). For each feature, three membership functions-low, medium, and high-are computed using triangular membership functions.

3.4. Model Training

A Decision Tree Classifier is employed with:

- Hill Climbing Search
- Grid Search

Optimal depth is selected based on performance.

3.5. Evaluation Metrics

- Accuracy
- Precision
- Recall
- Confusion Matrix
- Classification Report

4. Application Features

4.1. Data Exploration Tab

- Load Dataset
- Data Information
- Data Table

4.2. Visualizations Tab

- Visualization Options
- Feature Selection
- Plot Display

4.3. Model Training Tab

- Max Depth Selection
- Train Model
- Training Results

4.4. Evaluation Tab

- Evaluation Options
- Evaluate & Plot
- Evaluation Results

5. Results

After training, the model achieved high accuracy on the training set. Evaluation on the test set yielded satisfactory precision and recall metrics.

6. Conclusion

The developed application successfully integrates data preprocessing, fuzzy feature engineering, model training, and evaluation within a user-friendly GUI. It aids in the classification of breast cancer tumors, potentially assisting healthcare professionals in diagnostic processes.

7. Future Work

- Incorporate additional classification algorithms
- Implement cross-validation
- Enhance the GUI
- Deploy as a web application

References

- Dua, D. and Graff, C. (2019). UCI Machine Learning Repository.
- Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python.
- McKinney, W. (2010). Data Structures for Statistical Computing in Python.
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment.
- Waskom, M., et al. (2014). Seaborn: Statistical Data Visualization.