Breast Cancer Prediction Using Machine Learning

Satyanarayana Vinay Achanta (A02395874)

Introduction

In the intricate and evolving landscape of healthcare, early detection, and diagnosis of diseases like breast cancer are pivotal. Traditional diagnostic methods, while effective, often grapple with limitations in accuracy and efficiency. The "Breast Cancer Prediction" project is an ambitious endeavor that leverages advanced analytical techniques, including machine learning and deep learning, to transform the paradigm of breast cancer diagnostics. This project explores three innovative methodologies:

Breast Cancer Prediction using PyCaret: This method employs the PyCaret library to analyze and compare various machine learning models, with a specific focus on the impact of Principal Component Analysis (PCA).

Breast Cancer Prediction using XGBoost: Here, the powerful XGBoost algorithm is applied, renowned for its performance in classification tasks.

Breast Cancer Prediction using Deep Learning with Neural Networks: This approach utilizes the robust capabilities of neural networks, tapping into TensorFlow and Keras for advanced predictive modeling.

Kaggle Competition Link

Technical Analysis

Methodological Framework

1. **PyCaret Analysis:**

   Approach: Utilized the PyCaret library to benchmark various machine learning models. Analyzed performance with and without the implementation of Principal Component Analysis (PCA).

   Dataset: Employed the Breast Cancer Wisconsin (Diagnostic) dataset from Kaggle Competition, which comprises features like radius_mean, texture_mean, perimeter_mean, etc.

   Model Setup: Configured PyCaret with parameters such as normalization, feature selection, and session ID for reproducibility.

PCA Integration: Evaluated the impact of PCA on model performance by reducing dimensionality and focusing on principal features.

Models Compared: Included Logistic Regression, SVM, Ridge Classifier, Random Forest, and more, assessing metrics like accuracy, AUC, recall, and F1 score.

2. **XGBoost Application:**

Implementation: Applied the XGBoost algorithm, known for its efficiency in handling complex datasets.

Hyperparameter Tuning: Employed GridSearchCV for optimal tuning of parameters like max_depth, learning_rate, and n_estimators.

Evaluation Metrics: Focused on accuracy, precision, recall, and F1 score to assess model performance.

Data Handling: Preprocessing involved normalization using StandardScaler and handling missing values, followed by PCA for dimensionality reduction.

3. **Deep Learning Strategy:**

Framework: Developed using TensorFlow and Keras.

Model Architecture: Comprised of dense layers with activation functions and dropout layers to prevent overfitting.

Training and Evaluation: The model was trained on a split dataset and evaluated using accuracy, precision, recall, and F1 score metrics.

PCA Application: Like other methods, PCA was applied post-normalization for feature reduction.

Data Preprocessing and Management

- **Data Normalization:** Applied normalization techniques to scale the dataset features, enhancing model training efficiency and performance.
- **Handling Missing Values**: Implemented strategies to handle missing data points in the dataset, ensuring robustness in the model's input.
- **Training and Testing Split:** Divided the dataset into training and testing sets, with careful consideration of the distribution of diagnostic categories.
- **Feature Selection and Reduction:** PCA was utilized in specific models to concentrate on the most significant features by reducing data dimensionality.

Model Development and Architecture

- Machine Learning Models: Developed various machine learning models within the PyCaret framework, each with a specific architecture to handle binary classification tasks effectively.
- XGBoost: Configured the XGBoost model with parameters optimized for performance, focusing on constructing a sequence of decision trees for robust predictions.
- Neural Network in TensorFlow: The neural network consisted of multiple layers, including dense layers with different numbers of neurons and dropout rates, configured to capture complex patterns in the data effectively.

Results

PyCaret Method

- Without PCA: Logistic Regression excelled in accuracy among all tested models.
- With PCA: Enhanced results were observed in Logistic Regression, particularly in accuracy, precision, and F1 score.

XGBoost Technique

Model Performance: Post hyperparameter optimization, the XGBoost model showed high efficacy across various metrics like accuracy, precision, recall, and F1 score.

Deep Learning Approach

Model Efficacy: Exhibited remarkable performance with high accuracy, and balanced precision and recall.

Algorithm	Accuracy
PyCaret (Logistic Regression with PCA)	96.49
PyCaret (Logistic Regression without PCA)	94.74
XgBoost Classifier Model	97.36
Deep Learning Approach	98.24

Conclusion

In conclusion, the integration of PCA in machine learning models, alongside the deployment of sophisticated algorithms like XGBoost and neural networks, marked a significant advancement in predictive accuracy.