ML-AI Add-on: Final Assessment

Predicting Diabetes Status Using Classification Models

Context:

Diabetes is a chronic medical condition that can lead to serious health complications if not managed properly. Early prediction and diagnosis of diabetes are crucial for effective treatment and prevention of related complications. Healthcare professionals can benefit from predictive models that assess the risk of diabetes in patients based on their medical history and demographic information.

Objective:

The objective of this analysis is to build and compare various classification models to predict whether a patient has diabetes (Diabetes_Status = Positive) or not (Diabetes_Status = Negative) based on their medical history and demographic details. The goal is to identify the best-performing model that can be used by healthcare professionals for early detection and personalized treatment planning. Problem Statement: Given the Diabetes Prediction Dataset, which contains medical and demographic data of patients, including features such as age, gender, BMI, hypertension, heart disease, smoking history, HbA1c level, and blood glucose level, the task is to apply different classification algorithms to predict the likelihood of diabetes in patients. The models to be applied include: Logistic Regression, k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Naive Bayes, Decision Trees, Random Forests, MLP Classifier, XGBoost Classifier.

Link to Dataset:

https://raw.githubusercontent.com/tech4alltraining/aiml/refs/heads/main/datasets/classification/project/diabetes_prediction.csv

Tasks:

1. Understand the data and problem statement well, identify independent variables and dependent variable
2. Exploratory data analysis
    - Showing no of rows and columns
    - Info about data (columns and datatype)
    - Describe the data
3. Apply different preprocessing steps in dataset
    - Missing value handling (if applicable)
    - Drop unwanted columns (if applicable)
    - Remove duplicated rows (if applicable)
    - Drop Outliers (if applicable)
    - Scaling (if applicable)
    - Label Encoding (if applicable)
4. Model Selection and Training
    - Split the dataset into train data and test data

- Train each classifier on the training data: Logistic Regression, k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Naive Bayes, Decision Trees, Random Forests.
- Make predictions on the testing data using each classifier.

5. Performance Evaluation
   - Calculate and display the accuracy score for each classifier on the testing data.
   - Generate a classification report for each classifier, including precision, recall, and F1-score for each class.
   - Create a confusion matrix for each classifier and visualize.
6. Comparison and Conclusion:
   - Compare the performance of the classifiers based on accuracy, precision, recall, and F1-score.
   - Provide insights into which classifier performed best and why.