VIT®
BHOPAL
www.vitbhopal.ac.in

Project Report

(CSA2001)

Fundamentals in AI and ML

Slot – C11+C12+C13

Class ID: BL2025260100821

Student Name: Nihal Muhammed

Registration No.: 25BSA10061

Introduction

Diabetes is a major global health concern, and one of the leading global causes of death yearly. Often requires early diagnosis for effective intervention and management.

This project focuses on applying supervised machine learning techniques to develop a highly optimized predictive model capable of classifying patients as diabetic or non-diabetic based on the Pima Indians Diabetes Dataset.

The core objective is to create a robust model that maximizes the F1-Score, ensuring an optimal balance between minimizing critical missed diagnoses and reducing false alarms, which is vital for reliable screening tool.

Key Modules used:

- **Pandas & NumPy:** Core Modules for efficient data loading, cleaning and numerical operations.
- **Pipeline:** Chains all preprocessing and modelling steps into a single, cohesive workflow.
- **SimpleImputer & StandardScaler:** Modules responsible for handling missing data and normalizing features.
- **RandomForestClassifier:** The primary classification algorithm (an ensemble of decision trees).
- **GridSearchCV:** The optimization module responsible for automated hyperparameter tuning and cross-validation.
- **Metrics Modules:** Classes like f1_score, roc_auc_score and confusion_matrix used for comprehensive model evaluation.

Problem Statement

To develop a machine learning binary classification model using the Pima Indians Diabetes Dataset to accurately predict the onset of diabetes (Outcome = 1) based on eight numerical health indicators (e.g., Glucose, BMI, Insulin). The solution must implement a repeatable, robust data pipeline and be optimized to achieve the highest possible F1-Score, demonstrating reliable predictive capability.

Functional Requirements

1) Data Input & Loading:

– The system must read the pima_indians_diabetes_dataset.csv file and handle FileNotFoundError properly.

2) Data Preprocessing:

- Implementation of a Pipeline to manage sequential steps along with missing value imputation and feature scaling.

3) Prediction & Classification:

- Use Random Forest Classifier to predict the 'Outcome' (binary outcome).

4) Hyperparameter Tuning:

- Use GridSearchCV with cross validation (cv = 5) to systematically find the optimal model configuration out of the multiple possible combinations.

5) Reporting & Analytics:

- Output displaying core metrics (Accuracy, F1, auc-roc) and a detailed error analysis from the confusion matrix.

Non-Functional Requirements

1) Reliability:

- The model must be trained using cross-validation (cv = 5) to ensure the generalization error is stable across different subsets of the data.

2) Error Handling:

- The system must include try-except blocks for file loading and safely handle illogical zero values via imputation to maintain data integrity.

3) Maintainability:

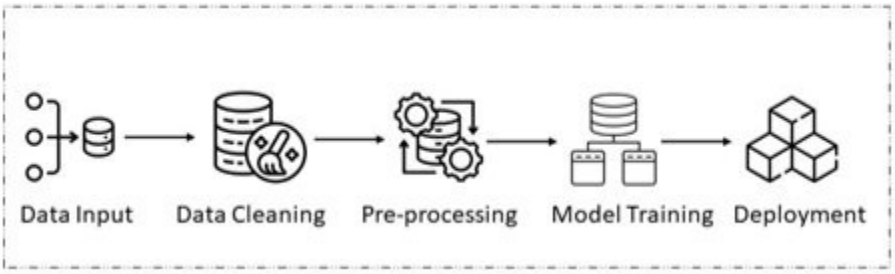
– The use of the Pipeline object ensures the preprocessing steps are modular and easily updated or replaced without altering the training workflow.

4) Performance:

- The training uses n_jobs = -1 in GridSearchCV to leverage all CPU cores, minimizing the total time required for hyperparameter optimization.

System Architecture & Design

The system follows a classic machine learning pipeline Architecture, ensuring data transformations are applied consistently and safely before modelling.



Key Components:

1) Data Ingestion: Reads the pima_indians_diabetes_dataset.csv

2) Preprocessing Pipeline: Sequential data transformation

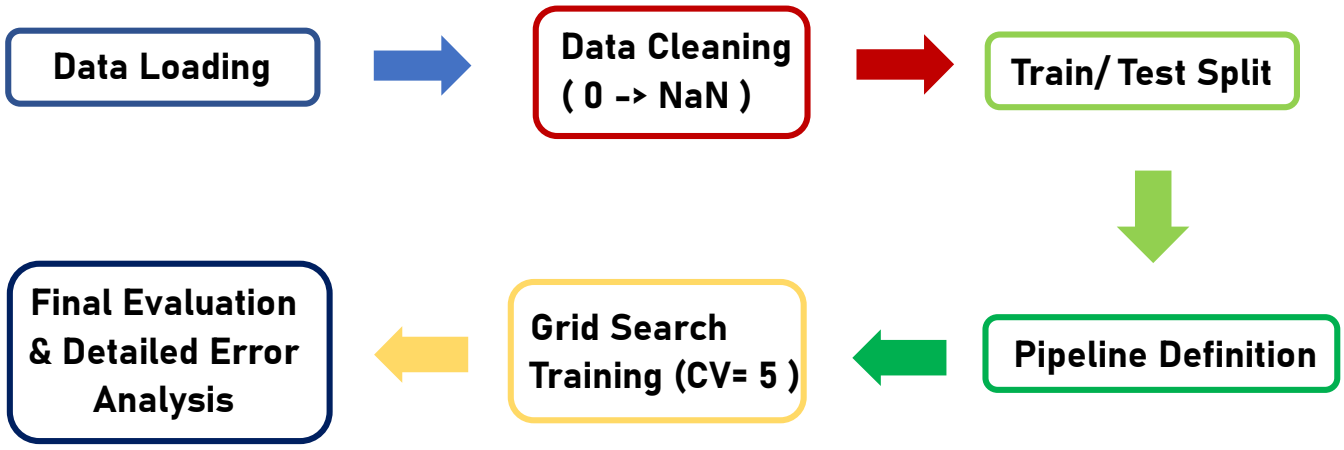
3) Model: Uses a Random Forest Classifier Model

4) Tuner: GridSearchCV manages the training iterations

5) Reporting: Output metrics and Error analysis

Process Flow (Workflow Diagram):

The program's logical workflow is strictly sequential:



Model Selection & Rationale

1) Dataset :

- **Pima Indians Diabetes**

- A well-known, numerical dataset for benchmarking binary classification enabling direct comparison with standard academic results.

2) Core Algorithm :

- **Random Forest Classifier**

- An Ensemble Model chosen for its high accuracy & robustness against data overfitting; effective on small/medium tabular datasets (like csv).

3) Preprocessing :

- **Pipeline w/ Imputer & Scaler**

- Standardizes the workflow and prevents data leakage by ensuring scaling factors are learned only from training data.

4) Tuning Method :

- **GridSearchCV**

- Systematically finds the best Hyperparameters, replacing manual testing with an exhaustive, cross-validated search.

5) Optimization Metric :

- **F1-Score (scoring = "f1")**

- Objective is to find the optimal balance between recall (catching diabetic patients) & precision (avoiding false alarms).

Implementation Details

The core implementation is contained in the python file, utilizing the scikit-learn framework for high-level Machine learning operations.

1) Data Preprocessing Flow:

The Pipeline ensures that following steps are executed:

a) Missing Value Imputation: SimpleImputer replaces illogical zero values with mean of training data

b) Feature Scaling: StandardScaler standardizes features to a mean of 0 and standard deviation of 1, preventing features with large numerical ranges from dominating the model.

c) Model Training: The preprocessed data is fed into the RandomForestClassifier Model for training it.

2) Hyperparameter Training:

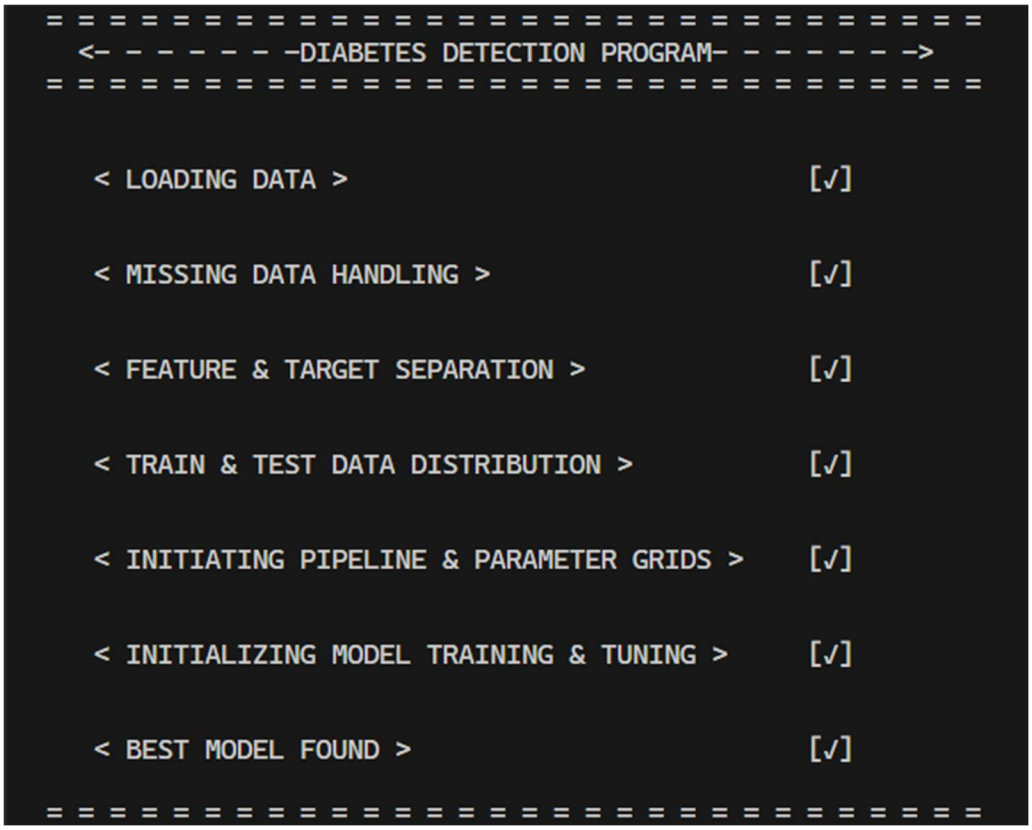
The model utilizes Grid Search with 5-Fold Cross-Validation to find optimal configuration that maximizes the f1-score.

Model Evaluation / Screenshots of Results

The model's performance metrics are derived from the execution of the final, best-tuned model on the unseen Test Set.

1) Loading Screen

Screen displaying the processes being executed by the program



2) Core Evaluation Metrics

The overall generalization and reliability of the final model:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    < MODEL EVALUATION (RANDOM FOREST CLASSIFICATION) >
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 • Best F1 Score Found (Tuning): 0.6139

 • Best Parameters:

     > model__max_depth : 15
     > model__min_samples_leaf : 2
     > model__n_estimators : 400

 • Test Set Accuracy: 74.03 %

 • Test Set F1 Score: 0.5946

 • Test Set AUC-ROC Score: 0.8261

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

3) Detailed Error Analysis

The model's performance breakdown on the 231 patients in the Test Set:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            < CONFUSION MATRIX ON TEST SET >
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

[[127  23]
 [ 37  44]]

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
             < CONFUSION MATRIX ANALYSIS >
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 • Total Samples Tested: 231

 • Total Actual Positive Cases (Diabetic): 81

 • Total Actual Negative Cases (Non-Diabetic): 150


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  - ERROR ANALYSIS -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 • Recall (Sensitivity): 0.5432
 * Model caught 54.32 % of all actual diabetic patients
 * Missed Cases: 37 [ !! CRITICAL ERROR !! ]

 • Precision: 0.6567
 * Model predicted positive correctly 65.67 % of the time
 * Falsely Predicted Diabetic: 23

 • Specificity: 0.8467
 * Model identified 84.67 % of all non-diabetic patients

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Testing Approach

The testing approach was built directly into the program structure:

1) Validation:

The use of GridSearchCV with cv = 5 serves as the primary internal validation, ensuring the optimal model configuration is robust and not specific to one arbitrary training split.

2) External Testing:

The model's performance is finally assessed on the independent 30% Test Set (x_test, y_test), providing an unbiased estimate of its real-world generalization ability.

Challenges Faced

1) Imbalance and Recall:

The primary challenge was the dataset's class imbalance (more non-diabetic patients) which resulted in low initial recall. Even optimizing for F1 struggled to overcome the tendency to predict the majority class (Non-Diabetic), leading to a high FN count.

2) Handling Illogical Zeros:

The necessity of converting illogical zeros (e.g., Insulin = 0) into NaN (Not A Number) and subsequently managing them with an Imputer required careful attention to avoid skewing the feature distribution.

Learnings & Key Takeaways

1) Importance of Pipeline:

The project proved the efficiency and safety of using the Pipeline in ensuring clean, non-leaking and reproducible preprocessing steps.

2) Metric Selection:

Optimizing for the F1-Score was proven effective, resulting in a balanced model with high precision and an ok recall. Future work can use the scoring = “recall” metric with careful threshold adjustments.

Future Enhancements

1) Cost-Sensitive Tuning: Adjust the scoring to “recall” or implement custom weighting with the Random Forest algorithm to explicitly penalize the costly False Negatives, driving recall above 75%

2) Threshold Adjustment: Apply post-tuning threshold adjustment (lowering the probability threshold from 0.5 to around 0.35) to immediately convert high-probability FNs into TPs.

3) Data Augmentation: Implement techniques like SMOT (Synthetic Minority Oversampling Technique) on the training set to address the class imbalance and provide the model with more examples of the minority (diabetic) class.

References

1) Python documentation:

https://docs.python.org/3/

2) scikit-learn:

https://www.geeksforgeeks.org/machine-learning/learning-model-building-scikit-learn-python-machine-learning-library/

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

3) Pima Indians Diabetes Dataset:

https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database