**NAAMII Task 2 – Methodology Report**

**Task Overview**

The task involved a tabular-data classification problem, requiring the training and evaluation of three machine learning models **Logistic Regression**, **Random Forest**, and **XGBoost** on provided datasets: training, test, and a blinded test set. The aim was to preprocess data effectively, train and tune the models using best practices, and generate probability-based predictions with performance metrics.

**1. Data Preprocessing & Feature Engineering**

**a. Dataset Overview**

* train\_set.csv, test\_set.csv, and blinded\_test\_set.csv were loaded and inspected.
* The target column was "CLASS" and identifier column was "ID".

**b. Data Cleaning**

* Infinite values (inf, -inf) were replaced with NaN.
* Missing values were imputed using **mean imputation**.

**c. Scaling & Transformation**

* All features were numeric; thus, no encoding was needed.
* A Pipeline using **StandardScaler** and **SimpleImputer** was applied to normalize the data.

**d. Feature Engineering**

* No new features were created since the dataset was already structured and numeric.
* Feature selection was not explicitly required as all numeric features were retained.

**2. Model Architectures & Hyperparameters**

Logistic Regression was implemented as the baseline model using the ‘liblinear’ solver, with the maximum number of iterations set to 1000. No additional regularization tuning was performed, as the goal was to establish a simple benchmark. For the Random Forest model, hyperparameter tuning was carried out using GridSearchCV with a 5-fold StratifiedKFold cross-validation strategy. The search space included variations in the number of estimators (set to 100), maximum tree depth (10 and 20), and minimum samples required to split an internal node (2 and 5). Similarly, the XGBoost model was optimized using GridSearchCV and 5-fold cross-validation. The tuning process considered the number of estimators (100), maximum depth of trees (3 and 5), and learning rate values (0.1 and 0.05). Additionally, the XGBoost classifier was configured with use\_label\_encoder set to False and eval\_metric specified as ‘logloss’ to avoid deprecated warnings and ensure consistent model evaluation.

**3. Cross-Validation Scheme**

* **5-Fold Stratified Cross-Validation** was used to ensure balanced class representation in each fold.
* Tuning was only done on the training set, and the test set was kept strictly for evaluation.

**4. Performance Metrics**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Dataset | Accuracy | AUROC | Sensitivity | Specificity | F1-Score |
| Logistic Regression | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Logistic Regression | Test | 0.5900 | 0.6466 | 0.4524 | 0.6897 | 0.4810 |
| Random Forest | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Random Forest | Test | 0.6300 | 0.6708 | 0.3095 | 0.8621 | 0.4127 |
| XGBoost | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| XGBoost | Test | 0.6100 | 0.6326 | 0.3571 | 0.7931 | 0.4348 |

**5. Probability Outputs**

Predicted class probabilities were generated for:

* train\_predictions.csv
* test\_predictions.csv
* blinded\_test\_predictions.csv

Each file contains:

* ID column
* Class\_0 and Class\_1 probability columns

**6. Strengths, Limitations & Future Work**

**Strengths:** Solid cross-validation and model tuning**,** Standardized preprocessing pipelineandClearly structured and reproducible codebase

**Limitations:** Feature interactions and domain-specific feature engineering were not exploredandEnsemble stacking/blending was not implemented due to time constraint

**Future Improvements:** Model ensembling for higher generalization can be applied**,** advanced feature selection (e.g., SHAP, permutation importance)**,** Apply SMOTE or class weighting if dataset is imbalancedandPerform full error analysis to improve sensitivity/specificity trade-offs.