NAAMI ASSIGNMENT - II

Author: Aakrit Dongol
Submission Date: 30 May 2025
Repository Link: *https://github.com/aakritd/naami-assignments*

1. Problem Understanding

This task involves a multi-class classification problem on tabular data. The objective is to develop models that predict class probabilities accurately using traditional machine learning algorithms, with a focus on generalization and performance across unseen datasets.

2. Tools and Technologies Used

Language:
Python

Platform:
Jupyter Notebook

Version Control:
Git & GitHub

Libraries Used:

- NumPy – numerical operations and array manipulations
- Pandas – data loading, manipulation, and preprocessing
- Matplotlib – data visualization
- Scikit-learn (sklearn) –
    - Model building: LogisticRegression, RandomForestClassifier
    - Data preprocessing: StandardScaler, SimpleImputer, SelectKBest, f_classif, PCA

    - Evaluation metrics: accuracy_score, classification_report, confusion_matrix, roc_auc_score

- XGBoost – gradient boosting classifier for robust classification tasks

- OS – to handle file system operations (e.g., creating directories)

3. Approach and Methodology

1. Data Preprocessing

- Dropping the records for which the data are not finite.
- Checking for the balance in the data based on the classes
- Standardization of the data to remove the bias
- Creation of two types of datasets. One with the original 3236 features (for Random Forest and XGBoost) and the other with 100 features (for Logistic Regression) using feature selection.

2. Model Implementation

For the classification task, following algorithms were used :

A. **Logistic Regression**

A LogisticRegression model from sklearn.linear_model was used as a baseline.The model was configured with:

- max_iter= 1000 to ensure convergence
- random_state=42 for reproducibility
- class_weight='balanced' to address class imbalance in the training data

The model was trained using the 100 best features selected through SelectKBest(f_classif).

After training, predictions (predict) and predicted probabilities (predict_proba) were generated for the test set.

B. **Random Forest Classifier**

A RandomForestClassifier model from sklearn.ensemble was used as a more complex baseline classifier. The model was configured with:

- n_estimators=100 to build 100 decision trees for robust ensemble learning.
- random_state=42 for reproducibility of results.

Trained on the full feature set (all 3000 features) without prior feature selection, leveraging the model's intrinsic feature importance mechanism to handle irrelevant features.

The model was trained on the training set and then used to generate predictions (predict) and predicted probabilities (predict_proba) on the test set.

C. **XGBoost Classifier**

An XGBClassifier from the xgboost library was used as a powerful gradient boosting model. The model was configured with:

- eval_metric='logloss' to optimize the logistic loss during training.
- random_state=42 for reproducibility.

The classifier was trained on the full training dataset (X_train, y_train) without prior feature selection. After training, predictions (predict) and predicted probabilities (predict_proba) were generated on the test set.

3. Model Evaluation

Confusion Matrices for all three algorithms are as follows :

For Logistic Regression

	Actual Positive	Actual Negative
Predicted Positive	**20**	**6**
Predicted Negative	**12**	**13**

For Random Forest

	Actual Positive	Actual Negative
Predicted Positive	**21**	**5**
Predicted Negative	**13**	**12**

For XGBoost

	Actual Positive	Actual Negative
Predicted Positive	**20**	**6**
Predicted Negative	**11**	**14**

Model`	Accuracy	Precision (class 0)	Recall (class0)	F1-Score (class 0)	Precision (class 1)	Recall (class 1)	F1-Score (class 1)	AUROC
Logistic Regression	**0.6471**	**0.62**	**0.77**	**0.69**	**0.68**	**0.52**	**0.59**	**0.6031**
Random Forest	**0.6471**	**0.62**	**0.81**	**0.70**	**0.71**	**0.48**	**0.57**	**0.6654**
XGBoost	**0.6667**	**0.65**	**0.77**	**0.70**	**0.70**	**0.56**	**0.62**	**0.6738**

Metrics	Logistic Regression	Random Forest	XGBoost
Sensitivity	0.52	0.48	0.56
Specificity	0.7692	0.8077	0.7692

Final predictions for the **blinded test set** were generated and saved in CSV format as required. The CSV file names for each of the algorithms Logistic Regression, Random Forest, and XGBoost are *'logreg_blind_preds.csv'*, *'rf_blind_preds.csv'* and *'xgb_blind_preds.csv'* respectively and are located in the folder *blind_test_results*.

4. Discussion and Conclusion

This assignment tackled a binary classification task using Logistic Regression, Random Forest, and XGBoost. XGBoost achieved the best performance with the highest accuracy (66.67%) and AUROC (0.6738), followed by Random Forest and Logistic Regression. While Logistic Regression offered better interpretability, ensemble methods like XGBoost and Random Forest captured more complex patterns, especially with the full feature set.

All models were evaluated using accuracy, AUROC, precision, recall, F1-score, sensitivity, and specificity. Median imputation was used for missing values, and class imbalance was addressed

using class weighting. Future improvements include hyperparameter tuning, advanced imputation methods, other feature engineering techniques, and ensemble stacking to further enhance performance and generalization on unseen data.