Appendix

Machine Learning Classifiers

Random Forest Classifier

The Random Forest (RF) classifier is an ensemble-based algorithm that aggregates the predictions of multiple decision trees to improve overall accuracy and reduce overfitting. Each tree $T_1, T_2, \ldots, T_M$ is trained on a randomly selected subset of the training data and a random subset of features. This diversity among the trees helps capture various patterns in the data. The final prediction $\hat{y}$ for an input feature vector x is obtained through majority voting across all trees:

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \ldots, T_M(x)) \tag{3}$$

where M is the total number of trees in the forest, and $T_m(x)$ denotes the class prediction from the m-th decision tree.

XGB Classifier

XGBoost (XGB) is a gradient boosting framework that builds additive tree models iteratively. Each new tree g_m minimizes the regularized loss function:

$$J(\theta) = \sum_{i=1}^{n} \ell(y_i, \hat{y_i}^m) + \sum_{m=1}^{M} \Omega(g_m) \tag{4}$$

where ℓ is the loss function between the true label y_i and the predicted value $\hat{y_i}^m$, and Ω represents the regularization to penalize complexity.

Support Vector Classifier

Support Vector Classifier (SVC) constructs a decision boundary that maximizes the margin between classes. The prediction function $\hat{y_s}$ is:

$$\hat{y_s} = sign(a^T x + c) \tag{5}$$

where a is the coefficient vector, c is the intercept, and x is the input vector. We utilized the RBF kernel to handle non-linear separability.

Logistic Regression Classifier

Logistic Regression (LR) is a linear classification algorithm that estimates class probabilities using the sigmoid activation function. For an input feature value i, the predicted probability $\hat{p}$ is computed as:

$$\hat{p} = \frac{1}{1 + e^{-(w^T i + q)}} \tag{6}$$

where w is the weight vector, and q is the bias term. This function maps real-valued input into a probability between 0 and 1.

K-Nearest Neighbors Classifier

The K-Nearest Neighbors (KNN) algorithm predicts the class $\hat{y}_k$ of an input sample $\boldsymbol{x}$ based on the majority vote of its k-nearest neighbors:

$$\hat{y}_k = mode(y_j : x_j \in N_k(x)) \tag{7}$$

where $N_k(x)$ is the set of the k-closest points to x in the training set.

Performance Metrics

The most critical aspect of evaluating machine learning models is the selection and interpretation of performance metrics. In this study, we used five evaluation metrics to compare model performance: Accuracy, Precision, Recall (Sensitivity), F1-Score, and Area Under the Curve (AUC). All these metrics are derived from the confusion matrix, which consists of four essential components: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). A True Positive (TP) refers to the number of correctly predicted positive instances. A False Positive (FP) represents the number of negative instances that were incorrectly predicted as positive. A True Negative (TN) is the number of correctly predicted negative instances, while a False Negative (FN) indicates the number of positive instances that were incorrectly predicted as negative.

Based on these components, the evaluation metrics are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{8}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{F1-Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{11}$$

Accuracy evaluates the overall correctness of the model by measuring the proportion of total correct predictions. **Precision** focuses on the reliability of positive predictions, which is particularly critical in scenarios where false positives carry high costs. **Recall** measures the model's ability to correctly identify actual positives, important when false negatives are especially undesirable. **F1-Score** is the harmonic means of precision and recall, offering a balanced measure when dealing with imbalanced classes.

AUC (Area Under the Curve) evaluates a model's ability to distinguish between classes by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) across all possible threshold values. While AUC is typically designed for binary classification tasks, our problem involves multiclass classification where AUC is not directly computed.

To adapt AUC for the multiclass context, we employed the One-vs-Rest (OvR) approach. In this method, AUC is calculated separately for each class by treating it as the positive class while combining all other classes as the negative class. This binary classification strategy is repeated for each class, and the resulting AUC scores are

averaged to provide an overall measure of model performance in distinguishing each class from the rest.

Formally, for a multiclass problem with K classes, the OvR AUC is defined as:

$$AUC_{OvR} = \frac{1}{K}\sum_{k=1}^{K} AUC(y_k, \hat{p}_k) \tag{12}$$

where:

- y_k is the binary indicator (1 or 0) for whether the true class is k,
- $\hat{p}_k$ is the predicted probability for class k,
- $AUC(y_k, \hat{p}_k)$ is the AUC score for class k versus all others.

Together, these metrics provide a comprehensive understanding of the model's performance, ensuring robust and reliable evaluation under various data conditions.

Feature Importance and Explainability

Understanding the rationale behind an ML model's prediction by identifying how input features contribute to the output is a crucial aspect of predictive analytics and model interpretability. Traditional ML models often rely on built-in feature selection methods or importance metrics such as information gain (IG) to quantify feature relevance. However, as Lundberg et al (2019) point out, gain-based feature importance measures can be inconsistent and misleading. To overcome these limitations, SHapley Additive exPlanations (SHAP) has emerged as a state-of-the-art local explainability technique. SHAP is grounded in cooperative game theory, originally introduced by Shapley (Shapley, 1953), and attributes the prediction of a model to each feature fairly distributing the output among the contributing features.

The SHAP value for a feature i is defined as:

$$\varphi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \tag{13}$$

where:

- N is the set of all features,
- S is a subset of features not containing i,
- $f(S)$ is the model prediction based only on the features in subset S,
- φ_i is the SHAP value for feature i, representing its marginal contribution.

This formulation allows SHAP to estimate the positive or negative contribution of each feature to a given prediction (local interpretability), while also aggregating these contributions to understand the feature's overall importance across all predictions (global interpretability).