Tabular-data Classification Pipeline

This report presents a methodology for binary classification on a high-dimensional dataset with limited sample size. The train dataset contains 3,238 numerical features with only 315 samples, presenting the classic "curse of dimensionality" challenge.

Data Preprocessing and Feature Engineering

• Comprehensive analysis of missing values, class distribution, outlier detection and skewness using bar charts & box plots.

• Infinite values conversion, missing value imputation using median to handle missing data robustly.

• Basic statistical assessment to understand data characteristics and identify quality issues.

• IQR method(factor=1.5) replaces outliers with median values, clips remaining extreme values to bounds.

• Yeo-Johnson transformation reduces skewness and approximates normal distributions for improved model performance (optional).

• “RobustScaler” for stable normalization, resilient to outliers and extremes.

• Skewness filter removes features with skewness >1.0 that negatively impact model performance (optional).

• Eliminate low-variance features(<0.01), typically reducing feature space by 50% (optional).

• Correlation filter removes highly correlated features (>0.95) while preserving information.

• SMOTE oversampling applied only to the training set, generates synthetic samples for class balance.

• SelectKBest with mutual information selects top K features capturing feature-target relationships.

• Features reduced to 10-50 features maintaining a favorable sample-to-feature ratio.

Model Architectures and Hyperparameters: Logistic regression serves as a baseline with hyperparameters including maximum iterations, regularization strength(C), l1 and l2 regularizations options, solver, and class_weight options for handling imbalance. Decision tree implementation uses either gini or entropy to measure split purity, maximum depth of the tree, minimum samples required to split an internal node, minimum number of samples required to be at a leaf node, and number of features to consider when looking for a split. Random forest

ensemble method employs estimators, constrained maximum depth of the tree and other hyperparameters similar to that of a decision tree.

Cross-Validation and Evaluation: StratifiedKFold cross-validation maintains class distribution across folds ensuring balanced evaluation for binary classification tasks. Randomized search replaces computationally expensive grid search, operating under reduced parameter space for efficient hyperparameter tuning. Each iteration generates a random set of hyperparameters. ROC-AUC serves as the primary evaluation metric due to threshold independence, supplemented by accuracy, recall, specificity, and F1-score for comprehensive model assessment. Strict train-test data separation prevents information leakage and provides unbiased performance estimates across all experimental configurations.

Results: The outcomes of training and test sets for each classifier are provided below.

	Random Forest		Decision Tree		Logistic Regression	
Metric	Train set	Test set	Train set	Test set	Train set	Test Set
Accuracy	0.679	0.65	0.707	0.76	0.685	0.640
AUC ROC	0.686	0.652	0.711	0.756	0.678	0.640
Sensitivity	0.717	0.666	0.725	0.738	0.645	0.642
Specificity	0.654	0.637	0.696	0.775	0.712	0.637
F1-score	0.637	0.615	0.661	0.720	0.617	0.600

Strengths, Limitations and Future Improvements: The above pipeline aims to improve generalization. It is clear that a strong pipeline can reduce overfitting. The pre-processing and feature selection processes improved performance up to 60-70% on all measures. The results reveal that there is not a significant difference in performance between each model on the training and testing sets.

However, the small sample size (315) fundamentally limits generalization capability despite aggressive feature reduction from the original 3,238 features. Aggressive filtering may inadvertently remove features with complex interactions or non-linear target relationships. So, the filters must be carefully applied. In the meantime, careful outlier and noise analysis would significantly improve the model's generalization capability. So, future improvements should focus on data collection and advanced feature selection methods. Additionally, implementation of gradient boosting algorithms, neural networks and increasing sample size to 1,000+ represents potential areas for achieving superior performance.