Journal of IoT in Social, Mobile, Analytics, and Cloud (ISSN: 2582-1369)
www.irojournals.com/iroismac/

An Explainable AutoML Pipeline for Multi-Task Tabular Data Using Optuna and SHAP

Abarna S.[1], Rakshitha A.[2], Shanmathi K.[3], Yogesh G.[4]

[1]Assistant Professor, [2,3,4]Student, Department of Artificial Intelligence and Data Science, Velalar College of Engineering and Technology, Erode, India.

E-mail: [1]abikavinvcet@gmail.com, [2]rakshi6578@gmail.com, [3]shanmathiakshatha@gmail.com, [4]yogi007575@gmail.com

Abstract

Developing interpretable models on structured tabular data is difficult because there is a sequence dependency among data pre-processing, model learning, hyperparameter tuning, and result interpretation. The existing AutoML systems focus more on predictive accuracy rather than having an integrated approach towards explainability for different types of learning tasks. In this research, we introduce an Explainable AutoML Pipeline for performing automated data pre-processing, multi-model learning, hyperparameter optimization using Optuna, model selection using a leaderboard, and finally interpreting the results using SHAP. The framework automatically performs preprocessing on the input data, optimizes various candidate models, and determines the most suitable learning algorithm without any human intervention while also offering explanations about the decision-making process on a per-feature level. Model validation is then done by applying it to the California Housing and Titanic datasets. In particular, the optimized LightGBM model scored 0.8470 on the regression R^2 metric, while the optimized XGBoost classifier managed to reach 81.56% accuracy, an F1-score of 0.7481, and an AUC of 0.8117. The SHAP analysis successfully detected the most influential predictive features, which increases the model interpretability.

Keywords - SHapley Additive exPlanations (SHAP), Hyperparameter Optimization, Automated Machine Learning (AutoML), Multi-Task Tabular Learning, Explainable Artificial Intelligence (XAI).

Journal of IoT in Social, Mobile, Analytics, and Cloud, September 2026, Volume 8, Issue 3, Pages 292-307
DOI: https://doi.org/10.36548/jismac.2026.3.007
Extended version of article received from **International Conference on Smart Innovations in Engineering Technology (ICSIET 2026)**
Received: 10.06.2026, received in revised form: 13.07.2026, accepted: 30.07.2026, published: 12.08.2026

1. Introduction

With the increased use of data-driven decision-making techniques in industries such as healthcare, financial services, manufacturing, cybersecurity, and business analytics, there has been a need for efficient machine learning methods that can effectively analyze the structured tabular data. As tabular data is one of the largest forms of data in industrial and organizational settings, it has become necessary to utilize predictive models that are effective and reliable for decision-making purposes. However, traditional machine learning approaches have several processes that are manual in nature, such as data preprocessing, feature engineering, predictive model selection, hyperparameter tuning, and performance assessment. This makes the entire process of creating predictive models complex, time-consuming, and highly dependent on expert knowledge.

The current improvements in Automated Machine Learning (AutoML) have made model building easier by automating the selection of algorithms and their hyperparameters, which increases the efficiency of learning [1], [6]. Explainable Artificial Intelligence (XAI) has further improved these frameworks by explaining model predictions using interpretable techniques like SHAP [2], [3]. Moreover, Optuna Optimization and Tree-based Ensemble models have proven to perform better on structured tabular data due to efficient search space and accurate predictions [5], [7], [12]. However, there are still a lot of domain-specific AutoML frameworks that solve either classification or regression challenges. Also, there is still no unified framework that integrates adaptive preprocessing, multiple models' comparison, automatic model selection, and XAI in one pipeline [4], [8] – [11].

To overcome the mentioned challenges, we propose a novel framework for Explainable AutoML pipeline for multi-task tabular data. The proposed work involves automated pre-processing, adaptive feature engineering, hyperparameter optimization using Optuna, multi-model learning, leaderboard-based model selection and SHAP-based explainability. The proposed framework automatically determines the type of prediction problem and configures the learning process for both classification and regression tasks with minimal user input. Through the combination of intelligent automation and model interpretability, this framework aims at improving the reproducibility, scalability and reliability of predictions in diverse applications of tabular data.

2. Related Works

Machine Learning (ML) has emerged as an essential technique for performing predictive analysis on tabular data in different domains like healthcare, financial services, cyber security, and industrial systems. However, designing a high-performing machine learning model on tabular data is usually associated with a lot of effort involving extensive preprocessing, feature engineering, model design and hyper-parameter tuning, and therefore, being time-consuming and highly expert-dependent. Research has shown that AutoML can greatly simplify the above steps not only reducing the human involvement but also improving the prediction accuracy due to smart selection and optimization of the models [1], [6]. Despite the mentioned progress, many available AutoML tools focus only on accurate predictions.

To increase the model interpretability, XAI methods have started to be used together with the AutoML systems. It has been found out by recent researches that combining automatic creation of models and explainability allows people to understand the contribution of features while retaining competitive predictive abilities [2], [3]. In addition, it has been discovered that SHAP explains the importance of features more consistently and globally than traditional explanation approaches which makes it fit for important decisions [8], [9]. Explainable learning has proved to be an efficient method of verifying the reliability of predictions in software engineering and other domains [10]. However, there are many domain-dependent AutoML-XAI approaches that cannot be generalized to any tabular learning problem.

The role of hyperparameters optimization has gained importance in the current era of AutoML as it is known to have a strong impact on model performance. Methods such as Bayesian Optimization using tools like Optuna have been proven to be more efficient than traditional optimization algorithms as they help identify the best parameter settings with minimum number of trials [1], [5]. Alongside this, benchmarking frameworks for AutoML have made the process of evaluating various machine learning automation frameworks much easier [6]. However, current optimization methods usually concentrate on boosting prediction accuracy and do not consider automated explanations.

Learning algorithm selection is another factor in analyzing tabular data. Empirical research has proven that gradient boosting and decision tree ensembles yield significantly better results compared to deep learning models when working with structured tabular data due to modeling heterogeneous interactions and distribution [7]. XGBoost, CatBoost, and LightGBM ensemble

approaches have proved to perform well in complex classification cases while being computationally efficient [12]. Moreover, there are new advanced preprocessing approaches that include feature selection and synthetic data generation to solve class imbalance issues [11]. While these approaches help to optimize different stages of machine learning pipeline, they do not provide an integrated solution that would automatically determine the task, preprocess, optimize, select a model, and analyze it for both classification and regression cases.

Despite the successful integration of AutoML and explainable AI in the recent literature, current approaches are often domain-specific such as metabolomics [2], software engineering [3], or educational settings [4], [10]. The emphasis in existing AutoML frameworks is on hyperparameters tuning and model accuracy whereas automatic adaptation of the pipeline to classification and regression tasks is rather limited [1], [5], [6]. Moreover, many researchers focus on evaluating specific machine learning models or ensembles without the implementation of a model selection process based on the leaderboard [5], [7], [12]. Despite the popularization of SHAP in model explanation, the combination of automated preprocessing, Optuna-based hyperparameter tuning, multiple model comparisons, and task-oriented pipeline design for generic tabular data is underexplored [3], [8], [9], [11]. Thus, the development of a unified explainable AutoML system that will be able to automatically detect prediction task, perform adaptive pre-processing, tune multiple models, select the best one using leaderboard evaluations and produce explanations seems essential for tabular data problems.

3. Proposed Methodology

3.1 Overall Framework

The proposed Explainable AutoML framework is aimed at automation of the whole machine learning pipeline for tabular datasets along with increasing the transparency of models and minimizing human involvement. As shown in Figure 1, the framework operates following a sequence of actions that include data preprocessing, multi-model learning, hyperparameter tuning, evaluation, and explanation analysis.

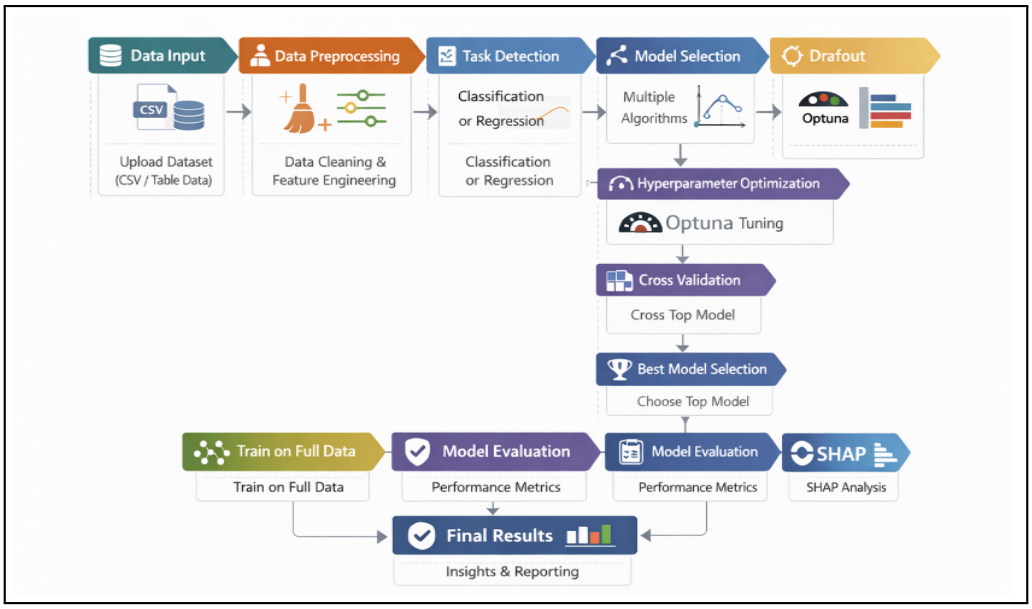

Figure 1. Workflow of the proposed explainable autoML framework

The input dataset is preprocessed and then feature engineered to guarantee the quality of the data before configuring the input automatically in the right way for the right prediction task. Then multiple machine learning models are trained and optimized using Optuna optimization to find the best configuration of the hyperparameters. The optimized models are validated using cross-validation and ranked according to the leaderboard approach. In the end, the best-performing model is chosen for predictions. In order to enhance the interpretability of the model, SHAP explainability is applied for the best-performing model in order to measure the effect of each feature on the output both globally and locally.

3.2 Dataset Description

To validate the proposed Explainable AutoML framework, two public datasets were used, which include regression and binary classification tasks. For regression, the California Housing benchmark was chosen, where the task is to predict median house values from demographic and geographic features [13]. To validate the binary classification ability, the Titanic benchmark was chosen, where the task is to predict survival based on demographic and travel features [14]. A split of 80:20 between train and test sets was used in order to maintain consistency in model training and performance evaluation. The properties of the benchmark datasets are provided in Table 1.

Table 1. Characteristics of the benchmark datasets

Parameter	**California Housing Dataset [13]**	**Titanic Dataset [14]**
Learning Task	Regression	Binary Classification
Data Type	Structured Tabular	Structured Tabular
Source	Scikit-learn Repository	Kaggle Titanic Dataset
Total Instances	20,640	891
Total Features	8	11
Numerical Features	8	5
Categorical Features	0	6
Target Type	Continuous	Binary (0, 1)
Class Distribution	Not Applicable	Survived: 342 Not-Survived: 549
Training Samples (80%)	16,512	712
Testing Samples (20%)	4,128	179

3.3 Automated Data Processing

The proposed architecture is designed for enabling automated data processing to enhance data quality and ensure a uniform input for training the models. At the beginning, input dataset profiling is performed to detect the presence of missing values, duplicates, data type and distribution of the features. The duplicates are eliminated to avoid redundancy, and missing values are filled with median imputation to maintain the statistical properties of the numerical features. Categorical features are encoded to provide numerical representation compatible with the machine learning models.

In order to increase the quality of the features, outliers are detected by means of IQR approach, and observations falling below the lower bound and above the upper bound are clipped. The lower and upper bounds are computed as

$$Q_{\text{Lower}} = Q_1 - 1.5 \times IQR \tag{1}$$

$$Q_{\text{Upper}} = Q_3 + 1.5 \times IQR \tag{2}$$

where,

$$IQR = Q_3 - Q_1 \tag{3}$$

Q_1 and Q_3 denote the first and third quartiles, respectively.

Features exhibiting high positive skewness are normalized using the logarithmic transformation to improve feature distribution and learning stability. The transformation is expressed as

$$x' = \log(1 + x) \tag{4}$$

where x represents the original feature value and x'denotes the transformed feature.

3.4 Multi-Model Learning and Hyperparameter Optimization

After the completion of data preprocessing, the resulting training dataset is employed to build several candidate machine learning models, for solving both classification and regression problems. The presented framework employs LightGBM, XGBoost, CatBoost, Random Forest, Extra Trees, Decision Tree, Logistic Regression, Ridge Regression, and Support Vector Machine, representing various aspects of the machine learning capabilities on structured tabular data. Each model is independently trained to allow a comprehensive evaluation of its predictive performance.

The hyperparameter optimization procedure is performed with the use of Optuna framework employing the Tree-structured Parzen Estimator (TPE) sampler to optimize over the predefined hyperparameter space. The optimization objective is formulated as

$$\theta^* = \arg\max_{\theta} f(\theta) \tag{5}$$

where θ represents the set of hyperparameters, $f(\theta)$ denotes the objective function evaluated through cross-validation, and θ^* corresponds to the optimal hyperparameter configuration

The selected hyperparameters are then used to train each candidate model, with consistent learning occurring through optimal hyperparameters. This process makes sure that the model has generalizability and reduces the human effort required in tuning hyperparameters in the traditional way.

3.5 Leaderboard-Based Model Selection and Performance Evaluation

The optimized candidate models are systematically evaluated and ranked using a leaderboard-based strategy to identify the most effective predictive model. The optimized candidate models are systematically

evaluated on the independent testing dataset and ranked according to their predictive performance. The model achieving the highest task-specific evaluation score is selected as the final prediction model for subsequent explainability analysis.

The selected model is further evaluated on the testing dataset using standard performance metrics appropriate to the prediction task.

For classification, the evaluation metrics are computed as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

For regression, the performance is quantified using

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{10}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{12}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively; y_i represents the actual value, $\widehat{y}_i$ denotes the predicted value, $\bar{y}$ is the mean of the observed values, and n is the total number of samples. The highest-performing model identified through this evaluation process is subsequently utilized for explainability analysis.

3.6 SHAP-Based Explainability

In order to enhance the level of transparency of the selected model for predictions, SHAP (SHapley Additive exPlanations) is used to measure the impact that individual features have on each prediction made. After selecting the model, SHAP values are calculated in order to interpret both global and local explanations, which allow comprehensively interpreting the decision-making process of the selected model. Global explanation is achieved using feature importance ranking and visualization, while the local explanation demonstrates how individual features impact a particular prediction.

The SHAP value for a feature is computed as

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

where ϕ_i denotes the contribution of feature i, F represents the complete set of input features, S is a subset of features excluding i, and $f(\cdot)$ denotes the prediction function of the trained model.

Algorithm 1: Explainable autoML pipeline for multi-task tabular data

Input: Tabular dataset D

Output: Optimized prediction model M^*, performance metrics, and SHAP explanations

Step 1: Load dataset D

Step 2: Perform data profiling

Step 3: Remove duplicate records

Step 4: Impute missing values using median imputation

Step 5: Encode categorical features

Step 6: Detect and treat outliers using the IQR method

Step 7: Apply logarithmic transformation to skewed features

Step 8: Remove near-constant features

Step 9: Split the processed dataset into training and testing sets (80:20)

Step 10: Detect the prediction task (Classification or Regression)

Step 11: Initialize candidate models:

{LightGBM, XGBoost, CatBoost, Random Forest, Extra Trees,
Decision Tree, Logistic Regression,Ridge Regression, SVM}

Step 12: **for** each model M_i **do**

Optimize hyperparameters using Optuna

Perform cross-validation

Train the optimized model

Evaluate model performance

Step 13: **end for**

Step 14: Rank all optimized models using the leaderboard

Step 15: Select the best-performing model M^*

Step 16: Evaluate M^* using the testing dataset

Step 17: Compute SHAP values for M^*

Step 18: Generate global and local feature explanations

Step 19: Produce the final prediction report and save M^*

Step 20: **Return** M^*, evaluation metrics, and SHAP explanations

4. Results and Discussion

4.1 Experimental Setup

The proposed Explainable AutoML model was implemented using Python programming language and was tested using two publicly available datasets that are used for performing regression and binary classification tasks. Specifically, the California Housing dataset was used to measure the efficiency of the proposed model in regression, while the Titanic dataset was used for measuring the efficiency in binary classification. All the experiments were conducted using an 80:20 train/test split, where the former was used for optimization and the latter was used for performance evaluation. Hyperparameters were optimized using the Optuna framework with cross-validation.

Table 2. Experimental configuration

Parameter	**Configuration**
Programming Language	Python 3.x
Development Environment	Jupyter Notebook
Machine Learning Library	Scikit-learn
Gradient Boosting Libraries	LightGBM, XGBoost, CatBoost
Hyperparameter Optimization	Optuna (TPE Sampler)
Explainability Method	SHAP
Validation Strategy	Cross-Validation
Train-Test Split	80:20
Regression Dataset	California Housing
Classification Dataset	Titanic

The experimental setup (in Table 2) included a wide array of machine learning models, the optimized machine learning models were then ranked based on a leader board approach, from which the most optimal one was chosen.

4.2 Regression Performance Analysis

The regression performance of the proposed Explainable AutoML framework was analyzed based on the California Housing dataset for the prediction of median house values. Several candidate models were created and optimized through the proposed pipeline, and a predictive analysis was conducted using the leaderboard-based approach. The comparison involved different algorithms such as gradient boosting, ensemble, linear, and support vector regressions. The comparative analysis has identified the most appropriate algorithm for structured tabular regression.

Table 3. Regression model performance on the california housing dataset

Rank	Model	CV Mean (R^2)	CV Std	Test R^2	RMSE	MAE
1	LightGBM	0.8402	0.0041	0.8470	44,777.6	29,803.4
2	CatBoost	0.8315	0.0054	0.8270	47,618.8	31,644.4
3	XGBoost	0.8228	0.0040	0.8142	49,345.3	33,219.8
4	Random Forest	0.8096	0.0055	0.8081	50,140.6	32,631.9
5	Extra Trees	0.7633	0.0075	0.7560	56,540.7	37,972.5
6	Decision Tree	0.7314	0.0074	0.7361	58,810.4	38,089.9
7	Ridge Regression	0.6744	0.0126	0.6185	70,700.5	51,303.8
8	Support Vector Regression (SVR)	0.0509	0.0059	0.0791	109,852.2	80,133.9

Table 3 shows that the gradient boosting algorithms always demonstrated better regression performance than other machine learning techniques. According to the leaderboard, the LightGBM algorithm had the best result, which suggests the efficiency of the method for capturing non-linear correlations in the California housing data. CatBoost and XGBoost algorithms also provided good results, thereby supporting the idea that ensemble boosting is an effective technique for structured tabular regression.

In order to make the model more interpretable, a SHAP analysis was done in order to measure how important each feature is in the prediction process. The SHAP beeswarm plot is presented in Figure 2 as an illustration of the importance and distribution of the contribution of each feature.

SHAP analysis shows that the most important predictors of house price prediction are the Median Income followed by the geographic features of latitude and longitude. Predictors with greater absolute SHAP

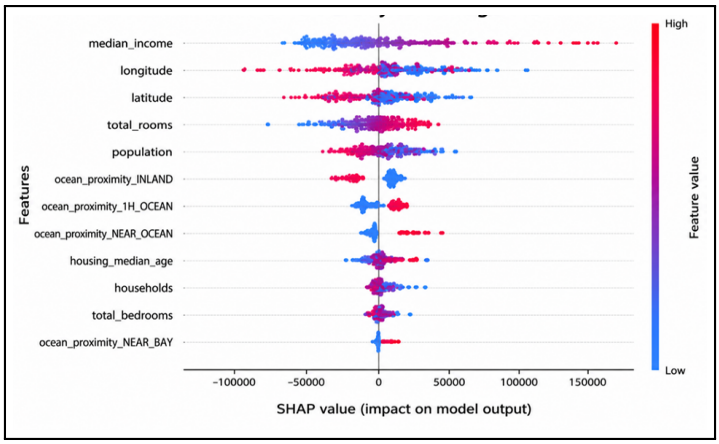

Figure 2. SHAP beeswarm plot for california housing (LightGBM)

values play a major role in the prediction model compared to those with lower SHAP values. The SHAP values distribution shows that the optimized LightGBM model is able to identify significant interactions between socioeconomic and geographic features, resulting in interpretable regression predictions.

4.3 Classification Performance Analysis

Comparative analysis includes tree ensemble models, linear classifiers, and support vector machines in order to select the best performing model in the case of binary classification. Performance comparison of all the models is given in Table 4. According to the leaderboard, the XGBoost model produced the best classification results after optimization, thus proving the efficiency of learning nonlinear dependencies from the Titanic dataset. The LightGBM and CatBoost models performed well and demonstrated the effectiveness of using gradient boosting algorithms to solve structured classification challenges.

Table 4. Classification model performance on the titanic dataset

Rank	Model	CV Mean (F1)	CV Std	Accuracy	F1 Score	AUC
1	XGBoost	0.7556	0.0338	0.8156	0.7481	0.8117
2	LightGBM	0.7634	0.0326	0.8045	0.7368	0.8074
3	Random Forest	0.7549	0.0357	0.7933	0.7338	0.8282
4	Logistic Regression	0.7350	0.0273	0.7877	0.7246	0.8209
5	CatBoost	0.7521	0.0362	0.7933	0.7176	0.8270
6	Extra Trees	0.7335	0.0502	0.7877	0.7077	0.8248
7	Support Vector Classifier (SVC)	0.7403	0.0448	0.7821	0.6880	0.8248
8	Decision Tree	0.7111	0.0271	0.7654	0.6441	0.8133

In order to improve the clarity of the selected classifier, the SHAP approach has been utilized to understand the significance of individual features during the prediction process. The SHAP feature importance plot shown in Figure 3 shows that sex, fare, age, and passenger class (Pclass) are the significant features influencing the prediction of survival. Higher SHAP values are used to denote the significance of the features while lower significance is denoted by lower SHAP values. It is clear that the optimized classifier has captured all the significant demographic and socioeconomic features.

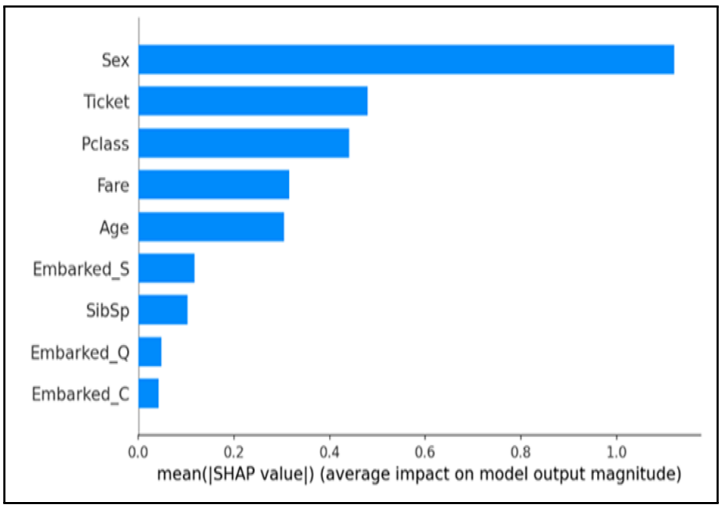

Figure 3. SHAP feature importance of the optimized XGBoost model for the Titanic dataset

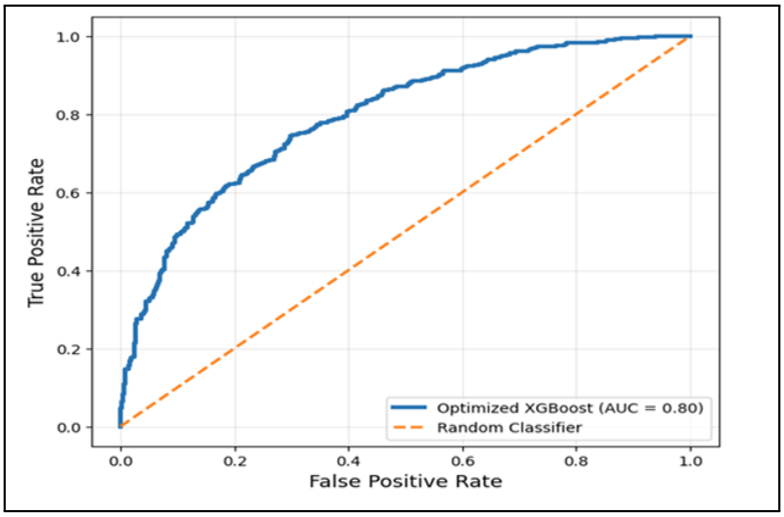

Figure 4. ROC curve of the optimized XGBoost model for the Titanic dataset

Further, the discriminative ability of the optimized classifier is exemplified through the ROC curve as depicted in Figure 4, showing that the optimized classifier exhibits good classifying capability by achieving an optimal balance between the true positive and false positive rates. This validates the reliability of the optimized classifier in classifying the two classes consistently.

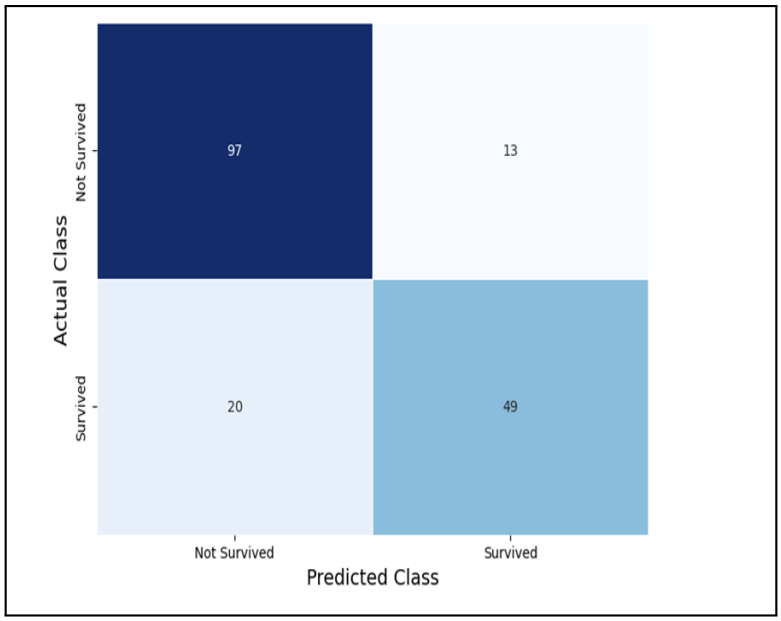

Figure 5. Confusion matrix for binary survival prediction on the Titanic dataset

The confusion matrix illustrated in Figure 5 shows that the optimal classifier can identify most of the survival and non-survival instances while resulting in relatively few errors. The ratio of the number of true positive instances, true negative instances, false positive instances, and false negative instances proves the balanced nature of predictions by the selecting the classifier and validating the selection.

5. Conclusion

This research study has designed an Explainable AutoML Pipeline for Multi-Task Tabular Data by incorporating automated data preprocessing, multi-model training, Optuna hyperparameter tuning, leaderboard-driven model selection, and SHAP-driven explainability into a single framework. The proposed pipeline successfully automated the entire process of model building for both regression and binary classification problems, thus minimizing manual efforts and increasing the transparency of models. Experimentation with the proposed framework on the California Housing and Titanic datasets illustrated the efficiency of the proposed framework as well, where the optimized LightGBM model produced an R^2 score of 0.8470 for the regression problem, and the optimized XGBoost classifier produced an accuracy of 81.56%, an F1-score of 0.7481, and an AUC of 0.8117. In addition, SHAP-driven explainability revealed the most prominent predictive features, thus making explanations transparent and reliable. Therefore, it can be concluded that the proposed framework provides a useful and reproducible tool for solving the multi-task tabular learning problems.

References

[1] Khan, Muhammad Salman, Tianbo Peng, Hanzlah Akhlaq, and Muhammad Adeel Khan. "Comparative Analysis of Automated Machine Learning for Hyperparameter Optimization and Explainable Artificial Intelligence Models." IEEE Access 2025, vol. 13: 84966-84991.

[2] Bifarin, Olatomiwa O., and Facundo M. Fernández. "Automated Machine Learning and Explainable AI (AutoML-XAI) for Metabolomics: Improving Cancer Diagnostics." Journal of the American Society for Mass Spectrometry 2024, vol. 35, no. 6: 1089-1100.

[3] Strudwick, James, Laura-Jayne Gardiner, Kate Denning-James, Niina Haiminen, Ashley Evans, Jennifer Kelly, Matthew Madgwick et al. "AutoXAI4Omics: An Automated Explainable AI Tool for Omics and Tabular Data." Briefings in Bioinformatics 2025, vol. 26, no. 1: bbae593.

[4] Cañete-Sifuentes, Leonardo, Victor Robles, Ernestina Menasalvas, and Raul Monroy. "Comparing Automated Machine Learning Against an Off-the-Shelf Pattern-based Classifier in a Class Imbalance Problem: Predicting University Dropout." IEEE Access 2023, vol. 11: 139147-139156.

[5] Kirubavathi, G., Dhanyatha Shree CA, and Nithish Kumar. "Enhanced DNS Cybersecurity: Optimization of XGBoost using Automated Hyperparameter Tuning via Optuna." In 2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC), IEEE, 2024: 1-7.

[6] Gijsbers, Pieter, Marcos LP Bueno, Stefan Coors, Erin LeDell, Sébastien Poirier, Janek Thomas, Bernd Bischl, and Joaquin Vanschoren. "Amlb: An Automl Benchmark." Journal of Machine Learning Research 2024, vol. 25, no. 101: 1-65.

[7] Grinsztajn, Léo, Edouard Oyallon, and Gaël Varoquaux. "Why do tree-based models still outperform deep learning on typical tabular data." Advances in Neural Information Processing Systems 2022, vol. 35: 507-520.

[8] Roshinta, Trisna Ari, and Szűcs Gábor. "A Comparative Study of Lime and Shap for Enhancing Trustworthiness and Efficiency in Explainable AI Systems." In 2024 IEEE International Conference on Computing (ICOCO), IEEE, 2024: 134-139.

[9] Rao, Sannidhi, Shikha Mehta, Shreya Kulkarni, Harshal Dalvi, Neha Katre, and Meera Narvekar. "A Study of LIME and SHAP Model Explainers for Autonomous Disease Predictions." In 2022 IEEE Bombay Section Signature Conference (IBSSC), IEEE, 2022: 1-6.

[10] Gezici, Bahar, and Ayça Kolukisa Tarhan. "Explainable AI for Software Defect Prediction with Gradient Boosting Classifier." In 2022 7th International Conference on Computer Science and Engineering (UBMK), IEEE, 2022: 1-6.

[11] Abraham, Asha, Habeeb Shaik Mohideen, and R. Kayalvizhi. "A Tabular Variational Auto Encoder-based Hybrid Model for Imbalanced Data Classification with Feature Selection." IEEE Access 2023, vol. 11: 122760-122771.

[12] Jaiswal, Sushma, and Priyanka Gupta. "Ensemble approach: XGBoost, CATBoost, and LightGBM for Diabetes Mellitus Risk Prediction." In 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), IEEE, 2022: 1-6.

[13] Scikit-learn Developers, "California Housing Dataset," Scikit-learn Documentation. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html

[14] Kaggle, "Titanic: Machine Learning from Disaster," Dataset. Available: https://www.kaggle.com/competitions/titanic