Journal of Artificial Intelligence Practice (2026)
Clausius Scientific Press, Canada

DOI: 10.23977/jaip.2026.090113
ISSN 2371-8412 Vol. 9 Num. 1

# *Interpretable Machine Learning in Enzyme Classification: A SHAP-Guided Analysis of Global Structural Features*

**Donghan Li**

*Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215123, China*

***Keywords:*** R language; Database; Machine learning; Bioinformatics; Shiny Web; Protein prediction

***Abstract:*** Accurate computational annotation of protein function addresses a critical bottleneck in bioinformatics. This study presents an explainable machine learning framework that predicts a protein's functional class from its physicochemical profile. A Random Forest model, trained on four structural descriptors from the Protein Data Bank, classifies proteins into three major enzyme classes: Hydrolase, Oxidoreductase, and Transferase. The model achieved robust performance (test accuracy: 71.42%), with SHAP analysis identifying molecular weight as the primary discriminative feature. To ensure practical utility and reproducibility, the model is deployed as an interactive Shiny application and an open-source R package, providing a reliable and accessible tool for the community.

## 1. Introduction

Proteins are central to virtually all biological processes, including enzymatic catalysis, signal transduction, molecular transport, and structural organization. Consequently, accurate protein function annotation is a core objective in molecular biology and bioinformatics [1]. Although advances in high-throughput sequencing and structural biology have led to a rapid expansion of available protein data, experimental functional characterization has not kept pace. This widening gap underscores the need for efficient and scalable computational approaches for protein functional classification [2].

Protein function is closely linked to structural and physicochemical properties. Beyond primary amino acid sequences, quantitative descriptors such as molecular weight, residue composition, hydrophobicity, and secondary structure content play a critical role in determining protein behavior and biological activity [3]. Public resources, notably the Protein Data Bank (PDB), curate protein structural information and derived features, enabling systematic investigation of structure–function relationships using computational methods. These tabular, feature-rich datasets are particularly well suited for machine learning–based modeling.

Machine learning approaches have demonstrated strong performance in protein function prediction by capturing complex, nonlinear relationships among multiple input features. Unlike traditional rule-based or alignment-driven methods, ML models learn discriminative patterns directly from data with minimal prior assumptions [4]. Ensemble learning methods, including Random Forest and Extreme

Gradient Boosting, are especially effective for structured biological datasets due to their robustness, ability to handle high-dimensional features, and strong predictive performance. These characteristics make them appropriate choices for classification tasks based on physicochemical protein descriptors.

Despite their accuracy, ensemble ML models often suffer from limited interpretability. In biological research, understanding how specific features influence predictive outcomes is essential for building trust, supporting hypothesis generation, and facilitating biological validation. Models that lack transparency risk being treated as black boxes, thereby limiting their scientific utility. As a result, explainable artificial intelligence (XAI) techniques have become increasingly important in protein-related ML studies.

Among XAI methods, SHapley Additive exPlanations (SHAP) has emerged as a widely adopted framework for interpreting complex ML models. Based on cooperative game theory, SHAP quantifies the contribution of each feature to model predictions relative to a baseline, providing both global and sample-level explanations [5]. In protein function classification, SHAP analysis enables biologically meaningful interpretation by highlighting key physicochemical or structural features that drive predictive decisions.

In addition to predictive performance and interpretability, reproducibility and accessibility are essential in modern computational biology. Interactive web applications address these challenges by allowing users to explore datasets and apply trained models without extensive programming expertise. The R Shiny framework provides a flexible platform for deploying bioinformatics databases and prediction tools through user-friendly interfaces [6].

In this study, an explainable and reproducible machine learning framework is presented for protein functional classification using physicochemical features derived from protein structural data. Random Forest and XGBoost classifiers are trained and evaluated using standard classification metrics, while SHAP analysis is employed to interpret model predictions. To enhance usability, the trained models are further integrated into an interactive Shiny web application and an accompanying R package, facilitating transparent, reproducible, and accessible protein function prediction.

## 2. Materials & Methods

### 2.1 Data Collection and Preprocessing

The overall computational workflow adopted in this study is illustrated in Figure 1. The "Structural Protein Sequences" dataset from Kaggle was selected for this project. Specifically, the file pdb_data_no_dups.csv is used as the primary data source. The goal was to classify protein functions based on their physicochemical properties. Based on the dataset documentation, four key features are selected that describe the intrinsic properties of proteins, avoiding experimental variables like crystallization temperature to prevent data leakage. The selected features are:

-Structure Molecular Weight:

○The mass of the protein structure.

-Residue Count:

○The number of residues, indicating the size of the polymer chain.

-Density:

○A crystallographic parameter for packing density.

-pH Value:

○The acidity or alkalinity of the protein environment.


Figure 1: The Overall Computational Workflow for Protein Functional Classification, Illustrating Data Preprocessing, Model Training, and Evaluation Steps.

Data cleaning was performed in R (v4.5.2). The top three most frequent protein classes (Hydrolase, Transferase, and Oxidoreductase) were first identified and filtered the dataset to keep only these categories. Then, by eliminating rows containing missing values (drop_na) and rectifying ambiguous labels such as “null” strings within the original data file, the dataset was prepared for training. The cleaned data was saved as final_protein_data.csv.

## 2.2 Model Development and Hyperparameter Optimization

All machine learning tasks were implemented using the caret (v 7.0-1) and randomForest(v 4.7-1.2) packages. To ensure correct evaluation, the data was split into a training set of 70% and a testing set of 30% using stratified random sampling, createDataPartition. This method ensures that the proportion of the three protein classes remains consistent in both sets. A fixed random seed 17604 is set to make sure the results are reproducible.

The Random Forest algorithm is chosen for classification because of its robustness and ability to handle non-linear relationships. The model was trained with the following configurations:

-Resampling: 5-fold Cross-Validation is used to estimate model performance and prevent overfitting.

-Ensemble Size: The model consists of 1,000 decision trees (ntree = 1000) to ensure stability.

-Parallel Computing: To speed up the training process, the doParallel (v1.0.17) package was utilized to run computations on multiple CPU cores with detectCores() - 2.

## 2.3 Model Interpretation Methods

Two methods were used to explain how the model makes predictions, avoiding the "black box" problem:

- Gini Importance: This metric measures how much each feature contributes to the purity of the decision trees.

- SHAP Analysis: The fastshap library was used to calculate SHAP values. This allows for the observation of not just which feature is important, but how specific values of a feature influence the prediction for each class.

# 3. Results

## 3.1 Dataset Characteristics

Before modeling, the distribution of the selected features was analyzed. As shown in Figure 2, the molecular weights of the proteins show a long-tail distribution.

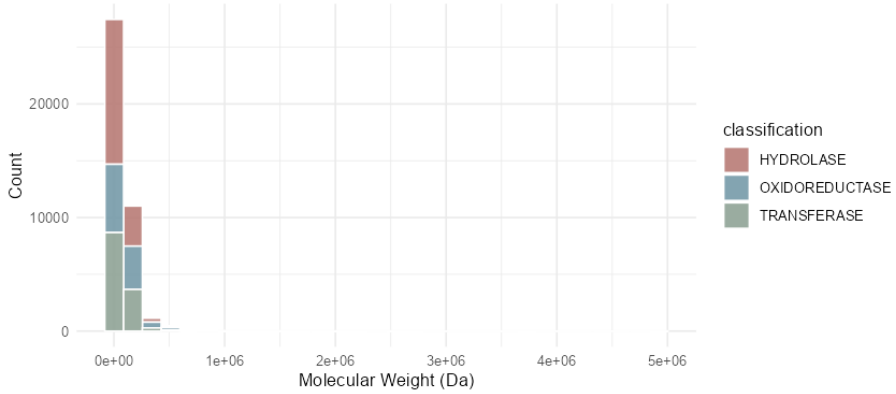


Figure 2: Distribution of Structure Molecular Weight (Da) across the Three Selected Protein Functional Classes.

Figure 3 displays the relationship between Residue Count and Matthews Density. Some clustering patterns among the three functional classes, which suggests that these physical properties can indeed help distinguish different protein types.

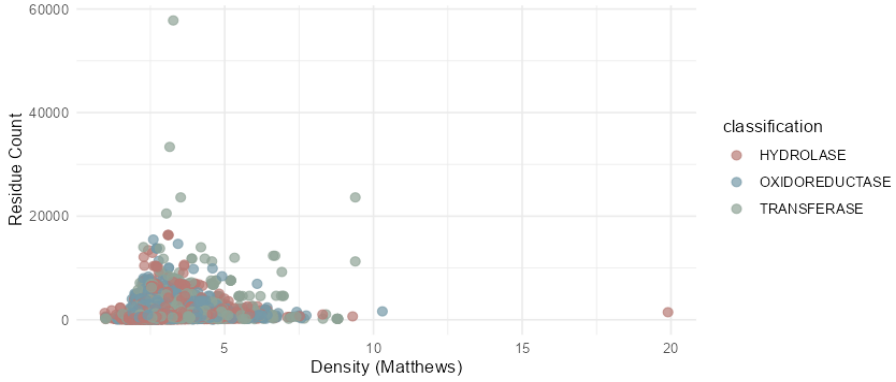


Figure 3: Scatter Plot Illustrating the Relationship between Residue Count and Matthews Density for Hydrolase, Oxidoreductase, and Transferase.

## 3.2 Random Forest Model Performance

The trained Random Forest model achieved a final accuracy of 0.7142 on the independent test set. The Multi-class AUC was also calculated, which was 0.8796, indicating a strong predictive performance. To further evaluate the model, the multi-class problem was treated as "One-vs-Rest" binary problems and plotted the performance curves, generated using ROCR (v1.0-11) and pROC (v1.19.0.1) packages:

ROC Curves (Figure 4): The curves for all three classes, Hydrolase, Oxidoreductase, Transferase, are well above the diagonal line, showing good sensitivity and specificity.

Precision-Recall Curves (Figure 5): These curves confirm that the model maintains reasonable precision even as recall increases, which is important for the imbalanced dataset.

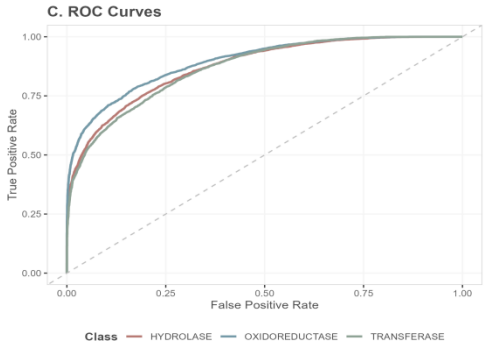


Figure 4: ROC Curves for the Random Forest Multi-Class Prediction.

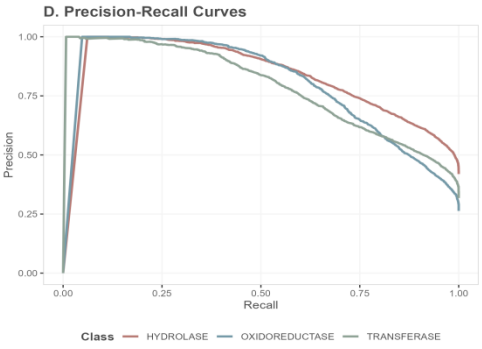


Figure 5: PR Curves Evaluating the Random Forest Model across the Three Functional Classes.

## 3.3 Feature Contribution and Interpretability

The features driving the model's decisions were analyzed, as summarized in Figure 6 and Figure 7.

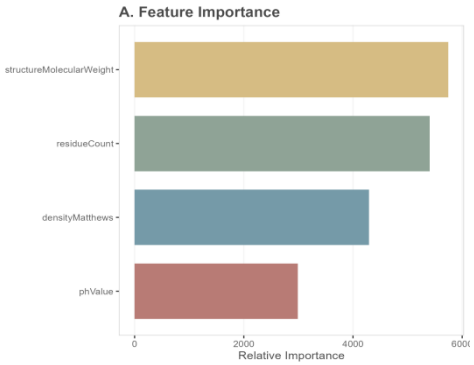


Figure 6: Feature Importance Ranking Derived from the Random Forest Model Using Gini Impurity.

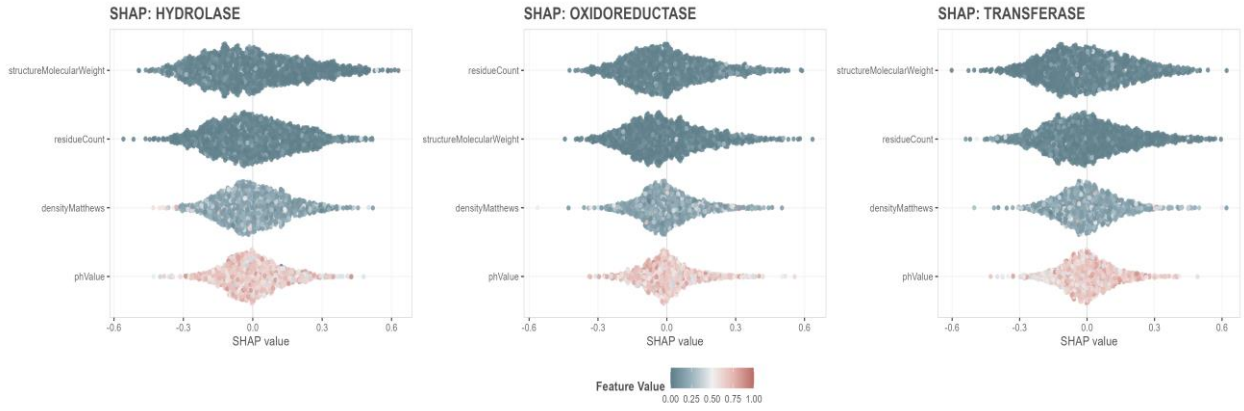


Figure 7: SHAP Summary Plots Illustrating the Global Feature Contributions for the Hydrolase, Oxidoreductase, and Transferase Classes.

Feature Importance (Figure 6): The results show that Structure Molecular Weight and Residue Count are the most important predictors. This makes biological sense, as the size and weight of a protein are fundamental to its function. The pH value had the lowest contribution

SHAP Analysis (Figure 7): The SHAP summary plots provide deeper insights. For example, specific ranges of Matthews Density show different impacts across the three classes. This confirms that the model is learning distinct biological patterns for Hydrolases, Transferases, and Oxidoreductases, rather than just memorizing data.

### 3.4 Model Robustness and Validity

To ensure the reliability of the classifier beyond standard metrics, the stability of the model was examined across the cross-validation folds. The consistent performance metrics observed across all 5 folds demonstrate that the model is robust and not overfitting to specific data subsets. Furthermore, the internal validity of the model is supported by the Feature Importance analysis as shown in Figure 6, where intrinsic structural properties like structureMolecularWeight significantly outweigh environmental factors like phValue. This confirms that the algorithm relies on stable physicochemical signatures rather than experimental noise.

Quantitatively, the ROC curves, Figure 4, exhibit a steep initial ascent for all three functional classes-Hydrolase, Oxidoreductase, and Transferase-indicating uniformly high sensitivity without class-specific bias. Moreover, the Precision-Recall curves, Figure 5, reveal that precision remains near 1.0 for the top-ranked predictions (Recall < 0.25) across all groups. This specific behavior suggests that while the model handles broad classification well, it is exceptionally reliable for high-confidence predictions. The high Multi-class AUC (0.8796) serves as definitive evidence of the model's discriminative power, ensuring it can function as a reliable backend engine for downstream applications regardless of deployment interfaces.

## 4. Web usage

This section provides a structured, step-by-step guide to the architecture, functionality, and usage of the *ProteinFunc Predictor* web application. Figure 8 are referenced at each critical step to visually illustrate user interactions, interface layout, and system feedback, ensuring clarity, usability, and reproducibility.

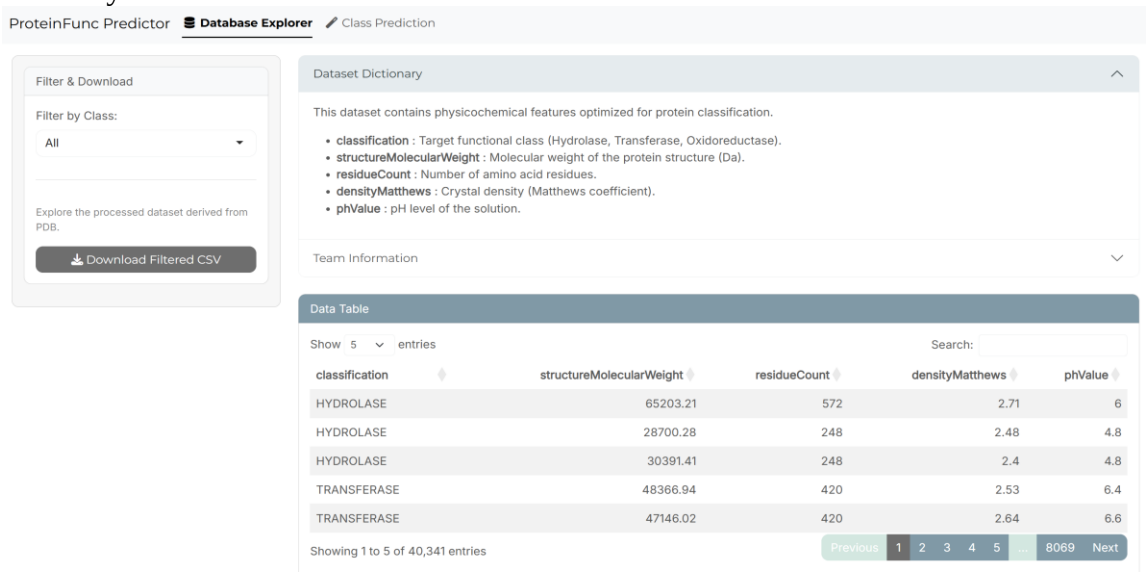


Figure 8: Landing Page – Full Application Overview.

## 4.1 Application Architecture and Design Philosophy

The ProteinFunc Predictor is a web-based bioinformatics application developed using the R Shiny framework, designed to transform static protein datasets into an interactive, machine learning–driven analytical environment. As shown in Screenshot 1, the application adopts a clean single-page layout with a persistent top navigation bar, enabling seamless navigation between modules without page reloads.

The user interface is built on the Bootstrap 5 framework with a customized Minty theme inspired by a Morandi-style color palette. Desaturated, low-contrast tones (e.g., muted blues and greens) are used to reduce visual fatigue during prolonged analytical use, while accent colors consistently highlight primary interactive elements.

Architecturally, the application consists of two core modules: Database Explorer and Class Prediction. Backend logic is implemented using reactive programming to ensure synchronized updates across tables, plots, and outputs. For computationally intensive operations, asynchronous processing is supported via the waiter package, maintaining interface responsiveness and a smooth user experience.

## 4.2 Module 1: Database Explorer

The Database Explorer module enables interactive inspection, filtering, and visualization of the curated protein dataset. As illustrated in Figure 9, a *sidebarLayout* structure separates user control from analytical outputs, ensuring consistent spatial organization and reduced cognitive load.

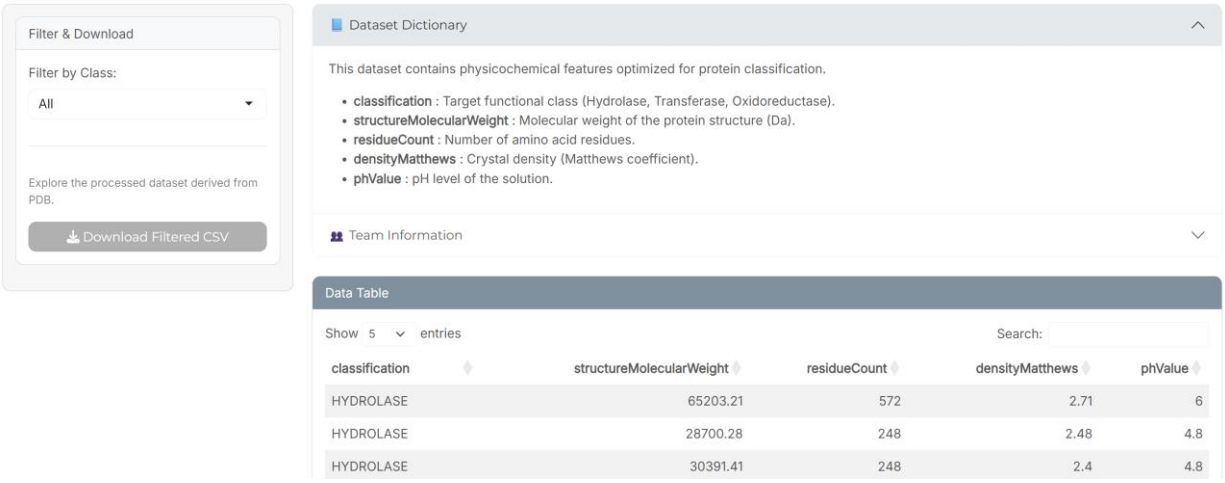


Figure 9: Database Explorer – Sidebar and Main Panel Layout.

### 4.2.1 Data Filtering and Accessibility

The sidebar hosts the primary data control widgets. Users can subset the dataset via the “Filter by Class” dropdown menu, with selections triggering immediate reactive updates across tables and visualizations. Figure 10 highlights this control as the central mechanism for defining the analytical context.

A “Download Filtered CSV” button is positioned beneath the filter controls, allowing users to export the currently displayed subset. This design ensures consistency between on-screen visualization and exported data, supporting reproducibility and external analysis.

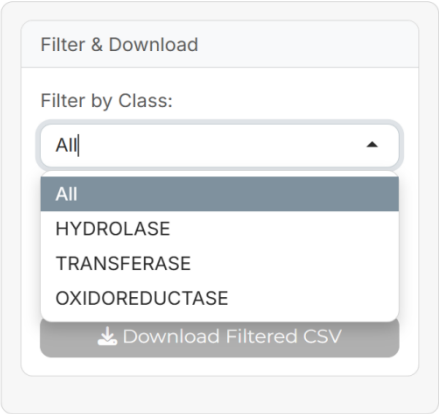


Figure 10: Filter Dropdown Highlighted.

### 4.2.2 Hierarchical Information Display

To manage information density, the main panel employs an Accordion component (Figure 11). The Dataset Dictionary panel is expanded by default to provide definitions of the four physicochemical features used in model training, ensuring interpretability before analysis.

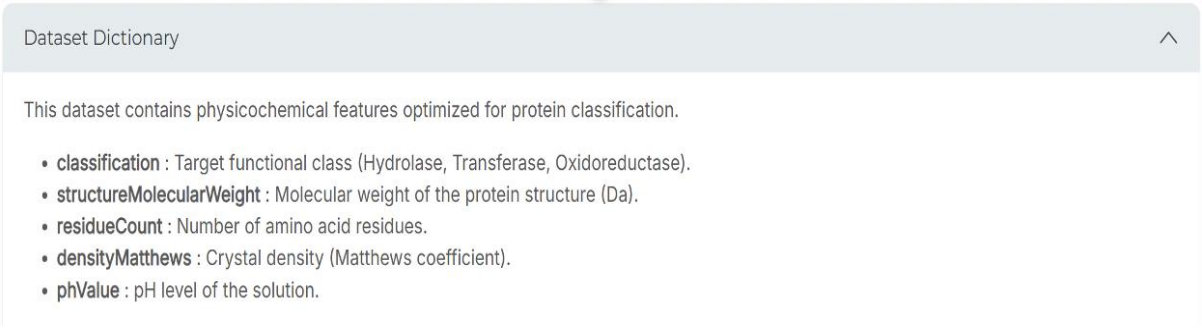


Figure 11: Accordion Panel.

### 4.2.3 Interactive Visualization and Model Performance

Filtered data are displayed using an interactive data table supporting pagination, sorting, and horizontal scrolling (Figure 12). Two ggplot2-based exploratory visualizations-a molecular weight distribution histogram and a residue count versus density scatter plot-are dynamically generated and color-coded by functional class.

At the bottom of the module, a consolidated evaluation panel integrates both interpretability and performance assessment (Figure 13). SHAP analyses for the three enzyme classes (Panel B) are presented first, illustrating the relative importance and directional effects of key physicochemical features. Below this, static ROC and Precision–Recall curves summarize overall classification performance from model training. This layout encourages users to interpret feature contributions prior to evaluating predictive reliability.

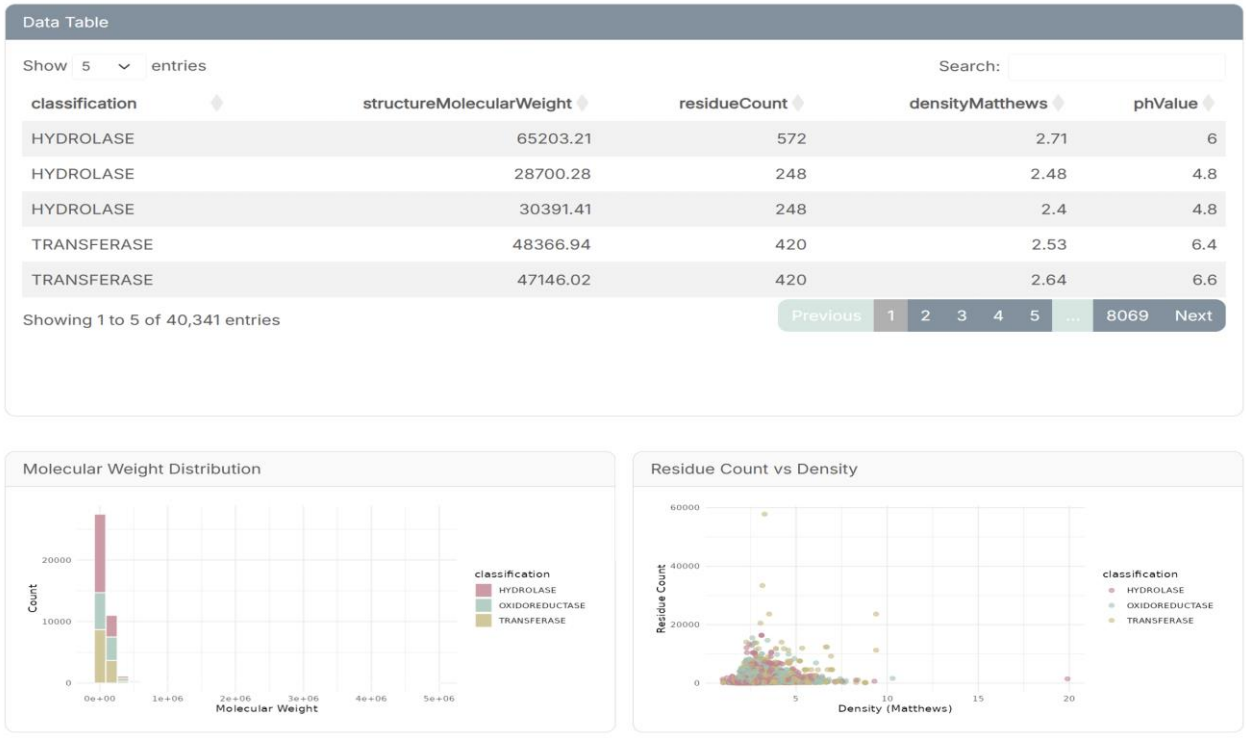


Figure 12: Data Table and Exploratory Plots.

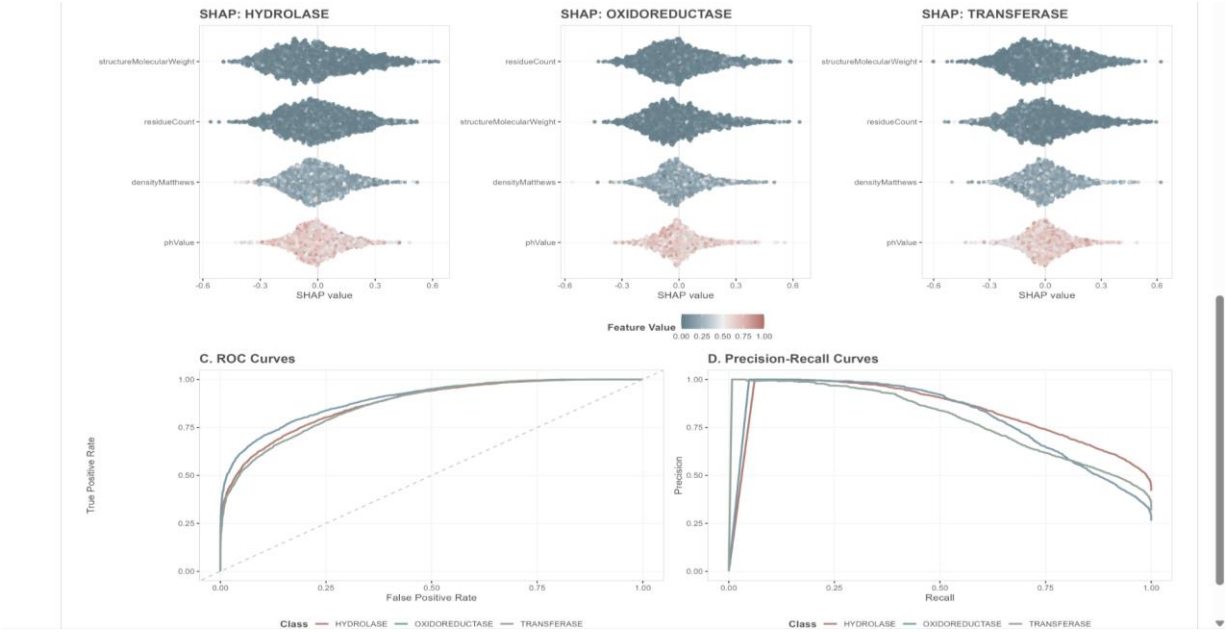


Figure 13: Embedded ROC and PRC Curves and SHAP.

## 4.3 Module 2: Class Prediction

The Class Prediction module converts the trained Random Forest model into an interactive inference tool. As shown in Figure 14, users can select between two prediction modes via radio buttons, ensuring flexibility for both batch and single-entry use cases.

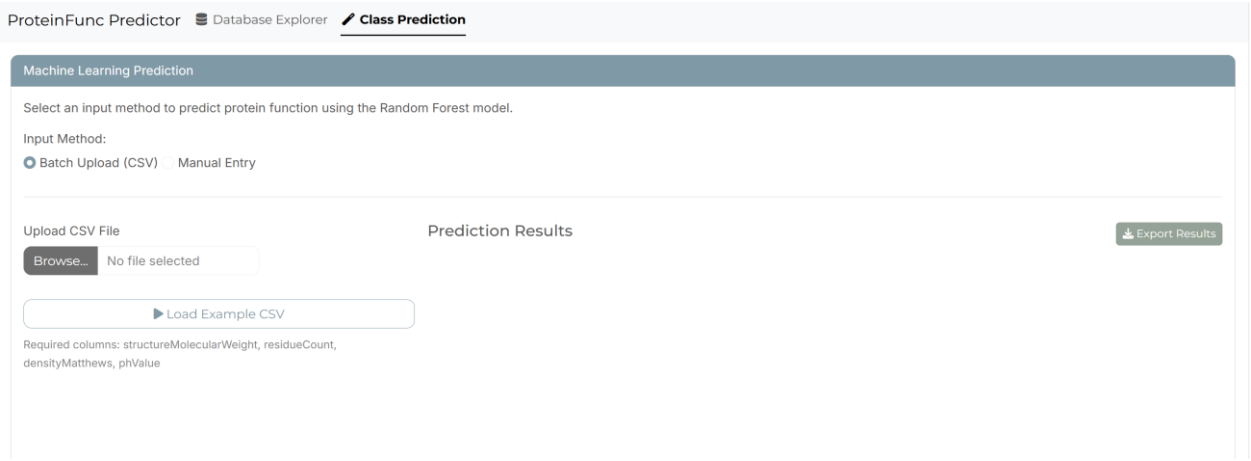


Figure 14: Class Prediction Module – Mode Selection.

### 4.3.1 Mode A: Batch Upload (CSV)

In Batch Upload mode, users submit external datasets in CSV format. Upon upload, backend validation checks confirm the presence and data types of the required features. If errors are detected, user-friendly notifications are displayed (Figure 15), preventing application crashes.

An "Load Example CSV" button is included to guide first-time users, demonstrating correct input structure without requiring prior data preparation.

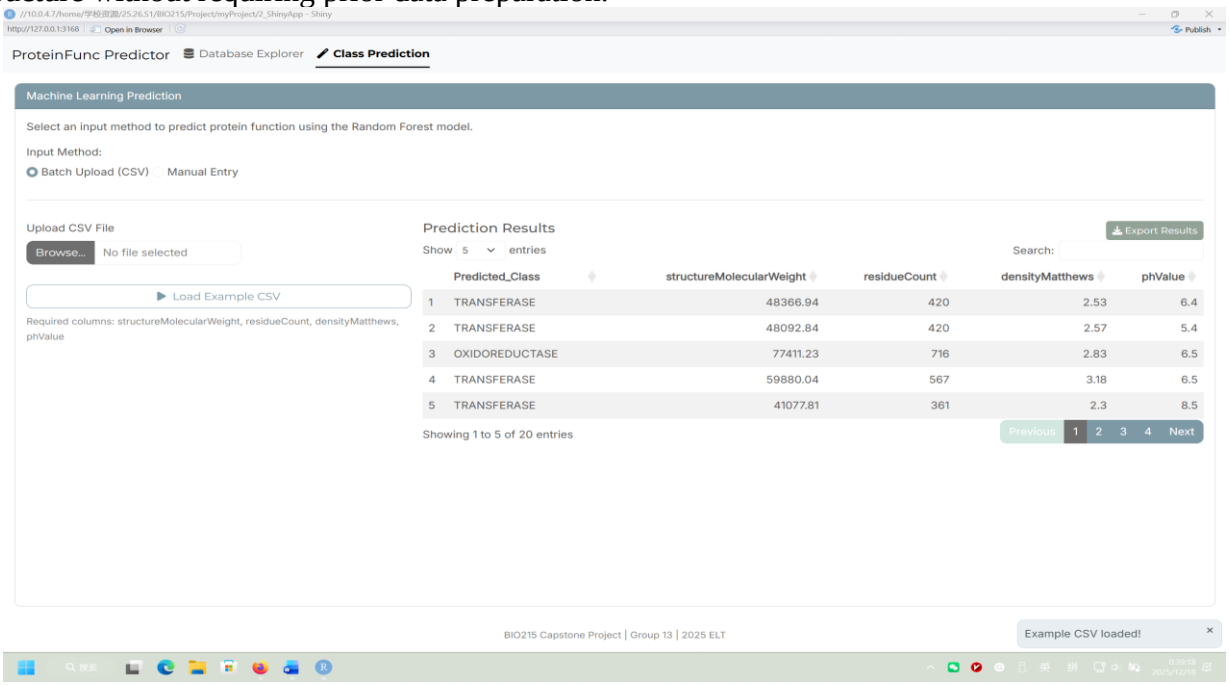


Figure 15: File Upload and Validation Message.

### 4.3.2 Mode B: Manual Entry

For quick queries, Manual Entry mode provides four numeric input fields. A "Fill Example Data" button pre-populates valid values, facilitating immediate testing during demonstrations or assessments. When "Run Prediction" is clicked, a loading spinner indicates processing, followed by a color-coded alert box displaying the predicted class (Figure 16). This clear visual feedback ensures results are intuitive and unambiguous.

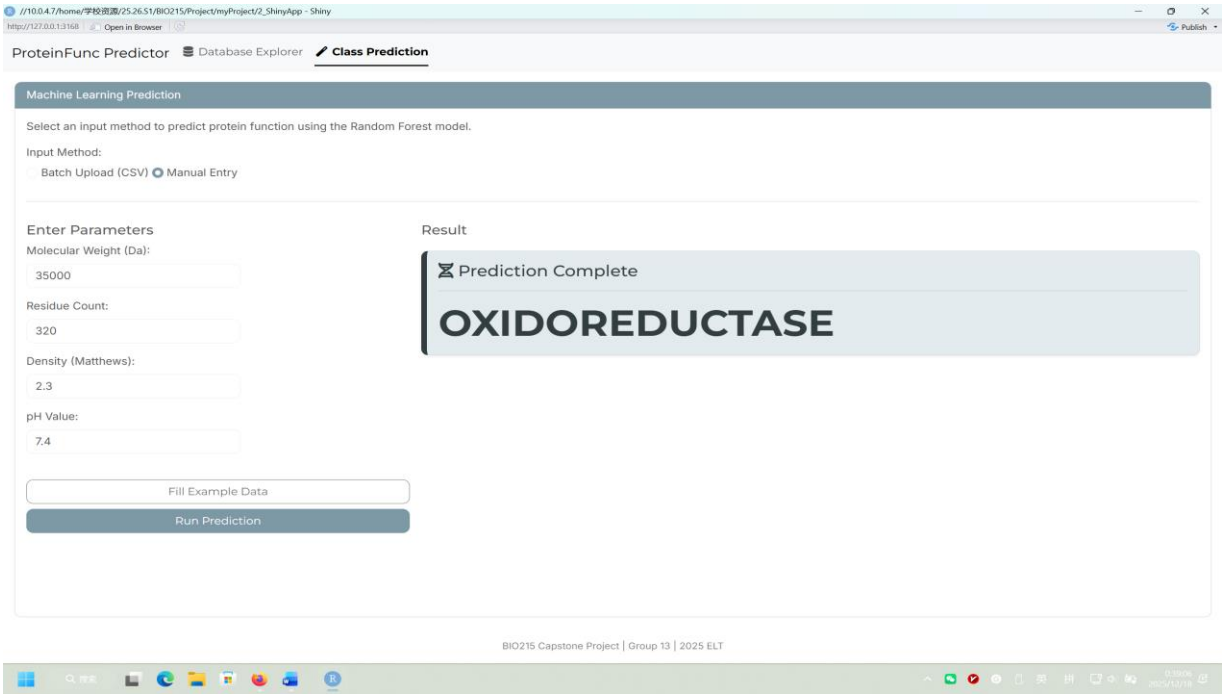


Figure 16: Manual Input Form and Prediction Output.

## 5. Discussion & Conclusion

### 5.1 In-Depth Review of Model Performance

The machine learning framework developed in this study demonstrates a strong capacity for predicting protein functional classes. Using only four global physicochemical descriptors, the Random Forest classifier achieved a test-set accuracy of 71.42%, indicating that these features capture substantial class-discriminative information. To provide a more nuanced evaluation beyond a single summary metric, model performance was further assessed using Receiver Operating Characteristic (ROC) and Precision–Recall (PR) curve analyses.

For all three enzyme classes (HYDROLASE, OXIDOREDUCTASE, and TRANSFERASE), the ROC curves consistently deviate from the random-classification diagonal, with key operating points achieving high true positive rates (up to 1.00) at controlled false positive rates (as low as 0.25). This indicates robust discriminative ability. The complementary PR curves show that high precision is maintained across a wide recall range, confirming that performance is not an artifact of class distribution but reflects genuine predictive skill.

Finally, the internal consistency of the model is supported by concordant feature importance rankings obtained from both Gini impurity and SHAP analyses. In both cases, structureMolecularWeight emerges as the most influential predictor, followed by residueCount, densityMatthews, and phValue. The stability of this hierarchy across different interpretability methods suggests that the model relies on robust and non-arbitrary feature relationships, lending additional credibility to its predictive behavior.

### 5.2 Utility for Biological Research and Deployed Tools

This project converts a predictive framework into an accessible, reproducible resource for computational and structural biology. As a proof of concept, a small set of tabular physicochemical descriptors derived from structural databases encodes meaningful functional signals, suggesting that

low-dimensional, structure-based representations can complement sequence-centric annotation when homology is insufficient.

The analysis emphasizes interpretability: SHAP results (Results, Figure 7) identify structureMolecularWeight and residueCount as dominant contributors across enzyme classes, motivating the testable hypothesis that protein size and structural complexity help distinguish major enzyme categories. Practically, two reusable tools are provided: an interactive Shiny app for code-free exploration and real-time prediction, and a reproducible R package (public on GitHub, installable via `devtools::install_github`) that encapsulates the pipeline and trained model, facilitating integration into broader bioinformatics workflows while ensuring transparency and reuse.

## 5.3 Limitations and Directions for Future Improvement

### 5.3.1 Constrained Physicochemical Feature Space

The predictive capacity of the model is inherently limited by the scope of its input features, constituting a methodological limitation of this study. The four global structural descriptors (structureMolecularWeight, residueCount, densityMatthews, phValue) capture only coarse structural information and lack sequence specificity, so they cannot resolve the fine-grained determinants of protein function.

Future work should expand the feature space with sequence-derived descriptors computable from primary sequences-e.g., k-mer frequency profiles [7] and amino-acid property indices [8]-and 3D structural features such as secondary-structure fractions [9]. Embeddings from pre-trained protein language models (e.g., ESM-2) can capture evolutionary and contextual signals that simple physicochemical summaries miss, improving predictive performance [10].

### 5.3.2 Restricted Functional Classification Breadth

The model was limited to classifying the three most common enzyme classes for tractability and dataset constraints. This improves training stability but narrows biological scope: it cannot identify rarer enzyme classes (e.g., Lyases, Ligases) or distinguish non-enzymatic proteins.

Extending the model beyond a proof-of-concept requires deliberate expansion of functional coverage by incorporating additional top-level Enzyme Commission (EC) classes [11] while explicitly addressing class imbalance. Adopting a hierarchical or multi-stage classification framework-first assigning broad categories and then making finer-grained predictions-would better reflect biological organization, reduce error propagation between levels, and allow targeted strategies for rare classes, thereby improving accuracy and interpretability at each decision step.

### 5.3.3 Model-Centric Algorithm Selection

Random Forest was chosen for its robustness, tolerance to feature noise, and interpretability. However, this represents a single-model implementation rather than a task-driven comparison. Other algorithms suitable for tabular biological data-such as XGBoost, LightGBM, and regularized neural networks-have not yet been evaluated within the current feature space.

Given the strong performance and efficiency of these methods on structured data, systematic benchmarking for protein functional classification is warranted. Future work should employ identical data partitions and validation protocols to ensure fair comparison, and explore ensemble strategies (e.g., stacking, voting) that may yield improvements beyond individual models [12,13].

### 5.3.4 Static Interpretation and Interactive Utility

SHAP provided global and local post-hoc explanations, but these outputs remain static in the analytical report and the Shiny app currently lacks interactive interpretation; consequently, users cannot inspect the rationale behind individual predictions, probe feature contributions, or run what-if analyses to see how changes in inputs affect model outputs.

Future versions might embed instance-level frameworks like shapper or DALEX [14,15] into the Shiny interface. This would let users visualize feature contributions (e.g., force or waterfall plots) and test hypothetical inputs in real time, transforming the system from an opaque predictor into an interactive environment for hypothesis generation and deeper exploration of protein structure–function relationships.

## 6. Conclusion

In this study, an explainable and fully reproducible machine learning pipeline was developed for protein functional classification. The proposed model achieves reliable predictive performance, as demonstrated by ROC and precision–recall curve analyses, while also yielding biologically interpretable insights derived from a compact set of physicochemical features. These results indicate that even simplified, structure-informed descriptors can capture meaningful functional signals relevant to enzyme classification.

Beyond model performance, particular emphasis was placed on usability and reproducibility. The deployment of an interactive Shiny application enables intuitive, code-free access to the model for exploratory analysis and real-time prediction, while the accompanying version-controlled R package ensures full transparency of the analytical workflow and supports seamless reuse within broader bioinformatics pipelines. All project resources are publicly available, facilitating independent validation, extension, and future methodological development by the wider research community.

## Reference


*[1] Salem, R., Aidaros, B. and Al-Obeidat, F. (2025) 'Exploring Deep Learning Models for Protein Sequence Classification: A Comparative Study', 2025 International Conference on Electrical, Communication and Computer Engineering (ICECCE), Electrical, Communication and Computer Engineering (ICECCE), 2025 International Conference on, pp. 1–6.*

*[2] Mi, J. et al. (2024) 'GGN-GO: geometric graph networks for predicting protein function by multi-scale structure features', Briefings in Bioinformatics, 25(6), pp. 1–10. doi:10.1093/bib/bbae559*

*[3] Li, C., Zheng, Y. and Jagodzinski, F. (2024) 'How pairs of insertion mutations impact protein structure: an exhaustive computational study', Bioinformatics Advances, 4(1), pp. 1–11.*

*[4] E. V. Malyugin and D. A. Afonnikov (2025) 'OrthoML2GO: homology-based protein function prediction using orthogroups and machine learning', Вавиловский журнал генетики и селекции, 29(7), pp. 1145–1154. doi:10.18699/vjgb-25-119.*

*[5] Nguyen, H.H., Viviani, J.-L. and Ben Jabeur, S. (2025) 'Bankruptcy prediction using machine learning and Shapley additive explanations', Review of Quantitative Finance & Accounting, 65(1), pp. 107–148. doi:10.1007/s11156-023-01192-x.*

*[6] Bini, G., Tamburello, G., Cacciaguerra, S., & Perfetti, P. (2025). sGs UnMix: a web application for spatial prediction and mixture modeling with a case study on volcanic soil CO2 fluxes. Environmental Modelling and Software, 193. https://doi.org/10.1016/j.envsoft.2025.106652*

*[7] Lesnick, M.L. et al. (2005) 'Identification of remote protein homologs by probabilistic comparison of sequence profiles using k-mer counts', Bioinformatics, 21(10), pp. 2302–2310.*

*[8] Kawashima, S. and Kanehisa, M. (2000) 'AAindex: Amino Acid Index Database', Nucleic Acids Research, 28(1), pp. 374.*

*[9] Kabsch, W. and Sander, C. (1983) 'Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features', Biopolymers, 22(12), pp. 2577–2637.*

*[10] Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa,*

*A., Fazel-Zarandi, M., Sercu, T., Candido, S. and Rives, A. (2022) 'Evolutionary-scale prediction of atomic-level protein structure with a language model', Science, 379(6637), pp. 1123–1130.*
*[11] Webb, E.C. (1992) Enzyme nomenclature 1992: recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes. San Diego: Academic Press.*
*[12] Dietterich, T.G. (2000) 'Ensemble methods in machine learning', in Multiple Classifier Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–15. Available at: https://doi.org/10.1007/3-540-45014-9_1.*
*[13] Džeroski, S. and Ženko, B. (2004) 'Is combining classifiers with stacking better than selecting the best one?', Machine Learning, 54(3), pp. 255–273.*
*[14] Maksymiuk, S., Gosiewska, A., Biecek, P., Staniak, M. and Burdukiewicz, M. (2020) shapper: Wrapper of Python Library 'shap' [R package]. Version 0.1.3. Available at: https://CRAN.R-project.org/package=shapper (Accessed: 20 December 2025).*
*[15] Biecek, P. (2018) 'DALEX: Explainers for Complex Predictive Models in R', Journal of Machine Learning Research, 19(84), pp. 1–5.*