LUND UNIVERSITY

SCHOOL OF ECONOMICS AND MANAGEMENT

# Predicting Protein Stability

## Using Supervised Machine Learning to Predict Protein Stability Caused by Amino Acid Variations

*Supervisor: Jonas Wallin and Mauno Vihinen*
*Student: Ibrahim Khan*
*Program: MSc Data Analytics and Business Economics*
*Lund University School of Economics and Management*

## Table of Contents

Predicting Protein Stability ........ 1

Using Supervised Machine Learning to Predict Protein Stability Caused by Amino Acid Variations ... 1

Acknowledgement ........ 4

Abstract ........ 5

1. Introduction ........ 6

1.1 Background ........ 6

1.2 Why does it matter to study protein stability? ........ 6

1.3 Impact of Variants on Protein Stability ........ 8

1.4 Research Question and Objectives ........ 9

2. Literature Review ........ 10

3. Methodology ........ 12

3.1 Models ........ 13

3.1.1 Random Forest ........ 13

3.1.2 XGBoost ........ 13

3.1.3 Neural Network ........ 13

3.1. 4 LSTM Model ........ 14

3.1.5 Logistic Regression Model ........ 14

3.2 Evaluation Metrics ........ 15

4. Data ........ 18

4.1 Preprocessing Data ........ 18

4.2 Studying the subsample of data. ........ 18

4.3 Features Extraction ........ 19

5. Analysis and Results ........ 22

5.1 Distribution of the Target Variable ........ 22

5.2 Dividing the Data for Evaluating the Models ........ 22

Parameter Optimization ........ 23

5.3 Two-layer Binary Classifier ........ 24

5.3.1 Results for Layer 1 ........ 24

5.3.2 Results for Second Binary Classifier ........ 26

5.4 Multi-class Classification ........ 28

5.5 Feature Importance ........ 30

Discussion ..........32
6. Conclusion and Future Research ..........33
7. References ..........34
Appendix..........37

**List of Figures**

Figure 1: Simple image of protein structure ..........7
Figure 2: Standard process of machine learning used in this study ..........12
Figure 3: Simple architecture of neural network. ..........14
Figure 4: Illustration of confusion matrix (image from Geeks for Geeks) ..........15
Figure 5: ROC Curve illustration (Developers.google.com) ..........16
Figure 6: Distribution of the target or label variable with cutoff values..........22
Figure 7: ROC Curve and AUC values for the first layer binary classifier ..........26
Figure 8: ROC Curve and AUC values for the second binary classifier..........28
Figure 9: ROC Curve values for the multiclass classification. ..........30
Figure 10: Feature importance plot for second binary classifier..........31
Figure 11: Feature importance for the first binary classifier..........31
Figure 12: Feature importance for the multi-class classification. ..........32

**List of Tables**

Table 1: Subsample selection of data..........18
Table 2: First layer binary classifier before and hyperparameter tuning. ..........23
Table 3: Second layer binary classifier before and hyperparameter tuning. ..........24
Table 4: Multi-class classification ..........24
Table 5: Final mode results for binary classifier 1..........25
Table 6: Results for binary classifier 2 ..........27
Table 7: Final model results for the multiclass classification..........29

# Acknowledgement

I am thankful to my family for their unwavering support during my academic journey. In addition, I am also grateful to my mentors from LUSEM and LUBMC for their guidance and support.

# Abstract

This study employs supervised machine learning methods to analyze the effect of amino acid variations on protein stability. It evaluates different models using a two-layer binary classifier method and multiclass classification to predict categories of increasing (I), decreasing (D), and neutral (N) stability. After preprocessing and exploring the dataset, a subsample was analyzed for single variant predictions. As this is the experimental data for the field of bioinformatics, feature engineering was carried out to collect the properties of single variants. Models such as Random Forest, XGBoost, Gradient Boosting, Neural Network, Logistic Regression, and LSTM were assessed. Cross-validation, parameter optimization, and feature selection were also performed to enhance performance.

Results showed improved predictions, with XGBoost outperforming other algorithms. Metrics like F1 score, AUC, precision, and recall were used to evaluate performance. The study demonstrates potential for further improvement with additional resources and shows that the two-layer binary classification performs better than multiclass classification in imbalanced datasets. Overall, this research lays the groundwork for future experiments on predicting protein stability using supervised machine learning methods.

**Key Words:** Machine Learning, Bioinformatics, Stability, Protein Variants

# 1. Introduction

## 1.1 Background

Stability is the foundational property influencing function, task, and regulation of biomolecules (Khan and Vihinen 2010). Changes in stability are usually found for the protein variants involved in diseases. The task to predict changes in protein stability due variations, represented as ΔΔG (change in Gibbs free energy), constitutes an important inquiry within the field of structural bioinformatics (Li, Yang et al. 2020). This inquiry not only deepens the theoretical understanding of protein folding and stability but also has practical implications across various sectors, including pharmaceuticals, genetic research, and the broader scope of molecular biology. Even though variations in stability can be studied experimentally, however those works are labor-intensive, time-consuming, and often expensive. Consequently, dependable computational techniques that can assist in predicting the stability changes are vital. With the highly availability of extensive biological datasets and the advancements in computational power, machine learning (ML) techniques, particularly those rooted in supervised learning, are being increasingly leveraged to tackle this complex problem. (Pak, Dovidchenko et al. 2023)

Traditional approaches to predicting protein stability, such as PON-tstab, have relied extensively on structure-based and empirical methods. These tools utilize static features extracted from protein structures and are often constrained by the availability and accuracy of experimental data. Moreover, they may not adequately capture the dynamic interactions within protein molecules that are crucial to understanding their stability upon mutation.

In contrast, modern machine learning models, such as tree models and deep learning frameworks, offer a dynamic approach to this challenge. These models can process large datasets to learn the intricate and complex patterns that dictate protein stability changes. This capability allows them to identify and understand the nuanced influence of amino acid variations on protein stability, providing a deeper and more comprehensive understanding than traditional experimental approaches (Gong, Jiang et al. 2023). Recent innovations, such as the ABYSSAL model which integrates the ESM2 architecture with attention mechanisms, have shown promising results by achieving high accuracy in forecasting stability variations in proteins because single variations (Pak, Dovidchenko et al. 2023).

## 1.2 Why does it matter to study protein stability?

Understanding protein stability is fundamental to numerous scientific and practical applications, influencing everything from drug design to industrial enzyme development. Moreover, insights into protein stability can lead to significant advancements in understanding diseases and developing novel treatments by elucidating how proteins fold, interact, and function under various conditions. By leveraging computational tools and machine learning techniques, researchers can predict and improve protein stability, driving innovation and addressing global challenges in health and industry (Yang, Ding et al. 2019).

The field of protein engineering through machine learning is gaining momentum, buoyed by recent achievements and significant advancements in related domains (Johnston, Fannjiang et al. 2023). It currently is comprised of interesting projects, such as knowing and predicting protein structures and functions, catalytic efficiency, enantioselectivity, protein dynamics, stability, solubility, aggregation, and more. Various machine learning techniques, including Random Forests, Support Vector Machines (SVM), and Gradient Boosting Machines (GBM), offer viable alternatives. These methods are especially useful for handling large datasets or when interpretability is essential. For instance, Random Forests are renowned for their robustness and ability to manage overfitting, making them suitable for complex biological data where numerous features may interact in nonlinear ways.

### What are Proteins?

Proteins are intricate, sizable molecules that perform numerous essential functions within the body. Composed of smaller units known as amino acids, these molecules form long chains. Examples of proteins include enzymes, antibodies, and muscle fibers. The encoding of proteins is carried out by genes, with the sequence of amino acids in protein being dictated by the nucleotide sequence of its corresponding gene. Protein stability refers to the propensity of a protein to maintain its structural integrity and functional state under varying physiological conditions. Stability is critical because it influences protein function, interaction, and overall cellular health. Changes in stability can lead to proteins becoming nonfunctional or gaining new, potentially harmful functions, which can result in diseases (Tsuboyama, Dauparas et al. 2023).

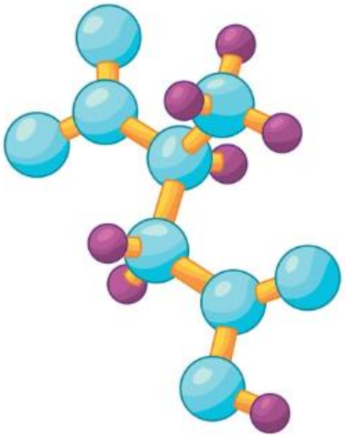


*Figure 1: Simple image of protein structure (Proteinatlas.org)*

### What is a Sequence?

In molecular biology, the term "sequence" typically denotes the arrangement of amino acids in a protein or sequence of nucleotides in a nucleic acid (DNA or RNA). This sequence is crucial as it dictates the protein's structure and function. Each amino acid in the protein's sequence plays a role in shaping the protein and determining its function. Likewise, the nucleotide sequence in a gene (a segment of DNA) demonstrates the amino acid sequence of the protein that the gene encodes. (Buric, Viknander et al. 2023)

**What are Variants?**

Variants refer to changes in the genetic sequence that can lead to differences in the protein sequence. These changes can occur naturally (as part of evolution or during cell replication) or can be induced artificially (Lodish, Berk et al. 2016).

## 1.3 Impact of Variants on Protein Stability

Genetic variations such as point variations can alter the amino acid sequence of proteins, potentially leading to changes in their structural stability (Sanavia, Birolo et al. 2020). This stability shift can be an increase, decrease, or neutral change. Predicting whether a variation will stabilize, destabilize, or not affect a protein is essential for multiple reasons:

- Disease Association: Many genetic disorders are directly linked to variations that destabilize proteins, causing loss of function or harmful gain of function.(Benevenuta, Birolo et al. 2023)
- Drug Development: Understanding how mutations affect protein stability can help in designing drugs that stabilize proteins or counteract the destabilizing effects of mutations (Birolo, Benevenuta et al. 2021).
- Functional Annotation: Forecasting or predicting the impact of variations on stability helps in annotating genetic variants with unknown effects, providing insights into their potential pathogenicity (Birolo, Benevenuta et al. 2021).

**Categorizing Stability Changes**

Dividing the impact of mutations into categories (increasing, decreasing, or neutral) aids in systematically assessing the potential consequences of each variant. This categorization is not just crucial for academic research but also for clinical genetics and pharmacogenomics, where such predictions guide therapeutic approaches and personalized medicine (Li, Yang et al. 2020).

**Relevance with the field of Data Analytics and Business Economics**

The biotechnology and healthcare sectors are on the verge of a revolution, driven by personalized medicine and targeted therapeutic interventions. The ability to predict protein stability through machine learning can significantly increase the pace at which new drugs are developed and brought to market, reducing costs, and improving outcomes. For businesses operating in these sectors, such advancements can lead to competitive advantages, opening new markets and enhancing profitability (Li, Yang et al. 2020).

**Methodology**

Having the raw data, it needed cleaning and pruning which was carried out to make the data ready for the further preprocessing and analysis. The study used the two layers random forest predictor model which was based on the earlier studies in the literature. Moreover, the study also carried out the multiclass classification. Different machine learning models were used, and their accuracy was determined on the subsample of data to identify the stability of single variants protein cells.

**Results**

XGBoost algorithm performed well relative to other models or algorithms utilized in the study. Neural Networks, LSTM and other models also performed better than the previous studies. The results reveal that there is increased potential in the machine learning models to predict the stability of protein cells accurately especially with the ensemble learning models when the resources are in place i.e., technical resources and powerful server. XGBoost achieved the accuracy of 78% and 79% in first and second layer respectively and it achieved 72% accuracy in the multiclass classification. In multiclass classification, the gradient boosting outperformed the random forest model achieving better accuracy and other metrics.

## 1.4 Research Question and Objectives

This thesis aims to assess the performance of various supervised machine learning models, including neural networks and tree-based methods such as Random Forests, in predicting protein stability changes caused by single variations. The study will also compare these models with established computational tools like PON-tstab and other methods developed in the field, focusing on accuracy and reliability metrics. The specific objectives of this study are to:

1. *Assess the predictive accuracy of different models on ΔΔG for variations.*
2. *Perform a comparative analysis of the machine learning models in predicting stability.*

By achieving these objectives, this research will significantly contribute to the bioinformatics community, providing insights into the strengths and limitations of both traditional and modern methods in protein stability prediction.

**Research Objective**

How can supervised machine learning models accurately predict the stability of protein variants, and how does this prediction compare to existing tools such as PON-tstab in terms of accuracy and reliability?

Rest of the essay follows the below structure.

- Literature review
- Methodology
- Data
- Results and Analysis
- Conclusion and Future Research
- References

# 2. Literature Review

Efforts to correctly predict changes in protein stability resulting from amino acid variations have become a key focus in bioinformatics, thanks to advancements in machine learning (ML) and the availability of large-scale datasets. This literature review covers essential studies that have made great contributions to this field, emphasizing the integration of traditional computational tools with modern ML techniques to improve predictive accuracy and reliability.

### Innovations in Machine Learning for Protein Stability Predictions

A landmark study published in *Nature* by Tsuboyama, Dauparas, et al. (2023) underscores the potential of machine learning in bioinformatics, specifically in predicting protein stability changes. This research demonstrates that ML models can effectively determine how amino acid substitutions influence protein stability, which has wide-ranging implications from metabolic processes to immune responses. The study introduces ABYSSAL, a novel predictive tool that utilizes deep learning techniques to analyze vast datasets, significantly improving the accuracy of protein stability predictions. Such advancements are crucial for knowing the functional influence of protein mutations, thereby aiding in drug development and disease comprehension.

### Comparison of Machine Learning Methods

Yang, Chen, et al. (2013) introduce a structure-based method that utilizes support vector machine classifiers to forecast the impacts of amino acid substitutions on protein stability. By concentrating on changes in amino acid contact energy, their approach surpasses the accuracy of existing models, providing a more dependable tool for protein design and the study of disease-related variations. Additionally, the study led by Yang, Urolagin et al. (2018) introduces PON-tstab, a tool that significantly enhances prediction accuracy through the correction of data quality issues in the ProTherm database and the application of a random forests approach. This study emphasizes the importance of high-quality training data in developing effective predictive models.

### Deep Learning Advances

AlQuraishi (2019) investigates the use of deep learning, a sophisticated subset from machine learning, to predict both protein stability and the three-dimensional structures of proteins from genetic sequences. This advancement demonstrates the extensive potential of machine learning in uncovering fundamental aspects of protein functionality and interactions.

### Machine Learning in Drug Discovery and Database Utilization

Chen et al. (2018) reviewed the use of machine learning in drug discovery, emphasizing techniques for predicting interactions between drugs and protein targets. Understanding the influence of drugs on protein stability is crucial for creating effective and safe therapies. Additionally, bioinformatics databases like the

Protein Data Bank (PDB) and UniProt are essential resources for machine learning research in protein studies, as highlighted by Berman et al. (2000) and The UniProt Consortium (2019). These databases supply the critical data needed to advance bioinformatics research, including machine learning applications for predicting protein stability.

### Evaluating Computational Prediction Methods

Vihinen (2012) provides a framework for evaluating the performance of computational prediction methods, introducing key metrics such as sensitivity, specificity, and Matthew's correlation coefficient. This systematic analysis, coupled with ROC analysis, helps researchers assess and select the most effective tools for their specific research needs. Niroula and Vihinen (2016) further discuss the role of next-generation sequencing in generating genetic variation data essential for disease diagnosis and the subsequent need for computational methods to interpret these variations.

### Implications and Future Directions

The integration of machine learning into bioinformatics is not merely enhancing our understanding of protein stability; it is revolutionizing approaches in drug discovery and genetic research. Predictive models that accurately determine how modifications to proteins affect their stability are leading to more targeted and effective therapeutic interventions. As machine learning technologies continue to evolve, their application in bioinformatics promises to unlock new frontiers in biomedical research and personalized medicine. (Dara, Dhamercherla et al. 2022)

Moreover, a study published in eLife introduced a deep learning model named RaSP, which offers rapid predictions of protein stability with notable accuracy. This model was benchmarked against established methods like Rosetta, showing its efficacy in handling various amino acid substitutions and providing a detailed assessment of protein stability across different environments (Blaabjerg, Kassem et al. 2023).

Another notable advancement is the creation of a method utilizing Equivariant Graph Neural Networks to predict variations in protein stability due to multiple amino acid substitutions. This approach adeptly connects atomic and residue scales by using a graph structure focused on mutant residues, which improves the predictive accuracy of stability changes and meets an essential requirement in protein engineering (Boyer, Money-Kyrle et al. 2023).

This study contributes to the literature by using higher number of features and using different machine learning methods to see how well they perform.

# 3. Methodology

This paper used the standard machine learning model implementation procedure to evaluate the stability of variations in the protein cells. The preprocessing of data was done in the first step right after collecting the data. Feature engineering was carried out which plays a critical role in the bioinformatics field because most of the features are extracted based on the best practices from the external sources. The data features were standardized. The model of performance was checked before and after feature selection. In addition, hyperparameter tuning or parameter optimization was also performed to check whether that improve the results or not.

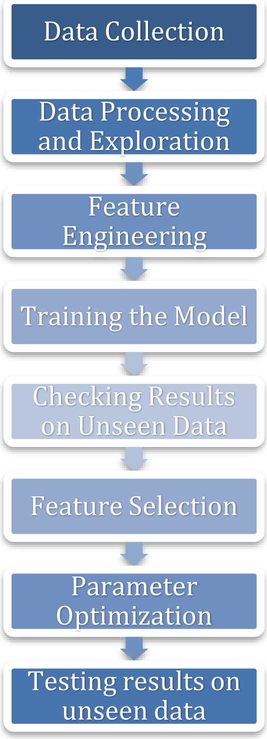

Data Collection
Data Processing and Exploration
Feature Engineering
Training the Model
Checking Results on Unseen Data
Feature Selection
Parameter Optimization
Testing results on unseen data

*Figure 2: Standard process of machine learning used in this study*

The preprocessing steps were carried out after exploring the data. After initial exploration, the data was cleaned from unnecessary and redundant values. The single variants protein stability predictors were separated from the double variants' protein stability. This paper carried out two methods to study the protein stability with the help of supervised machine learning models. This choice is also evident from the earlier studies in the literature. (Yang, Urolagin et al. 2018). The two methods are given as under:

**Two-layer Binary Classifier Method**

Here, the study divided the data into two binary classifiers. The first classifier included the data on observations where it was between Decreasing Stability and Non-Decreasing Stability. With non-decreasing stability, the study included both the Increasing and Neutral ones which were equal to the number of decreasing observations. The 2nd layer or second binary classifier was where the study further binary classified the non-decreasing into increasing and neutral classes. It is also useful for the cases where there is imbalance in the dataset which is the case in this study (Demidova and Klyueva 2021).

**Multi-Class Classification Model**

The study also evaluated the performance of supervised machine learning models on multiclass classification to predict the stability of protein cells due to the variations in amino acids. Here, I randomly selected the increasing, neutral, and decreasing stability of protein cells so that I have balanced dataset and then applied the models to check the performance. It is different from the two-layer binary classifier in a sense that it uses the three categories.

## 3.1 Models

The study used different algorithms to check the performance. Below is short discussion about them.

### 3.1.1 Random Forest

The Random Forest is an ensemble learning method that creates multiple decision trees during training and uses the mode of the classes for classification tasks. It is particularly effective for predicting protein stability due to its capacity to manage large, high-dimensional datasets common in bioinformatics. The model's robustness stems from its bootstrapping technique and random feature selection, which reduces overfitting and enhances generalization. Adjusting hyperparameters, such as the number of trees and the maximum depth, further improves model performance. For feature selection, the Gini index is often employed, helping the model identify the most relevant features that contribute to protein stability.

### 3.1.2 XGBoost

XGBoost (Extreme Gradient Boosting) is effective and efficient execution of gradient boosting algorithms. It excels in predictive performance due to its regularization capabilities, which prevent overfitting, and its parallel processing, which speeds up computation. XGBoost is particularly suitable for predicting protein stability because it is robust to outliers. Hyperparameter tuning, such as optimizing the learning rate, maximum depth, and subsample ratio, is crucial for maximizing the model's predictive power. Feature importance can be accessed via the ridge or lasso methods to remove the features that are not significantly contributing to the performance of the model.

### 3.1.3 Neural Network

Neural Networks, especially deep learning models, excel at complex pattern recognition tasks such as predicting protein stability. These models can identify intricate non-linear relationships in the data through their multiple layers of neurons. The flexibility and learning capability of neural networks make them ideal for bioinformatics applications, where the feature-outcome relationship is often highly complex. Enhancing model performance involves hyperparameter tuning, which includes adjusting parameters such as the number of layers, neurons per layer, learning rate, and activation functions. Techniques like backward feature elimination or forward selection or feature importance can be used to identify the most significant features.

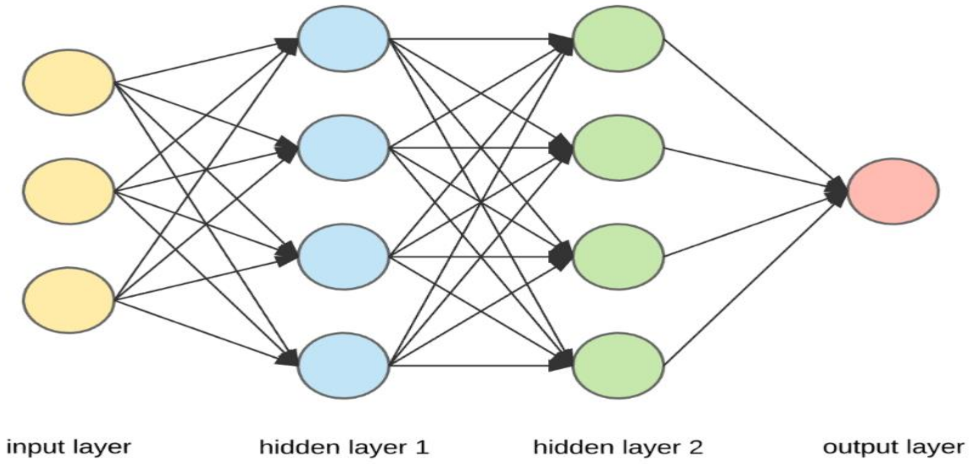


*Figure 3: Simple architecture of neural network.(upgrad.com)*

### 3.1. 4 LSTM Model

Long Short-Term Memory (LSTM) networks are a specialized form of recurrent neural networks (RNN) that excel in handling sequential data and time-series analysis. They are particularly effective in bioinformatics for tasks involving sequences, such as predicting protein stability over time, due to their ability to capture long-term dependencies and patterns. LSTMs utilize memory cells to retain information over extended sequences, making them well-suited for managing temporal dependencies in protein data. Improving LSTM performance involves hyperparameter tuning, which includes adjusting the amount of layers, units per layer, learning rate, and dropout rate. For feature selection, embedded methods like feature importance can be employed to identify critical sequence features that contribute to protein stability predictions.

### 3.1.5 Logistic Regression Model

For the binary classifier, I also utilized the logistic regression model to assess its accuracy on unseen data and its compatibility with the high-dimensional data typical in bioinformatics. The loss function employed in logistic regression is binary cross-entropy loss, or log-loss, which evaluates the performance of a classification model that outputs a probability value between 0 and 1.

### 3.1.6 Gradient Boosting

Gradient Boosting is a machine learning technique that builds an ensemble of decision trees sequentially, where each tree corrects the errors of the previous ones. This method is highly effective for both binary and multiclass classification tasks due to its ability to handle complex patterns in data. It reduces bias and variance, enhancing precision and recall in binary classification and providing robust predictions in multiclass scenarios.

## 3.2 Evaluation Metrics

Below are the metrics that were used for the assessment of the performance of the different models in this study. The metrics are chosen based on the earlier studies in the literature which have also used these metrics. (Yang, Urolagin et al. 2018)

**Confusion Matrix**

The confusion matrix refers to table that describes the performance of a classification model. It summarizes the number of of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.

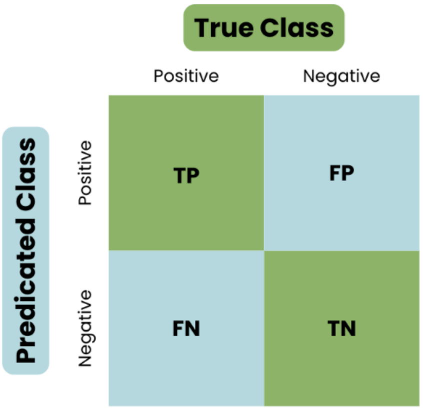


*Figure 4: Illustration of confusion matrix (image from Geeks for Geeks)*

**Accuracy**

Accuracy calculates the overall performance of a model in correctly predicting class labels, considering both true positives and true negatives. The formula for accuracy is:

$$Accuracy = TP + TN/TP + FP + TN + FN$$

where TP refers to true positives, TN refers to true negatives, FP shows false positives, and FN shows false negatives.

**Precision**

Precision assesses the model's accuracy in identifying positive values. It is determined as the ratio of true positives to the total number of true positives and false positives:

Precision (P)=TP/TP+FP

**Recall**

Recall, also referred to as Sensitivity or True Positive Rate, gauges the model's effectiveness in accurately identifying positive instances. It is computed as the ratio of true positives to the total number of true positives and false negatives:

Recall = TP/TP+FN

**Specificity**

Specificity assesses the model's capability to correctly identify negative instances. It is calculated as the ratio of true negatives to the total number of true negatives and false positives:
Specificity (Sp)=$TN$/TN+FP

**F1 Score**

The F1 Score refers the harmonic mean of precision and recall. It provides a one metric that balances both concerns, especially helpful when the class distribution is not balanced.

F1= $$\frac{\mathrm{2tp}}{\mathrm{2tp}+\mathrm{fp}+\mathrm{fn}}.$$

**Area Under the Curve (AUC)**

The AUC (Area Under the Curve) determines the model's ability to distinguish between classes. A higher AUC value indicates a well-performing model, as it shows greater separability between the classes. The formula for AUC is:

$\mathrm{AUC}=\int 01\mathrm{TPR}(\mathrm{FPR}-1(x))dx$

**Receiver Operating Characteristic (ROC) Curve**

The ROC curve refers to the graphical representation of classifier's performance across all classification thresholds. It plots the true positive rate (recall) against the false positive rate (1-specificity).

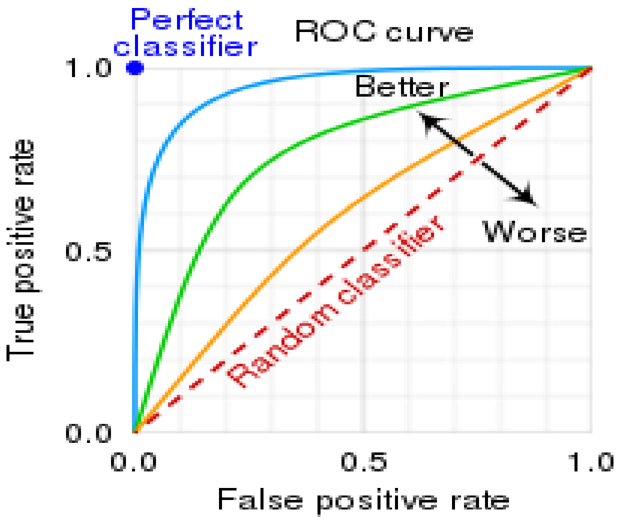


*Figure 5: ROC Curve illustration (Developers.google.com)*

**MCC**

The Matthews Correlation Coefficient (MCC) is a performance metric for binary classifiers in machine learning that considers all four elements of the confusion matrix: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The MCC is calculated using the formula:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}}.$$

This formula ensures that MCC provides a balanced measure of a classifier's performance across both classes, which is particularly useful in cases of class imbalance.

In summary, these evaluation metrics offer a detailed understanding of various facets of model performance, including overall accuracy, the balance between precision and recall, and the capacity to differentiate between classes. Employing these metrics ensures a comprehensive evaluation of the machine learning models used to predict protein stability.

**Cross Validation**

Cross-validation is an essential technique in evaluating the performance of machine learning models, particularly in bioinformatics where data can be high-dimensional and imbalanced. In this study, I employed 10-fold cross-validation to ensure the robustness and generalizability of our models. This method involves partitioning the dataset into ten equal parts, training the model on nine parts, and validating it on the remaining part, repeating this process ten times. The average performance metrics from these iterations provide a reliable estimate of the model's efficacy. Cross-validation is crucial as it mitigates overfitting, ensures that the model's performance is consistent across different subsets of the data, and provides a comprehensive assessment of its predictive capabilities.

# 4. Data

This study fetched the experimental data on protein stability changes due to mutations from Zenodo repository. The data was associated with the paper by Tsuboyama et al., 2022, where ΔΔG values were derived with protease cleavages.[1]

## 4.1 Preprocessing Data

The study adhered to standard preprocessing practices as a prerequisite for applying machine learning models. Since the dataset contained both single and double variants, it was necessary to separate them into distinct datasets, focusing solely on building a predictor for single variants. Initially, the raw data underwent a cleaning process to remove unnecessary and redundant information. Exclusions included data labeled as 'unreliable,' entries without variations, those linked to insertions and deletions, and records involving multiple mutations. After filtration, the dataset comprised information on 376,918 single variations across 396 proteins and 152,383 variations from double variants. This study is only concerned with the single variations, so the double variations were separated.

## 4.2 Studying the subsample of data.

With the values given in table 1, it is evident that the dataset is highly imbalanced with the increasing number of values only 15790 which is typical in the case of bioinformatics. Therefore, I worked on balancing the dataset and reduced the number of observations to match the increasing features to avoid the biases in the training of models. I randomly selected 31580 observations from decreasing observations inclusive of all the variants whereas I randomly selected 15790 from Neutral observation. So, now in the first layer of classification, the study performs the binary classification between the decreasing (D) features which are 31580 and the non-decreasing features which is the sum of increasing (I) and neutral (N). For layer 2, the study has neutral and increasing variants only to see if they are increasing or are neutral.

| **Classification** | **Number of Values** | **Categories of Stability** | **Number of observations after subsample** |
|---|---|---|---|
| Values below -0.5 | 192861 | Decreasing | 31580 |
| Values between -0.5 and +0.5 | 168267 | Neutral | 15790 |
| Values above 0.5 | 15790 | Increasing | All |

*Table 1: Subsample selection of data*

[1] https://zenodo.org/record/7401275#.Y6st59JBxD_

This is also shown in the figure 1.

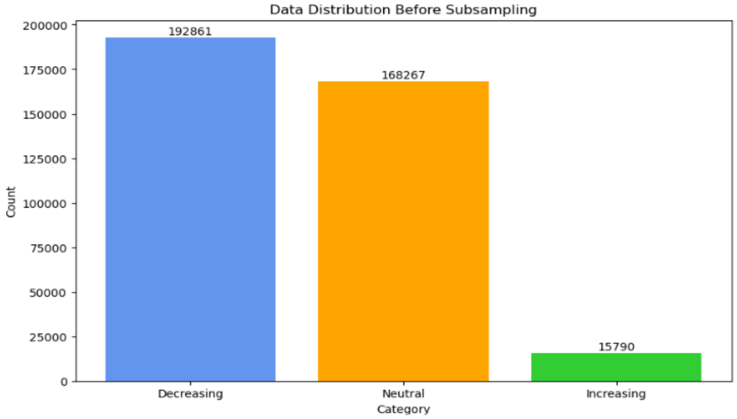


*Figure 6: Plot showing the initial distribution of three categories in single variants.*

In addition to using the subsample from the dataset, the study used the two methods to analyze the variations and how they impact protein stability:

1. 2-Layer Random Forest binary classifier
2. Mutli-Class Classification

Due to the technical issues and volume constraints, I used the 47390 observations with each category having equal number of observations (15790) for multiclass classification. In addition, I used the 63180 for 2-layer binary classification model. This is done in accordance with the previous work done in the similar field. The methods used here are in accordance with the previous studies in this field (Yang, Urolagin et al. 2018).

## 4.3 Features Extraction

In the context of predicting protein stability changes due to variations, the use of comprehensive feature engineering plays a pivotal role.(Tsuboyama, Dauparas et al. 2023). By utilizing a variety of biochemical and biophysical properties, one can significantly enhance the predictive capabilities of machine learning models. In total, 1483 features were collected to train machine learning models for stability prediction. This section discusses the specific features utilized in the study, explaining their biological relevance and computational treatment.

### 1. Amino Acid Index (AAI) Features (617)

The AAIndex features consist of a database of numerical indices that shows different physicochemical and biochemical properties of amino acids. These indices are essential for understanding the effects of variations at the molecular level. By computing the differences in these indices between the wild-type and mutant amino acids, the model can accurately assess how a substitution may impact the protein's stability (Kawashima, Pokarowski et al. 2007). This differential approach captures the direct effects of the amino acid change, providing a detailed understanding of its implications for stability. A total of 617 complete sets were utilized.

### 2. Neighborhood Features (25)

Neighborhood features provide context about the domestic sequence environment around the mutation site. These features include the properties of amino acids located near the mutation, potentially capturing local structural and functional effects that a simple substitution analysis might miss. Such features are crucial for understanding how changes in one part of the protein influence its nearby structures and, consequently, the overall stability (Torng and Altman 2017). Total of 25 features were derived from neighborhood.

### 3. Thermodynamics Indices: W(U), Gw(U), and Gs(U) (3)

- W(U) - Number of water molecules close to a residue in an unfolded state: This feature quantifies the hydration shell around amino acids in an unfolded protein state, offering insights into the solvation dynamics and hydrophobic interactions that are crucial for protein folding and stability.
- Gw(U) - Free energy contribution from the entropy of the first shell of water molecules in an unfolded state: This index measures the entropic energy contributions from water molecules near the mutated residue, which are essential for understanding the thermodynamic aspects of protein unfolding.
- Gs(U) - Interfacial free energy contribution of an unfolded state: This feature captures the free energy change due to interfacial interactions between the protein and its environment in an unfolded state. Such interactions are significant as they affect the protein's tendency to fold or remain unfolded (Nakagawa and Tamada 2021).

| Features | Dimensions | Notes |
|---|---|---|
| Variation based features | Variation features (nutationAll_features) | 436-D |
| | Neighbourhood features | 25-D |
| | Variation first position (position_features) | 1-D |
| | Dipeptide composition | 400-D |
| | AAIndex features | 617-D |
| | Residue feature | 1-D |
| | ProtDCal - Thermodynamics indices | 3-D |
| | ProtDCal - ISA - ECI features | 2-D |

*Table 2: Features extracted for this project*

### 4. ECI (Electronic Charge Index)

The Electronic Charge Index (ECI) represents the net electric charge of an amino acid, which affects how it interacts with its environment, particularly with charged or polar substrates. Changes in charge due to mutations can dramatically affect protein stability by altering electrostatic interactions within the protein or between the protein and surrounding water molecules (Requião, Fernandes et al. 2017).

**5. ISA (Isotropic Surface Area)**

The Isotropic Surface Area (ISA) of amino acids is a measure of the accessible surface area that is available for interaction with surrounding molecules. Variations in ISA due to mutations can affect protein stability by altering surface exposure and potential interactions with other biomolecules, which are crucial for maintaining structural integrity (Requião, Fernandes et al. 2017).

**6. Get_nutation_all, Group_all_features, and Cal_dipeptide_features**

- Get_nutation_all (400): This function aggregates mutation-specific features, providing a comprehensive dataset that includes all relevant biophysical and biochemical characteristics impacted by the mutation.
- Cal_dipeptide_features (400): This involves calculating features related to dipeptide properties, which consider the interaction between consecutive amino acids. These features are particularly valuable in understanding how variations affect local structural properties.

The above features are also in line with the previous studies in the literature (Khan and Vihinen 2010).

# 5. Analysis and Results

As mentioned earlier, the models that were discussed earlier to be adopted for this study were passed through the standard machine learning process. The features were standardized. Parameter optimization or hyperparameter tuning was carried out to see if the performance is improved. Feature selection was also performed for each model to check if that affects the performance of the model.

## 5.1 Distribution of the Target Variable

The target variable values ranged between -5 and 3. The classification threshold was established to identify variants contributing to increasing, decreasing, or neutral stability. The figure below illustrates the distribution of these values. Based on previous literature, the values were categorized into three groups: Increasing Stability, Neutral Stability, and Decreasing Stability. This categorization was determined by their proximity to the value 0.5.

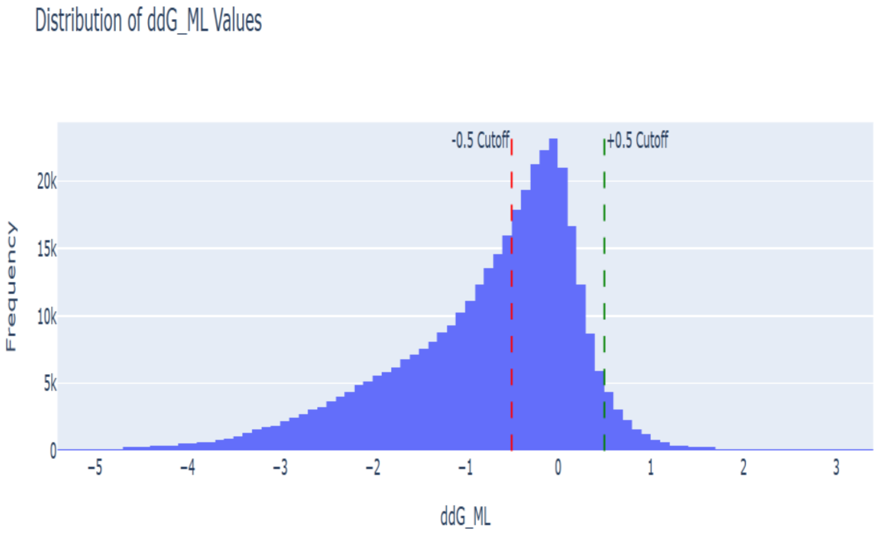


*Figure 7: Distribution of the target or label variable with cutoff values*

The distribution of the data studied reflects the real-world data of amino acid variations i.e., like this, the number of observations predicting the increase in stability of protein cells is always lower than those of decreasing and neutral stability predictor cells in real life as well.

## 5.2 Dividing the Data for Evaluating the Models

As it is evident that to teach the model, the dataset must be divided into training and testing phases. I did the same and divided my data into training and testing subsets which meant that 80% is allocated for the training and 10% is allocated for the testing. When dividing the data it was assured that the testing data does not have the parallel or repetitive data from the training phase to avoid the data leakage and properly test the blind data. The following proportion was maintained for both the multi-class classification and the 2-layer random forest prediction model.

- Training data: 80 % for training dataset
- Testing data: 20% for testing data

For training dataset, the data was divided into 10 partitions of equal sizes and utilized in ten-fold cross validation. As stated earlier, there are three categories of classification for the mutations in proteins. There is a category that predicts the increasing stability, decreasing and neutral ones. I designed the model in such a way that. The data was high dimensional with 1483 features and rows. I also standardized all the features which is a general procedure for training and testing the power and performance on the machine learning model. Feature importance for all the models was also performed to observe what are the important features.

## Parameter Optimization

As a general procedure in machine learning for enhancing the performance, the parameter optimization was also carried out in this study. Random search was used to select the best parameters and then the performance of those parameters was evaluated on the unseen data to see how well they performed. While random search was computationally expensive, the study also experimented with parameters individually in the models. The parameter optimization significantly improved the performance of all the models. The table in the following chapter presents the performance of models after hyperparameter tuning. The following tables show how the hyperparameter tuning improved the performance in all the models.

| First Layer Binary Classifier | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Performance Metrics** | **Random Forest** | | **XGBoost** | | **Neural Network** | | **LSTM** | | **Logistic Regression** | |
| | No HT | HT | No HT | HT | No HT | HT | No HT | HT | No HT | HT |
| Accuracy | 70% | 76% | 72% | 78% | 70% | 71% | 61% | 63% | 68% | 70% |
| Specificity | 76% | 71% | 71% | 72% | 70% | 71% | 43% | 43% | 64% | 63% |
| Precision | 70% | 71% | 76% | 78% | 71% | 72% | 61% | 62% | 66% | 69% |
| NPV | 71% | 70% | 70% | 72% | 70% | 71% | 60% | 60% | 67% | 67% |
| MCC | 40% | 42% | 54% | 55% | 41% | 43% | 31% | 32% | 42% | 43% |
| F1 Score | 71% | 72% | 73% | 74% | 70% | 72% | 60% | 61% | 64% | 65% |
| Recall | 69% | 70% | 71% | 72% | 71% | 71% | 59% | 61% | 67% | 68% |

*Table 3: First layer binary classifier before and hyperparameter tuning.*

| Second Layer Binary Classifier | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Performance Metrics** | **Random Forest** | | **XGBoost** | | **Neural Network** | | **LSTM** | | **Logistic Regression** | |
| | No HT | HT | No HT | HT | No HT | HT | No HT | HT | No HT | HT |
| Accuracy | 69% | 74% | 79% | 79% | 72% | 74% | 58% | 60% | 68% | 69.5% |
| Specificity | 75% | 70% | 73% | 73% | 70% | 71% | 44% | 46% | 60% | 62% |
| Precision | 71% | 73% | 79% | 79% | 72% | 74% | 60% | 61% | 68% | 69.5% |
| NPV | 72% | 73% | 75% | 75% | 71% | 72% | 57% | 60% | 67% | 69% |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| MCC | 41% | 43% | 57% | 57% | 45% | 47% | 20% | 20% | 37% | 39% |
| F1 Score | 74% | 71.4% | 79% | 79% | 73% | 74% | 58% | 60% | 66% | 69.4% |
| Recall | 70% | 72% | 74% | 74% | 72% | 74% | 57% | 60% | 67% | 69.4% |

*Table 4: Second layer binary classifier before and hyperparameter tuning.*

| Multi-Class Classification | | | | | | |
|---|---|---|---|---|---|---|
| Performance Metrics | Gradient Boosting | | Random Forest | | Neural Network | |
| | NHT | HT | No HT | HT | NHT | HT |
| Accuracy | 65% | 69% | 55% | 61% | 60% | 64% |
| Specificity | 65% | 68% | 58% | 60% | 62% | 64% |
| Precision | 67% | 68% | 57% | 60% | 62% | 64% |
| NPV | 32% | 31% | 35% | 39% | 33% | 37% |
| MCC | 52% | 53% | 40% | 41% | 41% | 46% |
| F1 Score | 66% | 68% | 58% | 60% | 61 | 64% |
| Recall | 66% | 67% | 60% | 62% | 62% | 64% |

*Table 5: Multi-class classification*

## 5.3 Two-layer Binary Classifier

### 5.3.1 Results for Layer 1

In layer 1, the dataset was prepared to show binary classification between the decreasing and non-decreasing protein stability categories. The non-decreasing group included both the increasing and neutral variants. The total number of observations were 63180 where 31740 were decreasing and the same number were non-decreasing This categorization was crucial for reducing the complexity of the model's outcome. For the loss function, the study used the default gini impurity in the tree models in random forest model.

In the implementation of the Random Forest classifier for the stability test data, different preparatory steps were undertaken to ensure the reliability and accuracy of the model's performance metrics. Hyperparameter tuning was performed with the random search and then the Random Forest was configured with 600 trees, a maximum depth of 20, minimum samples split of 10, a minimum samples leaf of 4, and a maximum features parameter set to 'sqrt'.

Cross-validation on the training data yielded a mean accuracy of 72%. After training the model on the entire training set, the test set accuracy was also 72%. Hyperparameter tuning using random search CV improved the test accuracy to 76%. The best hyperparameters were found by testing various combinations, enhancing the model's performance.

The XGBoost model was configured with specific hyperparameters: 500 estimators, a learning rate of 0.1, depth of 20, and subsampling and column sampling rates of 0.8 each. A pipeline was established that included a standard scaler and the XGBoost classifier to standardize the features before training. The cross-validation results showed a mean accuracy of 0.8108, indicating consistent performance across folds. The final model, trained on the full training set, achieved a test accuracy of 0.7812. Additionally, the model's precision, recall, F1 score, and Matthew's correlation coefficient (MCC) were all also moderate. The MCC, a robust measure for imbalanced datasets, was 0.5225, indicating a strong correlation between the predicted and actual labels. The ROC AUC score of 0.8953 further confirmed the model's high discriminative power.

In summary, the XGBoost model, after hyperparameter tuning, exhibited high accuracy, precision, recall, F1 score, and MCC, making it an excellent choice for the stability classification task. An MCC value of 0.54 suggests that the XGBoost model performs well across all classes (increasing, decreasing, and neutral stability of protein variants). It indicates that the model is not biased towards any class and can make balanced predictions.

| Performance Metrics | Random Forest | **XGBoost** | Neural Network | LSTM | Logistic Regression |
|---|---|---|---|---|---|
| Accuracy | 76% | **78%** | 71% | 63% | 70% |
| Specificity | 71% | **72%** | 71% | 43% | 63% |
| Precision | 71% | **78%** | 72% | 62% | 69% |
| NPV | 70% | **72%** | 71% | 60% | 67% |
| MCC | 42% | **55%** | 43% | 32% | 43% |
| F1 Score | 72% | **74%** | 72% | 61% | 65% |
| Recall | 70% | **72%** | 71% | 61% | 68% |

*Table 6: Final mode results for binary classifier 1*

To learn from the properties of the dataset, the neural network model created for the classification job uses a Sequential architecture with several dense layers. To avoid overfitting, the model has six hidden layers of 512, 256, 128, 64, 32, and 16 neurons each. Each hidden layer was followed by a dropout layer with a 30% dropout rate. Scaling of the input features and conversion of labels to a one-hot encoding scheme were done. The model was trained with early stopping to track validation loss, and it was constructed using the Adam optimizer and categorical cross-entropy loss function. Based on evaluation, the model's accuracy on the test set was about 71.39%. The precision, recall, and F1-score metrics showed that the model performed equally well in both classes.

In addition to the other models, a logistic regression model was established with a maximum of 10,000 iterations to assure convergence and a random state of 42 to ensure reproducibility. The test dataset was used to evaluate the model after it had been trained on the training data. The model produced probabilities and predictions, and its performance was assessed using a variety of indicators. This classification model produces a probability value between 0 and 1, and its performance was measured using the binary cross-entropy loss, also known as log-loss.

Additionally, an LSTM model for a binary classification task was created in the study using Keras. The two 50-unit LSTM layers in the architecture come after an input layer. Sequences are returned by the first LSTM layer, and the final sequence is output by the second LSTM layer. After every LSTM layer, dropout layers with a 30% dropout rate are added to reduce overfitting. At the end of the model, there is a dense layer with a single neuron that generates a binary output using a sigmoid activation function. The binary cross-entropy loss function and Adam optimizer were used to create the model, and accuracy was used to assess its performance.

The results of the first binary classifier are also plotted in the ROC Curve as shown below. As evident from the plot, the XGBoost outperforms the other models used in the first layer binary classifier. XGBoost represented by green line outperforms the other models followed by neural network, random forest, logistic regression and LSTM.

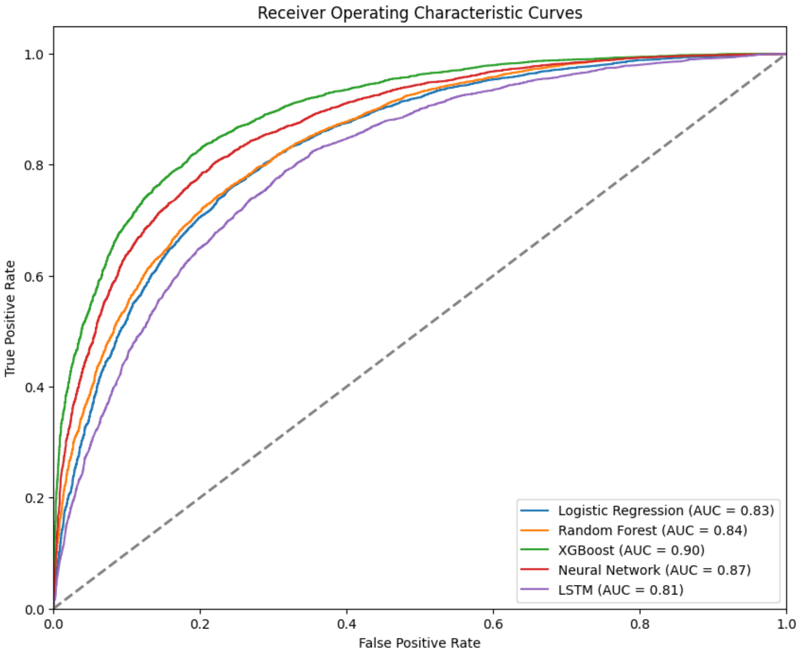


*Figure 8: ROC Curve and AUC values for the first layer binary classifier.*

### 5.3.2 Results for Second Binary Classifier

In the second layer, the study had the I and N category from the first layer. Both had the same number of observations i.e., 15790. The same models or algorithms used in the first layer were also utilized for the second layer. Features were standardized and parameter optimization was carried out to improve the performance of the models for the prediction purpose. I used the same parameters for the models that I used in the first binary classifier to ensure consistency in the model performance and evaluation of the two models.

700 trees, a maximum depth of 20, minimum samples split of 10, a minimum samples leaf of 4, and a maximum features parameter set to'sqrt' were the configuration parameters for the Random Forest. The training data's cross-validation produced a mean accuracy of 72%. Following the model's training across the whole training set, the accuracy on the test set was also 72%. By utilizing random search CV for hyperparameter tweaking, the test accuracy was increased to 76%. By experimenting with different combinations, the optimal hyperparameters were discovered, improving the model's functionality.

Like the layer 1 model, the XGBoost model was set up using the following hyperparameters: 500 estimators, 0.1 learning rate, 20 maximum depth, and 0.8 rates for both subsampling and column sampling. To standardize the features before to training, a pipeline comprising the XGBoost classifier, and a standard scaler was set up. A mean accuracy of 0.8108 was found in the cross-validation findings, suggesting consistent performance across folds. After training on the whole training set, the final model obtained a test accuracy of 0.8112. With a score of 0.57, other metrics, like the Matthews Correlation Coefficient (MCC), also outperformed the random forest.

To efficiently learn from the properties of the dataset, the neural network model for the classification job uses a Sequential architecture with several dense layers. To avoid overfitting, the model has six hidden layers of 512, 256, 128, 64, 32, and 16 neurons each. Each hidden layer was followed by a dropout layer with a 30% dropout rate. The labels were changed to a one-hot encoding style and the input features were resized. The model was trained with early stopping to track validation loss. It was assembled using the Adam optimizer and categorical cross-entropy loss function. Based on evaluation, the model's accuracy on the test set was about 71.39%. The precision, recall, and F1-score metrics showed that the model performed equally well in both classes.

| Performance Metrics | Random Forest | **XGBoost** | Neural Network | LSTM | Logistic Regression |
|---|---|---|---|---|---|
| Accuracy | 74% | **79%** | 74% | 60% | 69.5% |
| Specificity | 70% | **73%** | 71% | 46% | 62% |
| Precision | 73% | **79%** | 74% | 61% | 69.5% |
| NPV | 73% | **75%** | 72% | 60% | 69% |
| MCC | 43% | **57%** | 47% | 20% | 39% |
| F1 Score | 71.4% | **79%** | 74% | 60% | 69.4% |
| Recall | 72% | **74%** | 74% | 60% | 69.4% |

*Table 7: Results for binary classifier 2*

An ROC curve, which contrasts the performance of five distinct models—Logistic Regression, Random Forest, XGBoost, Neural Network, and LSTM—was also used to demonstrate the results. The trade-off between each model's true positive rate (sensitivity) and false positive rate is shown by the ROC curve. Each model's performance is gauged by its area under the curve (AUC), where a higher AUC denotes better performance. Plotting shows that the XGBoost model (green curve) performs the best out of all the models, with the highest AUC of 0.86. AUCs of 0.81 for the Neural Network model (red curve), 0.78 and 0.75 for the Random Forest (orange curve) and Logistic Regression (blue curve) models, respectively, are the next highest. With an AUC of 0.64, the LSTM model (purple curve) performs the worst. A random classifier with an AUC of 0.5 is represented by the dashed diagonal line, which serves as a benchmark for comparison.

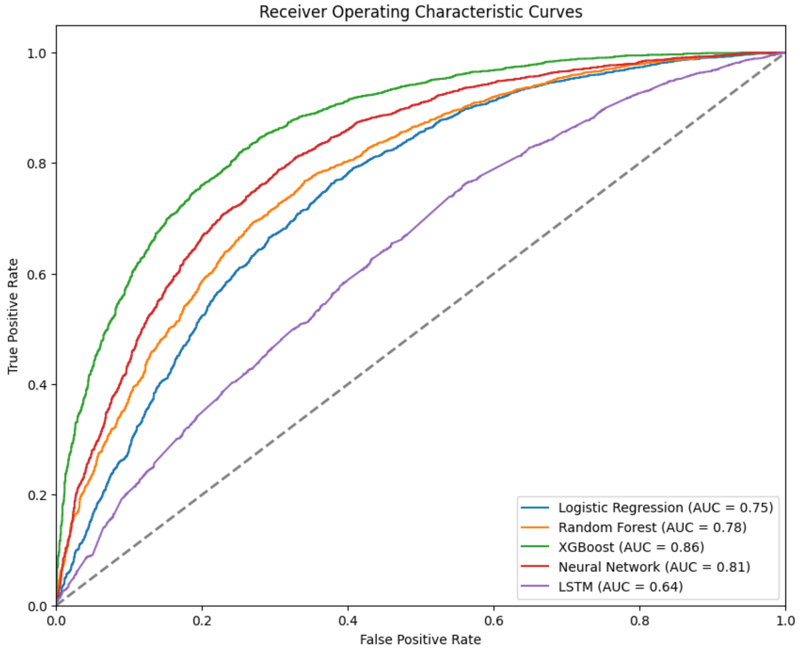


*Figure 9: ROC Curve and AUC values for the second binary classifier.*

## 5.4 Multi-class Classification

In addition to the two-layer binary classifier, this paper also explored multiclass classification using different machine learning models. 500 trees (n_estimators), a maximum depth of 20 (max_depth), a minimum of 10 samples needed to divide an internal node (min_samples_split), and a minimum of 5 samples needed at a leaf node (min_samples_leaf) were the hyperparameters that were set for a Random Forest model. On the training set, 10-fold cross-validation was used to assess the model's performance. Metrics like accuracy, ROC AUC, and Matthews Correlation Coefficient (MCC) were computed. Predictions were made on the test set and comparable assessment measures were supplied once the model had been trained on the whole training set. The loss function is based on entropy and Gini impurity, and the three categories being classified are increasing, decreasing, and neutral. I am referred to by class 0, N by class 1, and D by class 2.

The gradient boosting model, which balances model complexity and performance, is the second model used for multiclass classification. It uses 500 estimators, a learning rate of 0.1, and a maximum tree depth of 5. Cross-entropy loss was employed as the loss function for this multiclass classification problem, and accuracy and other pertinent metrics were used to assess performance.

The neural network utilized in this instance has a multi-layer design with three hidden layers that each include 128, 64, and 32 neurons. To provide non-linearity, each hidden layer uses the ReLU activation function. To prevent overfitting, the model further includes Dropout layers, which have a 0.5 dropout rate and randomly set a portion of the input units to zero during training. The output layer handles multiclass classification by predicting probability for three classes using the softmax activation function. To ensure effective training with adaptive learning rates, the model is optimized using the Adam optimizer after being assembled with the category cross-entropy loss function. It was trained with scaled data using the

StandardScaler, with a validation split of 0.2 to track performance throughout training, and across 100 epochs with a batch size of 64.

| Performance Metrics | **Gradient Boosting** | Random Forest | Neural Network |
|---|---|---|---|
| Accuracy | **69%** | 61% | 64% |
| Specificity | **68%** | 60% | 64% |
| Precision | **68%** | 60% | 64% |
| NPV | **31%** | 39% | 37% |
| MCC | **53%** | 41% | 46% |
| F1 Score | **68%** | 60% | 64% |
| Recall | **67%** | 62% | 64% |

*Table 8: Final model results for the multiclass classification*

The results of the performance of the models evaluated in the multiclass classification were also also plotted via the ROC AUC Curve as it can be seen in the figure. The ROC curve plot is a "one-vs-rest" (OvR) analysis for a multiclass classification problem involving three classes (Class 0, Class 1, and Class 2). In a "one-vs-rest" approach, the problem is broken down into multiple binary classification problems. For each class, the classifier is trained to distinguish that class against all other classes combined.

In this plot, each line represents the ROC curve for one class against the rest for each classifier (Gradient Boosting, Random Forest, and Neural Network). The ROC curves show how well each classifier can separate one class from the other two classes across various threshold settings.

For Gradient Boosting, the AUC values are 0.89, 0.78, and 0.88 for Class 0, Class 1, and Class 2, respectively. This means Gradient Boosting performs exceptionally well in distinguishing Class 0 and Class 2 from the rest, but slightly less so for Class 1. Similarly, for Random Forest, the AUC values are 0.82, 0.72, and 0.83 for the same classes, indicating solid but slightly lower performance compared to Gradient Boosting. The Neural Network shows AUC values of 0.84, 0.74, and 0.85, demonstrating robust and consistent performance across all classes.

The curves illustrate that Gradient Boosting excels in differentiating specific classes, while Neural Networks provide a balanced performance across all classes. Random Forest, though effective, tends to be slightly behind the other two in terms of AUC values. The AUC values above 0.7 for all models and classes indicate good overall discriminatory power, with the "one-vs-rest" approach providing detailed insights into each class's classification capability within the multiclass context. The confusion matrix are shown in the appendix.

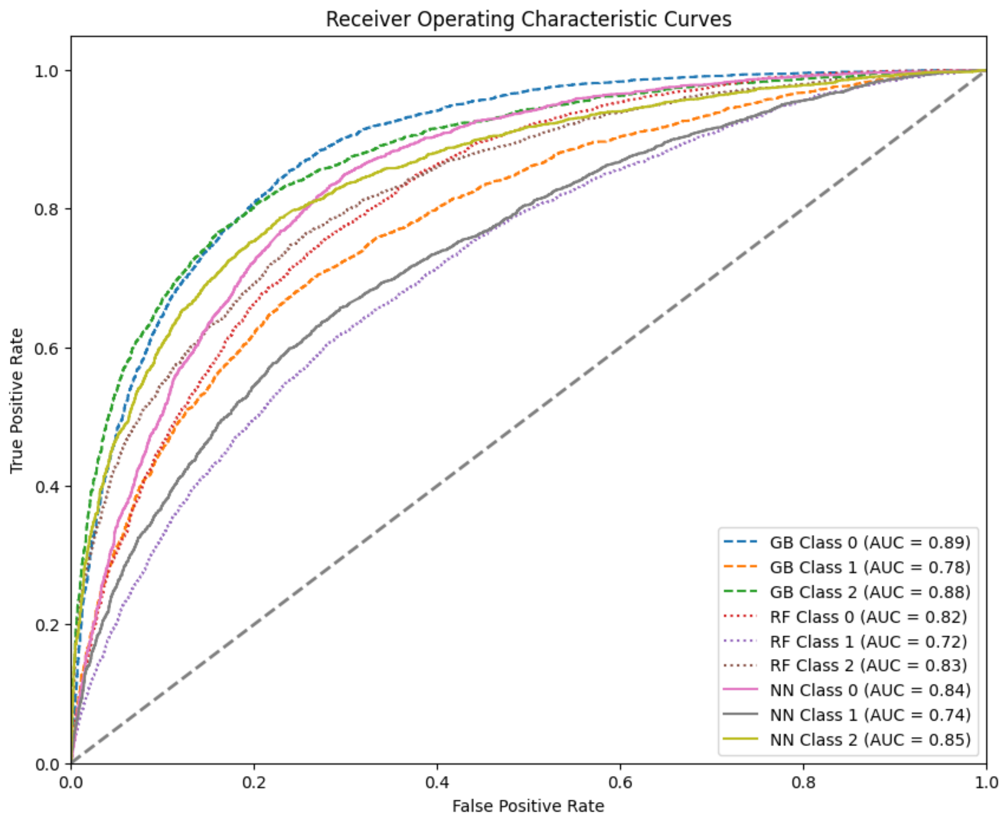


*Figure 10: ROC Curve values for the multiclass classification.*

## 5.5 Feature Importance

The study also carried out the feature importance. Although, the feature selection and evaluating the models with the important and selected features did not improve much the accuracy, it helped in identifying what are the top 10 important features in the two-layer binary classifier and in the multiclass classification. To select the important features, the gini index or loss function was used and the features that contributed lesser to the target variable were excluded and top 10 features were chosen. Likewise, in the XGBoost, Logistic Regression, and Neural Network models. The feature selection was carried out based on the features that have lower standard deviation or are highly correlated with the other features. The plots below show the feature importance for both the binary classifiers in the study.

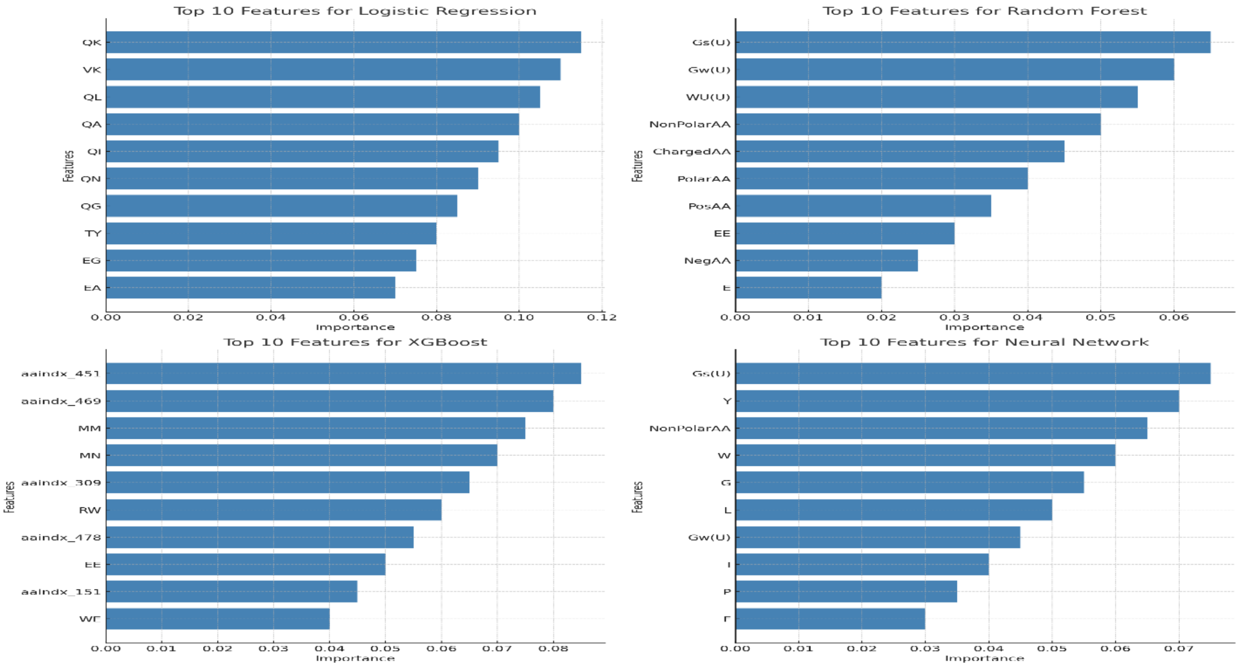


*Figure 11: Feature importance plot for second binary classifier*

Similarly feature selection was performed for the layer 1 and the results were obtained for the models used in the study.

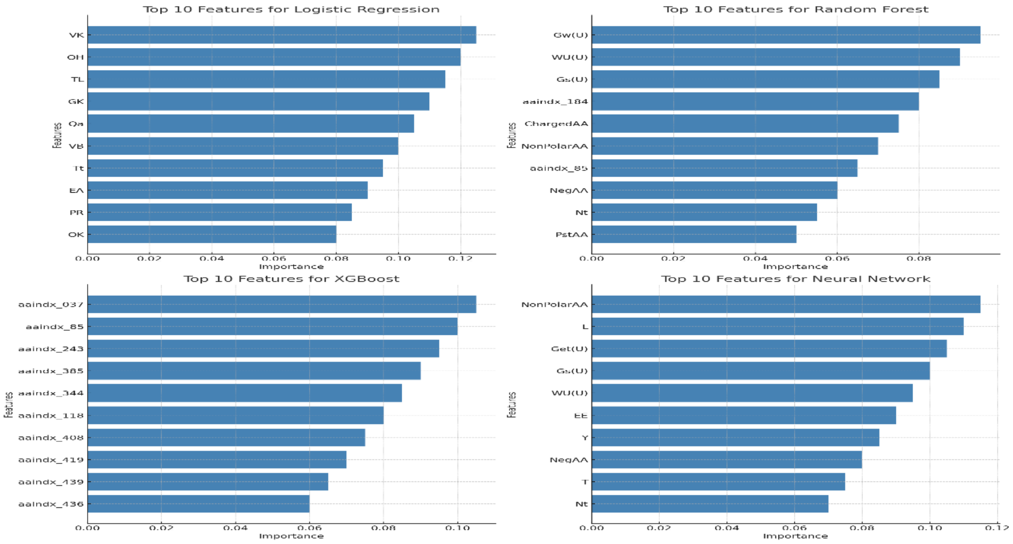


*Figure 12: Feature importance for the first binary classifier.*

Lastly, the study also performed the feature importance in the case of multiclass classification and obtained the results as shown in the figure 11.

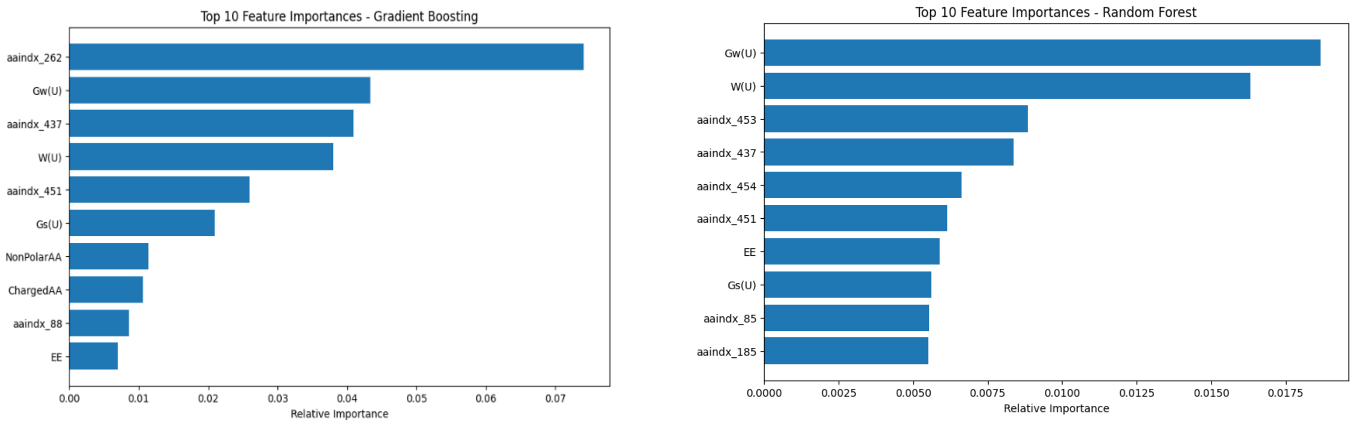


*Figure 13: Feature importance for the multi-class classification.*

## Discussion

The previous studies that also used the task of protein stability prediction got similar results, however, the models this study have used here have achieved better performance particularly in the case of binary classifiers. This could also indicate that there is more potential in the machine learning models to effectively contribute to improving the prediction of the protein stability. For instance, the Pont stab study achieved accuracy of 62% for layer 1 and 67% for layer 2 as binary classifier while this study achieved 78% and 79% respectively for the XGBoost. The tendency of having lower accuracy for the neutral category is also matching with the previous studies in the case of multiclass classification. When compared with the ABYSSAL study, the models also performed well in terms of accuracy in the multiclass classification where this study got 68% accuracy whereas the ABYSSAL had gotten 63% accuracy.

# 6. Conclusion and Future Research

This study used supervised machine learning algorithms to measure the performance for predicting the stability of protein cells due to amino acid variations. After carrying out the data exploration and cleaning different models were trained to predict the stability of protein cells. Due to lack of sufficient resources, the study used subsample of the data to check the performance of machine learning models. The results and performance of different models revealed that the machine learning algorithms can predict the stability of proteins accurately. While the study performed better than the previous methods like Pon-tstab, there is still room for improving the accuracy especially given the high dimensional settings of the data. The hyperparameter tuning helped in improving the accuracy and getting better results from the baseline models. This also demonstrates that if enough resources are allocated, the accuracy and other metrics could be improved which could be useful in the future.

The interesting finding that this study got was the importance of different models citing different features as important. During the feature selection and importance part, the study found that the three features introduced in the thermodynamic indices had more influence than the others. This could be indicative of the future research and studies using more of these features to accurately assess the performance of the stability of protein cells. XGBoost after the random search evaluation outperformed the other models such as neural network, random forest, logistic regression in the two layers binary classification model.

The study highlighted how important machine learning is to be improving our knowledge of protein stability. XGBoost, Random Forest, Neural Networks, and Long Short-Term Memory (LSTM) models were utilized to accurately forecast changes in stability. Of them, the models with the highest accuracy in binary classification (79%), and multiclass classification (69%), were XGBoost and Gradient Boosting. These findings highlight the potential of machine learning models to offer predictions that are more thorough and accurate than those obtained from more conventional techniques, which would improve the effectiveness of genetic variation annotation and medication development. Additionally, the outcomes demonstrate that the two-layer binary classifier outperforms the multiclass classification models in the cases of imbalance dataset.

The study's findings not only validate the effectiveness of machine learning in predicting protein stability but also suggest a promising future for these technologies in bioinformatics. By improving prediction accuracy, these methods can facilitate more targeted and effective therapeutic interventions, drive innovation in drug design, and provide valuable insights into protein functionality and disease mechanisms. As machine learning techniques continue to evolve, they hold the potential to revolutionize the field of protein engineering, leading to significant advancements in personalized medicine and biotechnology.

# 7. References

Pak, M.A., Dovidchenko, N.V., Sharma, S.M. and Ivankov, D.N., 2022. New mega dataset combined with deep neural network makes a progress in predicting impact of mutation on protein stability. bioRxiv. [Preprint]. Available at: https://doi.org/10.1101/2022.12.31.522396 [Accessed 1 March 2024].

AlQuraishi, M., 2021. Machine learning in protein structure prediction. Current Opinion in Chemical Biology, 65, pp.1-8.

Niroula, A. and Vihinen, M., 2016. Variation Interpretation Predictors: Principles, Types, Performance, and Choice. Human Mutation, 37(6), pp.579-597.

Pak, M.A. et al., 2023. New mega dataset combined with deep neural network makes a progress in predicting impact of mutation on protein stability. bioRxiv. [Preprint]. Available at: https://doi.org/10.1101/2022.12.31.522396 [Accessed 1 March 2024].

Tsuboyama, K. et al., 2023. Mega-scale experimental analysis of protein folding stability in biology and design. Nature, 620(7973), pp.434-444.

Vihinen, M., 2012. How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. BMC Genomics, 13(Suppl 4), S2.

Yang, Y. et al., 2013. Structure-based prediction of the effects of a missense variant on protein stability. Amino Acids, 44(3), pp.847-855.

Yang, Y. et al., 2018. PON-tstab: Protein Variant Stability Predictor. Importance of Training Data Quality. International Journal of Molecular Sciences, 19(4).

Niroula, A. and Vihinen, M., 2016. Variation Interpretation Predictors: Principles, Types, Performance, and Choice. Human Mutation, 37(6), pp.579-97. doi: 10.1002/humu.22987. Epub 2016 Apr 15. PMID: 26987456.

Blaabjerg, L.M., Kassem, M.M., Good, L.L., Jonsson, N., Cagiada, M., Johansson, K.E., Boomsma, W., Stein, A., Lindorff-Larsen, K., 2023. Rapid protein stability prediction using deep learning representations. eLife, 12, e82593.

Benevenuta, S., et al., 2023. Challenges in predicting stabilizing variations: An exploration. Frontiers in Molecular Biosciences, 9.

Benevenuta, S., et al. (2023). "Challenges in predicting stabilizing variations: An exploration." Frontiers in Molecular Biosciences **9**.

Birolo, G., et al. (2021). "Protein Stability Perturbation Contributes to the Loss of Function in Haploinsufficient Genes." Frontiers in Molecular Biosciences **8**.

Blaabjerg, L. M., et al. (2023). "Rapid protein stability prediction using deep learning representations." eLife **12**: e82593.

Boyer, S., et al. (2023). Predicting protein stability changes under multiple amino acid substitutions using equivariant graph neural networks.

Buric, F., et al. (2023). "The amino acid sequence determines protein abundance through its conformational stability and reduced synthesis cost." bioRxiv: 2023.2010.2002.560091.

Dara, S., et al. (2022). "Machine Learning in Drug Discovery: A Review." Artif Intell Rev **55**(3): 1947-1999.

Demidova, L. and I. Klyueva (2021). "The two-stage classification based on 1-SVM and RF classifiers." Journal of Physics: Conference Series **1727**(1): 012007.

Gong, J., et al. (2023). "THPLM: a sequence-based deep learning framework for protein stability changes prediction upon point variations using pretrained protein language model." Bioinformatics **39**(11).

Johnston, K. E., et al. (2023). "Machine Learning for Protein Engineering." ArXiv.

Kawashima, S., et al. (2007). "AAindex: amino acid index database, progress report 2008." Nucleic Acids Research **36**(suppl_1): D202-D205.

Khan, S. and M. Vihinen (2010). "Performance of protein stability predictors." Hum Mutat **31**(6): 675-684.

Li, B., et al. (2020). "Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks." PLoS Comput Biol **16**(11): e1008291.

Lodish, H. F., et al. (2016). Molecular Cell Biology. New York, W.H. Freeman and Company.

Nakagawa, H. and T. Tamada (2021). "Hydration and its Hydrogen Bonding State on a Protein Surface in the Crystalline State as Revealed by Molecular Dynamics Simulation." Frontiers in Chemistry **9**.

Pak, M. A., et al. (2023). "New mega dataset combined with deep neural network makes a progress in predicting impact of mutation on protein stability." bioRxiv: 2022.2012.2031.522396.

Requião, R. D., et al. (2017). "Protein charge distribution in proteomes and its impact on translation." PLOS Computational Biology **13**(5): e1005549.

Sanavia, T., et al. (2020). "Limitations and challenges in protein stability prediction upon genome variations: towards future applications in precision medicine." Comput Struct Biotechnol J **18**: 1968-1979.

Torng, W. and R. B. Altman (2017). "3D deep convolutional neural networks for amino acid environment similarity analysis." BMC Bioinformatics **18**(1): 302.

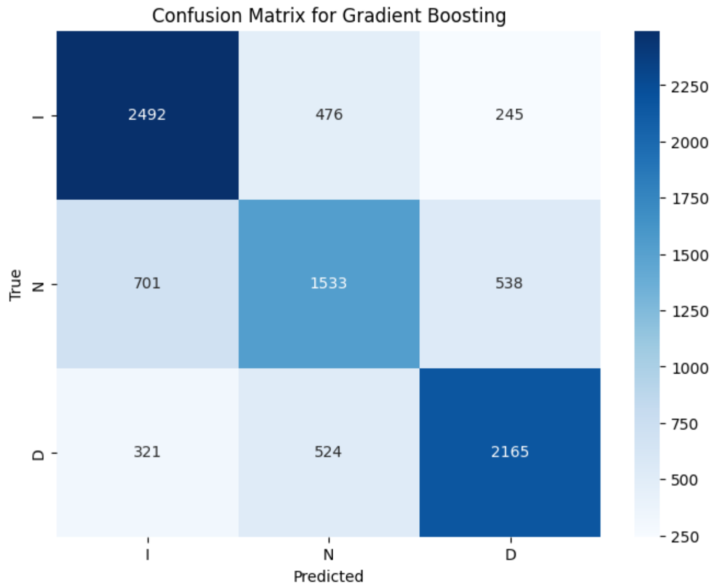
Tsuboyama, K., et al. (2023). "Mega-scale experimental analysis of protein folding stability in biology and design." Nature **620**(7973): 434-444.

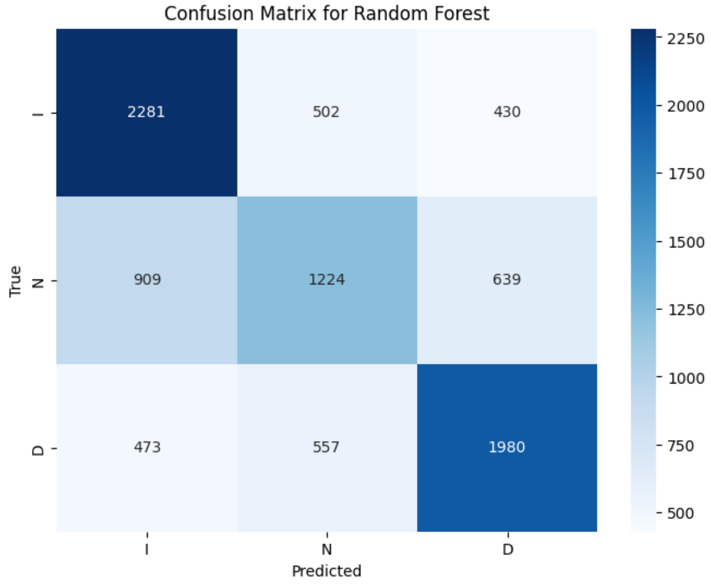
Yang, Y., et al. (2019). "ProTstab – predictor for cellular protein stability." BMC Genomics **20**(1): 804.

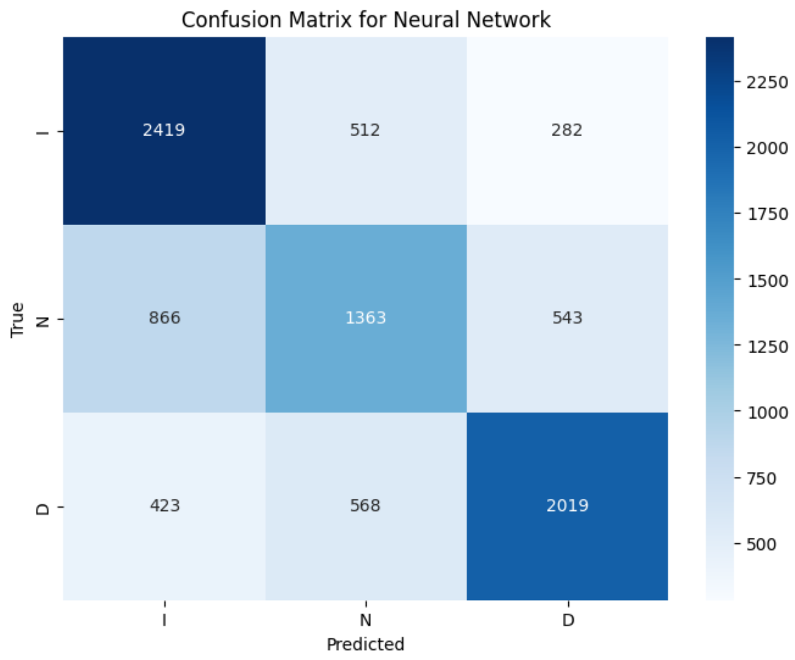
Yang, Y., et al. (2018). "PON-tstab: Protein Variant Stability Predictor. Importance of Training Data Quality." Int J Mol Sci **19**(4).

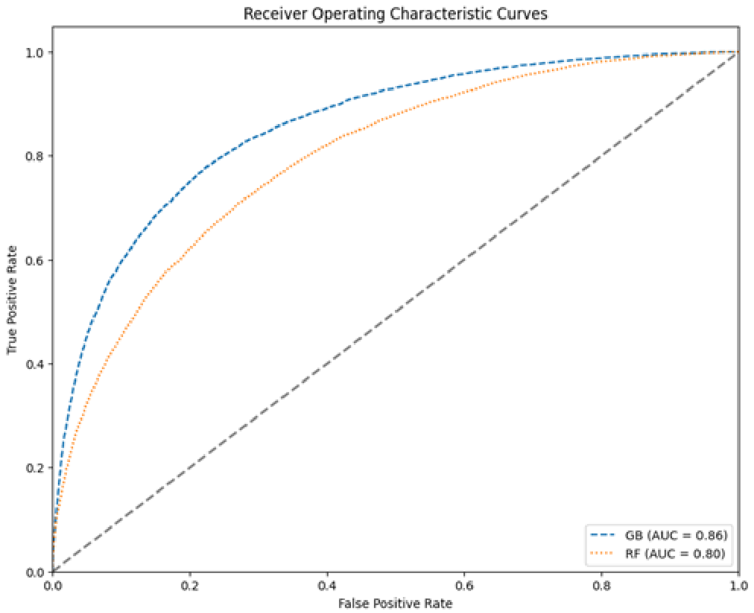
# Appendix

*For multiclass classification*









| **Performance Metrics** | **Gradient Boosting** | | **Random Forest** | |
|---|---|---|---|---|
| | **No FS** | **FS** | **No FS** | **FS** |
| Accuracy | 69% | 68% | 61% | 55% |
| Specificity | 68% | 68% | 60% | 58% |
| Precision | 68% | 67% | 60% | 57% |
| NPV | 31% | 32% | 39% | 35% |
| MCC | 53% | 52% | 41% | 40% |
| F1 Score | 68% | 66% | 60% | 58% |
| Recall | 70% | 70% | 83% | 81% |