HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 2 (2023)

Evaluating Machine Learning Models on Predicting Change in Enzyme Thermostability

Avnith Vijayram[1] and Jacklyn Luu[#]

[1]Thomas Jefferson High School for Science and Technology
[#]Advisor

ABSTRACT

Enzymes are efficient catalysts for biological reactions and can potentially be designed to speed up non-biological reactions, such as reactions in industrial processes. However, physically experimenting with new protein designs is time consuming, and an efficient method to predict protein stability is needed. Our research problem is finding the best machine learning model to predict the change in enzyme thermostability after a single point mutation in the amino acid sequence. We trained several machine learning models and found that the XGBoost model had the best performance with an R2 score of 0.593 (R2 score is a metric where higher is better and a perfect model would have a score of 1).

Introduction

Our research investigates the performance of different machine learning models on the problem of predicting the effects of a single point mutation in a protein on its thermostability. A single point mutation is when a single amino acid in the amino acid sequence is either deleted or replaced with another amino acid, and this can significantly change the protein structure and its thermostability. Thermostability is the ability of a protein to resist denaturation or changing shape after an increase or decrease of heat in the environment.

This research problem is important for the field of bioengineering, and more specifically enzyme engineering. Enzymes are very efficient catalysts for reactions and can potentially be used to speed up reactions in industrial processes. Engineered enzymes can also be used for a variety of purposes, including enzymes that help reduce pollution by catalyzing reactions that remove pollutants from the environment (Mousavi et al., 2021). The field of enzyme engineering works toward being able to design an enzyme to catalyze any specific reaction. Thus, it is important to be able to predict protein structure and stability.

A major problem faced in the bioengineering field is that most naturally occurring enzymes aren't suited to work in most industrial environments (Mousavi et al., 2021). Due to the difference in heat and sometimes in pH, those proteins will denature and will not function effectively. The ability to find proteins that are more thermostable, or proteins that can resist higher temperatures and keep their shape, would solve this problem and open the door for major advancements in this field.

Our research helps to predict the relative change in enzyme thermostability after a single point mutation. That is, we are predicting the difference in thermostability after a single amino acid is either inserted, deleted, or swapped for another amino acid. By doing this, we make the enzyme engineering process easier since we no longer must experimentally determine the change in thermostability of a new enzyme.

This is a supervised learning problem, meaning that the data used to contain the model will contain the target values (thermostability). The input data will be the pH and the amino acid sequence, which can be represented as quantitative data. In addition, this is a regression problem since our model will predict a quantitative value representing the change in thermostability.

Background

Several other approaches have been made to solve our research problem. One of these is the AlphaFold model, which doesn't solve the problem of thermostability prediction directly, but solves the related problem of enzyme structure prediction. Given a protein's sequence of amino acids, AlphaFold can predict the 3D structure that the protein will adopt with high accuracy. AlphaFold was first published in 2020 and won the biennial CASP protein structure prediction competition. It is a major leap in the field of bioinformatics, as it achieved high accuracy in predicting protein structure even at the atomic level (Jumper et al., 2021). In addition, it opened the door for further research into protein structure and related problems, such as protein thermostability prediction.

As our dataset comes from a Kaggle competition, some of the participants in that competition have shared their approaches. One of these participants, Chris Deotte, a Kaggle Grandmaster who placed 967th place in the competition, shared a Kaggle notebook containing the methods he used to solve the problem. His approach focused on finding "mutation groups" in the Kaggle data, where every pair of sequences in that group differed by at most 2 amino acids. This meant that all of them were a single point mutation of a "wild type" sequence, which can be determined easily after finding the group. By isolating these mutation groups, the model is able to easily learn the relative differences in thermostability of different types of mutations (Deotte, 2022).

Another technique mentioned in the Kaggle post was acquiring data from other outside sources. For example, Chris Deotte mentions using data from JinyuangSun's GitHub for one of the versions of the model. Additionally, using the AlphaFold database, he also found the protein structure of the wild type sequence for each of the mutation groups described above. By using these additional types of data, his model became more robust and accurate (Deotte, 2022).

Dataset

As mentioned previously, we used a dataset from the Kaggle competition, "Novozymes Enzyme Stability Prediction", which was hosted by Novozymes, a biotechnology company that finds enzymes in nature and optimizes them for use in industry (*Novozymes Enzyme Stability Prediction*, 2022). The dataset contained two features (amino acid sequence and pH), one target value (thermostability), and a total of 28981 samples. The dataset contained three parts. The first part was the dataset that would be used for training, which included both the features (amino acid sequence and pH) and the target value (thermostability). Since the original dataset had missing or incorrect data values, the Kaggle competition provided an updated training dataset with corrected values. Finally, there was a test set, which contained single point mutations of a single wild type protein. After training a model on the test set, a Kaggle competitor would use it on the test set and submit the predicted results to Kaggle. For our research purposes, however, the test set is not useful.

The Kaggle competition provided a Protein Data Bank (PDB) file of the predicted protein structure of the wild type used in the test set. We created a visualization of the protein, which can be seen in Figure 1, using the py3Dmol and biopython libraries and an online resource on GitHub (Engelberger et al., 2021).

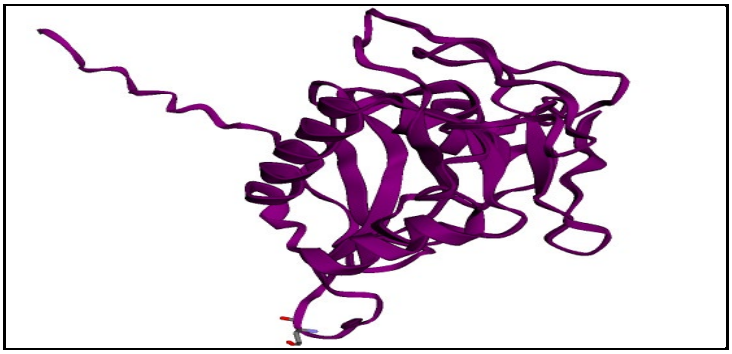

Figure 1. Protein Structure of Wild Type used in Kaggle Competition. Python libraries py3Dmol and biopython were used to create this visualization.

We also explored the distribution of training data's pH and thermostability. As seen in Figure 2, thermostability data is right skewed with a mean of about 55 while most of the pH values are just 7.0 with some outliers.

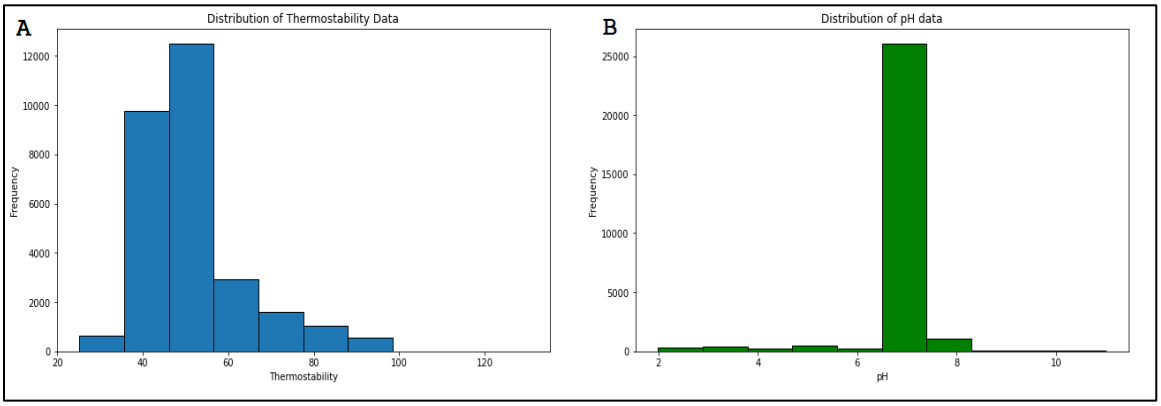

Figure 2. Distributions of Thermostability and pH Data. As seen in Figure 1a, the mean thermostability is 51.36 with standard deviation 12.05. In Figure 1b, the mean pH is 6.87 with standard deviation 0.79.

Methods

Data Preprocessing

After loading the updated dataset, we dropped unnecessary features in the data. One of those features was data source, which gave links to research articles that collected the data. We also dropped the sequence id feature in the training dataset because it was the same as the row number. However, for the Kaggle test set, we kept that column since it was necessary for submitting to the Kaggle competition.

Feature Engineering

We performed feature engineering on the dataset. Instead of representing the protein sequence data as a string of amino acids, we instead created a feature for each amino acid. Then, in each row, we counted the number of times each amino acid was present in the protein sequence. Through this, we had 26 new rows of numerical

data. Subsequently, we split the data into a train and a test dataset, with the test dataset being 20% of the original dataset and the training dataset being the remaining 80% of the data.

Model Development and Evaluation

For our baseline models, we used Linear Regression and Random Forest Regression. Linear regression works by fitting a line to the data so that the error is minimized. The model then uses this line to predict new values. It is one of the simplest machine learning models, which makes it a good baseline model. In Random Forest Regression, multiple Decision Tree models are bootstrapped into an Ensemble Learning Model. This means that several different Decision Tree Regressors are trained on the data, and the Random Forest model takes the average of the results. Through this process, the final result is more reliable since each of the Decision Tree models will make errors in different places, and the overall effect will cancel out (Beheshti, 2022).

In order to properly evaluate our models, we used three different metrics: mean squared error (MSE), mean absolute error (MAE), and R2 score. Error is measured by the difference of the predicted value and the actual value. Since error may be both positive and negative, mean squared error and mean absolute error both offer ways to ensure that the calculated error is always positive. In mean squared error, large errors are magnified compared to mean absolute error. The last metric, R2 score, also called the coefficient of determination, indicates the proportion of predicted values that are "near" the line (*Coefficient of Determination*, 2023).

After creating and evaluating our baseline models, we moved on to more specialized models, specifically XGBoost and Neural Network models. XGBoost is a type of ensemble model similar to Random Forest, but where using the Gradient Boosting algorithm, each additional Decision Tree model aims to correct the errors of the previous models. XGBoost is a popular model, and especially dominates in many Kaggle competitions (*XGBoost*, 2023). On the other hand, Neural Networks are also a popular model, and are inspired by how brains process information. Neural Networks are made of many neurons, which each do a simple task: they multiply the input by a certain amount, called a weight, and then add another amount, called a bias, and then outputs the new number. With many layers and connections between neurons, a Neural Network can be trained to recognize many patterns without being explicitly programmed (*Types of Neural Network algorithms*, 2022).

From our initial results of XGBoost and Neural Networks, it was clear that XGBoost was the better model, so we focused on the XGBoost model for hyperparameter tuning. We chose three of them to optimize: max_depth, n_estimators, and learning_rate. Max_depth refers to the maximum depth of each tree in the model; higher numbers for this parameter can lead to greater accuracy, but also overfitting. N_estimators is the number of Decision Tree models used in the XGBoost ensemble model. Learning_rate determines the "step size" of each iteration of the model. A lower learning rate means that a model will need more training rounds, but it also may end up performing better than a model with a higher learning rate (Martins, 2021). To tune these hyperparameters, we used the GridSearch technique with 3-fold cross-validation. GridSearch finds the best hyperparameters by exhaustively trying every combination of parameters and finding the best one, and 3 fold cross validation ensures that the hyperparameters found are not due to overfitting to specific train/test splits.

After finding the best hyperparameters for our XGBoost model, we trained a model using those parameters in order to submit our results to Kaggle.

Results and Discussion

Data Preprocessing

Our final training dataset contained 22 columns (21 features and one target value) and 28981 rows. Our final test set was also 22 columns and 2413 rows. Figure 3 shows the structure of each dataframe. The test set contains a column for seq_id instead of the target value tm since it is used for submitting model results to Kaggle.

A Enzyme Training Dataframe (28981, 22)

	pH	tm	A	C	D	E	F	G	H	I	...	M	N	P	Q	R	S	T	V	W	Y
0	7.0	75.7	45	1	13	30	13	38	3	14	...	8	5	18	6	25	11	14	37	4	3
1	7.0	50.5	28	0	10	52	6	18	4	13	...	2	6	8	22	30	14	12	13	3	3
2	7.0	40.5	50	9	27	32	21	65	11	16	...	6	15	20	25	31	33	30	30	3	16
3	7.0	47.2	20	5	19	29	12	16	7	10	...	2	9	16	9	10	16	19	14	3	4
4	7.0	49.5	86	14	78	78	32	84	40	71	...	31	65	128	54	63	148	120	124	16	47

B Enzyme Test Dataframe (2413, 22)

	seq_id	pH	A	C	D	E	F	G	H	I	...	M	N	P	Q	R	S	T	V	W	Y
0	31390	8	22	4	15	8	10	19	0	6	...	0	19	17	13	3	18	8	13	6	6
1	31391	8	22	4	15	7	10	19	0	6	...	0	19	17	13	3	18	8	13	6	6
2	31392	8	22	4	15	7	10	19	0	6	...	0	19	17	13	3	18	8	13	6	6
3	31393	8	22	5	15	7	10	19	0	6	...	0	19	17	13	3	18	8	13	6	6
4	31394	8	22	4	15	7	11	19	0	6	...	0	19	17	13	3	18	8	13	6	6

Figure 3. First 5 rows of Final Training and Test Dataframes. As seen in Figure 3A and 3B, both the training and test dataframe have 22 columns, and the majority of the columns correspond to amino acids. However, the training dataframe contains the target variable thermostability (tm), while the test dataframe instead has a seq_id column.

Model Development Results

Table 1 shows the performance of different models against various metrics (mean squared error, mean absolute error, and R2 score). Linear Regression was the worst performing model, followed by Random Forest Regression. Before hyperparameter tuning, our two best models were the XGBoost model and the Neural Network model. While the baseline XGBoost model had similar accuracy to the Random Forest Regression model, the Neural Network model was slightly better. However, when we experimented with both models, we found that while there wasn't much room for improvement in the Neural Network model, the XGBoost model had a slightly better performance with some better hyperparameters. Thus, we decided to tune only the XGBoost model.

Table 1. Model Results on Different Metrics

Model	Metric		
	MSE	MAE	R2
Linear Regression	122.014	8.337	0.131
Random Forest Regression	61.675	5.694	0.561
Neural Network	60.327	5.716	0.570
XGBoost	61.588	5.775	0.562
XGBoost with hyperparameter tuning	56.894	5.458	0.593

After multiple rounds of tuning, we concluded that: increasing max_depth almost always improved the model, but in the end, we used a value of 10, the best value of N_estimators is around 475, and the optimal learning rate is 0.05. Further rounds of tuning after this did not improve the model by much, so these parameters are likely close to optimal.

Conclusion

In our research, we trained and compared several machine learning models to predict changes in enzyme thermostability after single point mutations in the amino acid sequence. We chose several models (Linear Regression, Random Forest Regression, Neural Networks, and XGBoost) to train and test against several metrics (mean squared error, mean absolute error, and R2 score). As the Linear Regression and Random Forest Regression models are simpler models, they were useful as a baseline model but did not perform well compared to the latter two models. On Neural Network models and XGBoost models without hyperparameter tuning, the Neural Network model had a slightly better performance; however, we decided to tune the XGBoost model since it seemed that it had more room for improvement. After tuning the XGBoost model, we had a final model, XGBoost with hyperparameter tuning, which was significantly more accurate than the previous models.

The goal of our research was to determine the best methods and types of models to use in order to approach this problem, which is one of the main challenges currently faced in the field of enzyme engineering. Making progress on solving this problem means that we are one step closer to being able to design enzymes for a variety of purposes, such as catalyzing industrial reactions or helping to absorb pollutants from the environment.

Acknowledgments

I would like to thank my advisor Jacklyn Luu and Inspirit AI for their help and guidance throughout this project. I would also like to acknowledge other Kaggle competitors for sharing their ideas and code on the contest page.

References

Beheshti, N. (2022, March 2). *Random Forest Regression*. Towards Data Science. Retrieved February 26, 2023, from https://towardsdatascience.com/random-forest-regression-5f605132d19d

Deotte, C. (2022, September). *How to use Kaggle's train data*. Kaggle. Retrieved February 26, 2023, from https://www.kaggle.com/competitions/novozymes-enzyme-stability-prediction/discussion/358320

Engelberger, F., Galaz-davison, P., Bravo, G., Rivera, M., & Ramírez-sarmiento, C. A. (2021). Developing and implementing cloud-based tutorials that combine bioinformatics software, interactive coding, and visualization exercises for distance learning on structural bioinformatics. *Journal of Chemical Education*, *98*(5), 1801-1807. https://doi.org/10.1021/acs.jchemed.1c00022

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-paredes, B., Nikolov, S., Jain, R., Adler, J., . . . Silver, D. (2021). Highly accurate protein structure prediction with Alphafold. *Nature*, *596*(7873), 583-589. https://doi.org/10.1038/s41586-021-03819-2

Martins, D. (2021, May 14). *XGBoost: A complete guide to fine-tune and optimize your model*. Towards Data Science. Retrieved February 26, 2023, from https://towardsdatascience.com/xgboost-fine-tune-and-optimize-your-model-23d996fab663

Mousavi, S. M., Hashemi, S. A., Iman moezzi, S. M., Ravan, N., Gholami, A., Lai, C. W., Chiang, W.-H., Omidifar, N., Yousefi, K., & Behbudi, G. (2021). Recent advances in enzymes for the bioremediation of pollutants. *Biochemistry Research International*, *2021*, 1-12. https://doi.org/10.1155%2F2021%2F5599204

Novozymes Enzyme Stability Prediction. (2022). Retrieved from https://kaggle.com/competitions/novozymes-enzyme-stability-prediction

Coefficient of Determination - R2 score. (2023, January 10). GeeksforGeeks. Retrieved February 26, 2023, from https://www.geeksforgeeks.org/python-coefficient-of-determination-r2-score/

Types of Neural Network algorithms in Machine Learning. (2022, September 27). Omdena. Retrieved February 26, 2023, from https://omdena.com/blog/types-of-neural-network-algorithms-in-machine-learning

XGBoost. (2023, February 6). GeeksforGeeks. Retrieved February 26, 2023, from https://www.geeksforgeeks.org/xgboost/