CAFA6 Protein Function Prediction

Final Project Report

Student: Kit M. Kennedy

Course: Machine Learning CS 3120

Project Type: Kaggle Competition (Option A)

This project used the CAFA6 Kaggle competition as a way to practice building a full machine learning workflow from start to finish. The goal of the competition is to predict the biological functions of proteins using only their amino acid sequences. These functions come from the Gene Ontology, and each protein can have several of them at once, which makes this a multilabel classification problem.

Predicting protein function matters because doing it in a lab is slow and expensive. Machine learning can give early guesses that help scientists decide what to test. The data is messy, high-dimensional, and very imbalanced, which made this a good challenge for practicing real-world ML skills.

The dataset included protein sequence embeddings from a model called ESM2 (1280-dimensional vectors), GO term labels for part of the proteins, and the Gene Ontology hierarchy. Because the full training set was more than 140,000 proteins, I worked with a random sample of 3,000 to keep the runtime manageable. I also limited the label space to the 300 most common GO terms, since many of the others appear only a handful of times.

I converted each protein’s GO terms into a binary vector using MultiLabelBinarizer. To reduce the size of the embedding vectors and help the model train faster, I standardized the features and applied PCA to bring them down from 1280 dimensions to 128. After that, I split the data 80/20 into a training and validation set.

For the model, I used logistic regression in a One-vs-Rest setup, which means one classifier per GO term. It was simple, fairly fast, and a good baseline for a project like this. I used the saga solver, added balanced class weights, and increased the maximum number of iterations to help it converge. Since it’s multilabel, the model outputs probabilities, so I picked a threshold of 0.2 for deciding whether to assign each label. That produced predictions that were not too sparse or too dense.

On the validation set, the model achieved a micro F1 score of 0.076 and a macro F1 of 0.034, which is normal for a basic model on this competition. The prediction density was about 7 percent. The final submission file had a little over 3.2 million rows, and when I submitted it to Kaggle, it received an F-max score of 0.128. This confirmed that the pipeline worked correctly and matched the competition format.

There were a few challenges along the way. Logistic regression produced convergence warnings because the data is high-dimensional. The label imbalance was noticeable, since some GO terms appear hundreds of times and others only a few. Working through the steps of a multilabel problem also took some trial and error, and the output file was large enough that I had to be careful with the thresholding.

Even with these challenges, this project taught me a lot about handling large datasets, applying PCA, and building clean ML workflows inside the Kaggle environment. If I continued the project, I would try using more of the training data, create separate models for the different ontology categories, explore different thresholds, and possibly try stronger models like XGBoost or a small neural network. I would also take more time trying to figure out how to handle label imbalance.

Overall, the project met my main goals: understand the dataset, clean and prepare the data, build a functioning model pipeline, evaluate it properly, and make a valid Kaggle submission that scored on the leaderboard.