Classification of Family Types of Proteins

Name:	**Singh Arjita Satyaprakash**
Registration No./Roll No.:	21264
Institute/University Name:	IISER Bhopal
Program/Stream:	DSE
Problem Release date:	17-08-22
Date of Submission:	19-11-23

1 Introduction

This project focuses on classifying the family types of proteins. The given dataset encompass various attributes, such as ‘structure ID’ (unique 3D structure identifier), ‘experimental technique’ (structure insights), ‘residue count’ (amino acid number),“macromolecule type,’ ‘resolution’, etc. The given Raw data, consists of 125453 training instances and 13940 test instances. We have 2160 distinct classes or protein family types. Maximum number of data points (19010) belong to the class HYDROLASE; followed by TRANSFERASE (14405), OXIDOREDUCTASE (11326), LYASE (3946),etc. 693 family types of protein have 2 instances each.

2 Methods

2.1 Data Cleaning

The Raw Data was misaligned, in order to deal with this I created an empty column ‘miscellaneous’ at column index ‘0’ and shifted the wrongly aligned rows to the left so as to match the data in the cell with that of the column header. In the ‘phValue’ column there was a value ‘M Tris/HCl pH 7.0 0.05 M NaCl 0.03 M Lithiumsulfat’ at index 41235, this was because ‘0.1 M Tris/HCl pH 7.0 0.05 M NaCl 0.03 M Lithiumsulfat’ was separated after 0.1 in order to deal with this I had to further shift the row. In the ‘phValue’ column there was another value ‘724’, to fix this I replaced it with NaN.
For the label Encoding of ‘experimentalTechnique’, I first split the column at ‘, ’ because it referred to more than 1 technique being performed. Then I one-hot encoded the resulting DataFrame of separated values. Further, I have resolved the issue of duplicated columns in the one-hot encoded DataFrame by creating a new DataFrame with unique column names and combining it with the original DataFrame, in such a way that if either of the duplicated columns have value ‘1’, the unique column should also have the value ‘1’. I have then updated the original DataFrame, by removing the original ‘experimentalTechnique’ column and adding the modified one hot encoded DataFrame with unique column names.
This column had no missing values.
I have imputed the missing values of ‘macromoleculeType’ with the mode value i.e. ‘Protein’(109134 out of 12543 instances). Then I have followed a similar strategy for one hot encoding as above by separating the values at ‘’.
I have dropped the columns ‘miscellaneous’ as it only contained 4704 non null values, ‘structureId’ as it contained 125055 unique values, ‘pdbxDetails’ as it contained 81814 unique values, ‘crystallizationMethod’ as it contains 518 unique values and 31.66% missing values and ‘publicationYear’ as it seemed irrelevant for protein type classification.
for the numerical features, with mean (if the feature is normally distributed) or with median (if it is skewed). I plan on dropping the feature “publication year” as by visual inference it seems irrelevant for classification. I plan on dropping the feature “phValue” and “crystallizationTempK” as about

25.30% and 31.16% of the instances for the respective features have missing values. I'm dropping the first column 'miscellaneous' as it only contained 4704 non-null values.
I haved imputed the missing values of 'phValue' with mean, 'resolution' with mean, 'crystallizationTempK' with mean, 'densityMatthews' with mean and 'densityPercentSol' with mean.
On analyzing the various feature scaling technique such as Standard Scaler, MinMax Scaler and Robust Scaler, I got the best results with Robust Scaler.

2.2 Under Sampling

I also followed another strategy to deal with the problem of class imbalance, I randomly selected 4 instances of each class if possible otherwise 2 instances each and created a synthetic dataframe (5544 rows), on this I trained various models. The results differed for every run.
It is likely that this method doesn't capture all the necessary patterns in the dataset and would not perform great on the test data.

LINK TO GITHUB REPOSITORY

3 Results

Proceeding with Robust Scaler since both algorithms improved performance for it. For SVM and Random Forest, session crashed on colab and it couldn't allocate storage on device.
Since Multinomial naive bayes does not take negative values, for it I have used MinMaxScaler and the result is

Model		**Precision**	**Recall**	**F1-Score**
Naive Bayes	Accuracy			18%
	Weighted Average	8%	18%	7%
	Macro Average	1%	0%	0%
SVM	Accuracy			20%
	Weighted Average	15%	20%	9%
	Macro Average	1%	1%	1%

Table 1: After Scaling

Random forest could not run as it couldn't allocate the required storage.

The optimal configuration for the Decision Tree classifier involves using MinMaxScaler for feature scaling, SelectKBest with a k value of 25 and the chi squared score function for feature selection, and a DecisionTreeClassifier with specified parameters including ccp_alpha of 0.009, entropy criterion, maximum depth of 40, and 'sqrt' for maximum features.The results are as follows
Accuracy: 19%
Precision: 0.73%
Recall: 1.08 %
F1-Score: 0.75%

For 5 folds Decision Tree Classifier's confidence score (0.182) is poor

Model		**Precision**	**Recall**	**F1-Score**
Decision Tree	Accuracy			36%
	Weighted Average	36%	36%	36%
	Macro Average	18%	19%	18%
KNN	Accuracy			20%
	Weighted Average	20%	20%	19%
	Macro Average	6%	6%	5%
Naive Bayes	Accuracy			4%
	Weighted Average	11%	4%	5%
	Macro Average	1%	2%	1%
Logistic Regression	Accuracy			15.97%
	Weighted Average	%	4%	5%
	Macro Average	1%	2%	1%

Table 2: Before Scaling

Model	**Scaling Technique**	**Accuracy**
Decision Tree	MinMax	36%
	StandardScaling	36
	Robust Scaler	37%
KNN	MinMax	24%
	StandardScaling	27
	Robust Scaler	28%

Table 3: Scaling

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Random Forest	44.54%	44.36%	44.54%	42.57%
SVC	7.93%	6.93%	7.94%	6.94%
Logistic Regression	4.33%	3.13%	4.33%	3.37%
Decision Tree	39.76%	38.03%	39.76%	37.29%
KNN	15.78%	12.50%	15.78%	13.00%

Table 4: Under Sampling

4 Conclusion

Classification of family types of protein via Machine Learning can help in drug discovery, protein engineering, personalized medicine, diagnostics, and agriculture. By analyzing protein sequences, structures, and interactions, machine learning can identify novel drug targets. It can reduce the number of experiments a researcher has to do in order to identify a particular protein.