Novozymes Enzyme Stability Prediction

Help identify the thermostable mutations in enzymes

Yu Tianxiong

Economics Research, EEP

tyuy@edu.hse.ru

Table of Contents

Background

Goal of the Competition:

- Enzymes are proteins that act as catalysts in the chemical reactions of living organisms. The goal of this competition is to **predict the thermostability of enzyme variants**. The experimentally measured thermostability (melting temperature) data includes natural sequences, as well as engineered sequences with single or multiple mutations upon the natural sequences.

Prize Money: $ 25,000

Background
timeline

- September 21, 2022 - Start Date;
- December 27, 2022 - Entry Deadline. You must accept the competition rules before this date in order to compete;
- December 27, 2022 - Team Merger Deadline. This is the last day participants may join or merge teams;
- January 3, 2023 - Final Submission Deadline.

Background

competition

In this competition, you are asked to develop models that can predict the ranking of protein stability (as measured by melting point, t_m) after single-point amino acid mutation and deletion.

Novozymes(a company) finds enzymes in nature and optimizes them for use in industry.

- In industry, enzymes replace chemicals and accelerate production processes;
- They help our customers make more from less, while saving energy and generating less waste;
- ...

Background

challenge of predicting stability

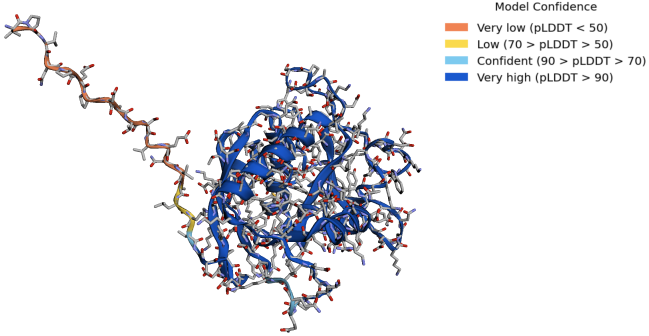
However, many enzymes are only marginally stable, which limits their performance under harsh application conditions.

Novozymes finds enzymes in nature and optimizes them for use in industry.

Computational protein stability prediction based on physics principles such as FoldX, Rosetta, and others. Recently, many machine learning methods were proposed to predict the stability impact of mutations, more and more protein structures are being solved thanks to the recent breakthrough of AlphaFold2. However, accurate prediction of protein thermal stability remains a great challenge.

Introduction

e.g. Alphafold2 prediction of wildtype 3d structure



Introduction

a brief intro on related knowledge

- **Enzymes** are proteins that act as catalysts in the chemical reactions of living organisms, it means that enzymes accelerate reaction speed, modifying substances called substrates, and the substrates which are chosen to bind with the enzymes to be modified, will depend in each enzyme. normally enzymes are proteins but also can be RNA;

- **Proteins** are large, complex molecules that play many critical roles in the body. They do most of the work in cells and are required for the structure, function, and regulation of the body's tissues and organs;

Introduction

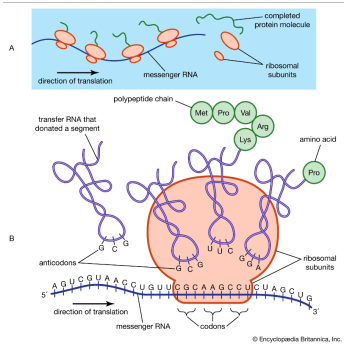
a brief intro on related knowledge

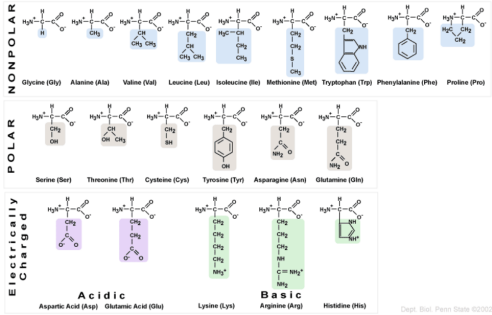
- Protein is made from twenty-plus basic building blocks called **amino acids**;
- An **amino acid** is an organic molecule that is made up of a basic amino group (NH2), an acidic carboxyl group (COOH), and an organic R group (or side chain) that is unique to each amino acid;
  - Amino Acid structure (particularly the R-Group) is determined by a particular codon (triplet of Nucleotides).
- There are only 5 types of **nucleotides**

Introduction

a brief intro on related knowledge

Graphically explanation





Introduction
summary

- To make a **protein** we use instructions (DNA/RNA, Nucleotides), to build up the protein chain by adding one **amino acid** at a time (in our instructions each codon tells us what amino acid comes next). The stability of protein is releted to natural sequences, their three dimensional structures and environments.

Dataset description

- In this competition, you are asked to develop models that can predict the ranking of protein thermostability (as measured by melting point, tm) after single-point amino acid mutation and deletion.

- For the training set, the protein thermostability data includes natural sequences, engineered sequences with single or multiple mutations upon the natural sequences. The data are mainly from different sources of published studies such as Meltome atlas—thermal proteome stability across the tree of life. Many other public datasets exist for protein stability; please see the competition Rule 7C for external data usage requirements.

File description

- **train.csv** - the training data, with columns as follows:
  - seq_id: unique identifier of each protein variants
  - protein_sequence: amino acid sequence of each protein variant. The stability (as measured by tm) of protein is determined by its protein sequence
  - pH: the scale used to specify the acidity of an aqueous solution under which the stability of protein was measured
  - data_source: source where the data was published
  - tm: target column. Since only the spearman correlation will be used for the evaluation, the correct prediction of the relative order is more important than the absolute tm values

File description

- **train_updates_20220929.csv** corrected rows in train, please see this forum post for details
- **test.csv** the test data; your task is to predict the target tm for each protein_sequence (indicated by a unique seq_id)
- **sample_submission.csv** a sample submission file in the correct format, with seq_id values corresponding to test.csv
- **wildtype_structure_prediction_af2.pdb.csv** the 3 dimensional structure of the enzyme listed above, as predicted by AlphaFold

File description

train.csv

	seq_id	protein_sequence	pH	data_source	tm
0	0	AAAAKAAALALLGEAPEVVDIWLPAGWRQPFRVFRLERKGDGVLVG...	7.0	doi.org/10.1038/s41592-020-0801-4	75.7
1	1	AAADGEPLHNEEERAGAGQVGRSLPQESEEQRTGSRPRRRRDLGSR...	7.0	doi.org/10.1038/s41592-020-0801-4	50.5
2	2	AAAFSTPRATSYRILSSAGSGSTRADAPQVRRLHTTRDLLAKDYYA...	7.0	doi.org/10.1038/s41592-020-0801-4	40.5
3	3	AAASGLRTAIPAQPLRHLLQPAPRPCLRPFGLLSVRAGSARRSGLL...	7.0	doi.org/10.1038/s41592-020-0801-4	47.2
4	4	AAATKSGPRRQSQGASVRTFTPFYFLVEPVDTLSVRGSSVILNCSA...	7.0	doi.org/10.1038/s41592-020-0801-4	49.5

Data preprocessing
procedure

- Substitue old dataset with updated ones
- Consider whether some of the data_source providers are unreliable
  - add dummy
- Missing values
- Abnormal values

Data preprocessing
feature engineering

- Dealing with protein_sequence
  - frequency for each amino acid
  - length for protein_sequence
  - different kinds of amino acids
  - relative position for amino acid
  - multicollinearity for relative positions
  - mutation

Data preprocessing

processed train data

	seq_id	protein_sequence	pH	data_source	tm	A	C	D	E	F	...	SLL	SLS	SSA	SSG	SSL	SSS	TLL	VAA	VL
0	0	AAAAKAAALALLGEAPEVVDIWLPAGWRQPFRVFRLERKGDGVLVG...	7.0	1	75.7	45	1	13	30	13	...	2	0	0	0	0	0	0	0	
1	1	AAADGEPLHNEERAGAGGVGRSLPQESEEQRTGSRPRRRRDLGSR...	7.0	1	50.5	28	0	10	52	6	...	0	0	0	0	0	0	0	0	
2	2	AAAFSTPRATSYRILSSAGSGSTRADAPQVRRLHTTRDLLAKDYYA...	7.0	1	40.5	50	9	27	32	21	...	0	0	1	0	0	0	0	0	
3	3	AAASGLRTAIPAQLRLHLLQPAPRPCLRPFGLLSVRAGSARRSGLL...	7.0	1	47.2	20	5	19	29	12	...	0	0	0	0	0	0	0	0	
4	4	AAATKSGPRRQSQGASVRTFTPFYFLVEPVDTLSVRGSSVILNCSA...	7.0	1	49.5	86	14	78	78	32	...	0	1	1	1	2	3	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
31385	31385	YYMYSGGGSALAAGGGAGRKGDWNDIDSIKKKDLHHSRGDEKAQG...	7.0	1	51.8	33	12	38	31	18	...	0	0	0	0	0	0	0	0	
31386	31386	YYNDQHRLSSYSVETAMFLSWERAIVKPGAMFKKAVIGFNCNVDLL...	7.0	1	37.2	37	5	21	29	22	...	0	0	0	1	0	1	0	0	
31387	31387	YYQRTLGAELLYKSFGEMPKSAQDSAENCPSGMQFPDTAIAHANV...	7.0	1	64.6	13	1	7	7	7	...	0	0	0	0	0	0	0	0	
31388	31388	YYSFSDNITTVFLSRQAIDDDHSLSLGTISDVVESENGVVAADDAR...	7.0	1	50.7	47	5	34	36	23	...	0	1	0	1	0	0	0	2	
31389	31389	YYVPDEYWQSLEVAHKLTFGYGYLTWEWVQGIRSYVYPLLIAGLYK...	7.0	1	37.6	34	5	15	32	26	...	1	0	0	0	0	0	0	0	

28981 rows × 191 columns

Data preprocessing

processed test data

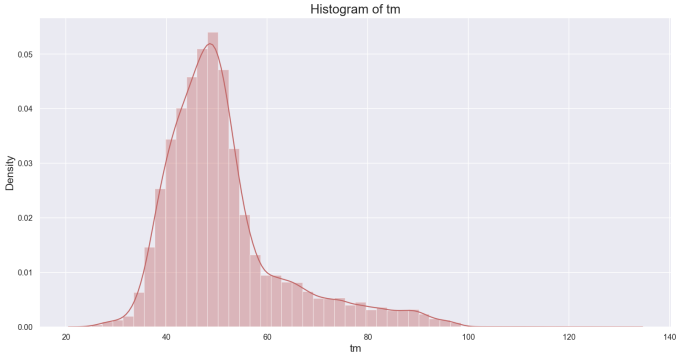
	seq_id	protein_sequence	pH	data_source	A	C	D	E	F	G	...	SLL	SLS	SSA	SSG	SSL	SSS	TLL	VAA	VLA	VLL
0	31390	VPVNPEPDATSVENVAEKTGSGDSQSDPIKADLEVKGQSALPFDVD...	8	1	22	4	15	8	10	19	...	0	0	0	0	1	0	0	0	0	0
1	31391	VPVNPEPDATSVENVAKKTGSGDSQSDPIKADLEVKGQSALPFDVD...	8	1	22	4	15	7	10	19	...	0	0	0	0	1	0	0	0	0	0
2	31392	VPVNPEPDATSVENVAKTGSGDSQSDPIKADLEVKGQSALPFDVDC...	8	1	22	4	15	7	10	19	...	0	0	0	0	1	0	0	0	0	0
3	31393	VPVNPEPDATSVENVALCTGSGDSQSDPIKADLEVKGQSALPFDVD...	8	1	22	5	15	7	10	19	...	0	0	0	0	1	0	0	0	0	0
4	31394	VPVNPEPDATSVENVALFTGSGDSQSDPIKADLEVKGQSALPFDVD...	8	1	22	4	15	7	11	19	...	0	0	0	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2408	33798	VPVNPEPDATSVENVILKTGSGDSQSDPIKADLEVKGQSALPFDVD...	8	1	21	4	15	7	10	19	...	0	0	0	0	1	0	0	0	0	0
2409	33799	VPVNPEPDATSVENVLLKTGSGDSQSDPIKADLEVKGQSALPFDVD...	8	1	21	4	15	7	10	19	...	0	0	0	0	1	0	0	0	0	1
2410	33800	VPVNPEPDATSVENVNLKTGSGDSQSDPIKADLEVKGQSALPFDVD...	8	1	21	4	15	7	10	19	...	0	0	0	0	1	0	0	0	0	0
2411	33801	VPVNPEPDATSVENVPLKTGSGDSQSDPIKADLEVKGQSALPFDVD...	8	1	21	4	15	7	10	19	...	0	0	0	0	1	0	0	0	0	0
2412	33802	VPVNPEPDATSVENVWLKTGSGDSQSDPIKADLEVKGQSALPFDVD...	8	1	21	4	15	7	10	19	...	0	0	0	0	1	0	0	0	0	0

2413 rows × 190 columns

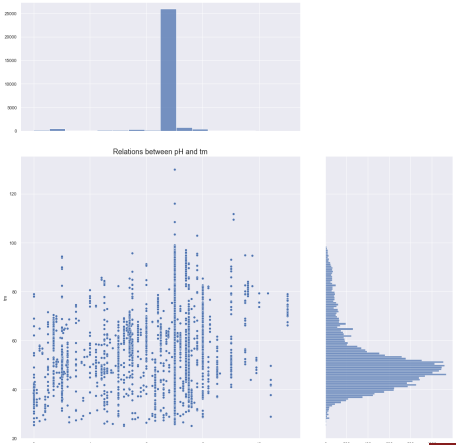
Statistical analysis

	pH	length	kinds	tm	position	wildtype	mutation
count	28981.0000	28981.0000	28981.0000	28981.0000	28981.0000	28981.0000	28981.0000
mean	6.8737	450.4686	19.7070	51.3600	1.0104	4.1522	1.5093
std	0.7894	415.1590	0.7089	12.0567	0.1137	0.5260	1.4637
min	1.9900	5.0000	5.0000	25.1000	1.0000	-3.5000	-4.5000
25%	7.0000	212.0000	20.0000	43.6000	1.0000	4.2000	1.9000
50%	7.0000	351.0000	20.0000	48.8000	1.0000	4.2000	1.9000
75%	7.0000	537.0000	20.0000	54.6000	1.0000	4.2000	1.9000
max	11.0000	8798.0000	20.0000	130.0000	4.0000	4.2000	4.5000

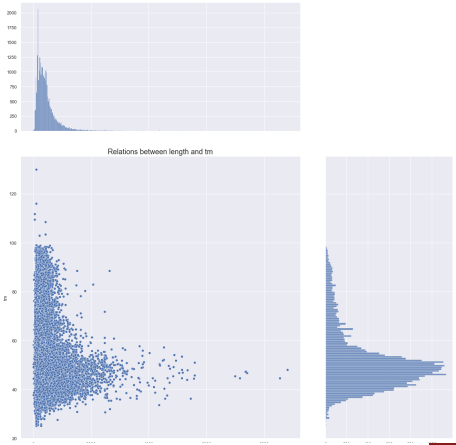
Check tm distribution



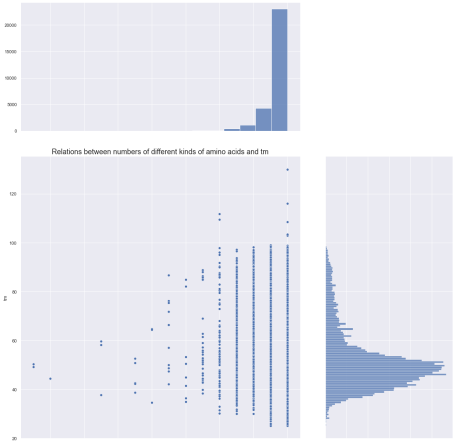
Relations between pH and tm



Relations between length and tm



Relations between amino acid kinds and tm



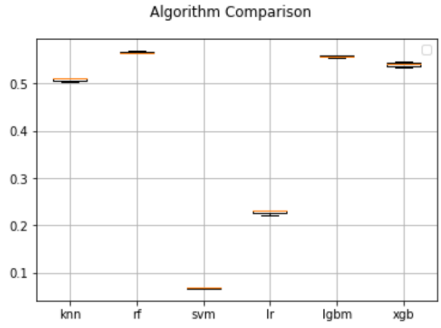
Modeling
start with some basic models

- Baseline
  - ”knn”, KNeighborsRegressor()
  - ”rf”, RandomForestRegressor()
  - ”svm”, SVR()
  - ”lr”, LinearRegression()
  - ”lgbm”, LGBMRegressor()
  - ”xgb”, XGBRegressor()

Modeling

boxplot algorithm comparison

scoring = R2, 3-fold cross validation indicates merely svm and lr may not be a good fit



Modeling

searching better hyperparameters for each model using GridSearchCV

- Models
  - RF
  - LGBM
  - XGB
  - KNN
- Criteria
  - R2
  - MSE
  - MAE

Modeling

algorithm comparison after tuning

after fine tuning, in MSE, MAE and R2 sense, lgbm $<$xgb $<$rf $<$knn, lgbm performs the best

	RF	XGB	LGBM	KNN
R2	0.569736	0.578673	0.597628	0.521262
best_params	{'max_depth': 25, 'n_estimators': 250}	{'learning_rate': 0.1, 'n_estimators': 200}	{'learning_rate': 0.05, 'n_estimators': 250, '...	{'n_neighbors': 10, 'weights': 'distance'}
MSE	61.813245	60.529299	57.806156	68.777118
MAE	5.714213	5.77645	5.564431	5.943662

Modeling

ensemble

consider ensemble two models, from here first naively pick knn and lgbm for balancing the model performance and the executing speed, ensemble model stacks as follows

```
StackingRegressor(cv=5,
                  estimators=[('XGB',
                               Pipeline(steps=[('columntransformer',
                                               ColumnTransformer(transformers=[('passthrough',
                                                                                'passthrough',
                                                                                ['pH',
                                                                                 'length',
                                                                                 'kinds',
                                                                                 'position',
                                                                                 'wildtype',
                                                                                 'mutation'])])),
                                               ('xgbregressor',
                                                XGBRegressor(base_score=None,
                                                             booster=None,
                                                             callbacks=None,
                                                             colsample_bylevel=None,
                                                             colsample_bynode=None,
                                                             colsample_bytree=None,
                                                             early_sto...
                                                             reg_lambda=None, ...))])),
                              ('LGBM',
                               Pipeline(steps=[('columntransformer',
                                               ColumnTransformer(transformers=[('passthrough',
                                                                                'passthrough',
                                                                                ['data_source',
...
                                                                                 'DL',
                                                                                 'EA', ...])])),
                                               ('lgbmregressor',
                                                LGBMRegressor())])],
                  final_estimator=LinearRegression())
```

Modeling
ensemble2

then let's try another approach, ensemble xgb and lgbm, with same stack

```
StackingRegressor(cv=5,
                  estimators=[('XGB',
                               Pipeline(steps=[('columntransformer',
                                               ColumnTransformer(transformers=[('passthrough',
                                                                                'passthrough',
                                                                                ['pH',
                                                                                 'length',
                                                                                 'kinds',
                                                                                 'position',
                                                                                 'wildtype',
                                                                                 'mutation'])])),
                                               ('xgbregressor',
                                                XGBRegressor(base_score=None,
                                                             booster=None,
                                                             callbacks=None,
                                                             colsample_bylevel=None,
                                                             colsample_bynode=None,
                                                             colsample_bytree=None,
                                                             early_sto...
                                                             reg_lambda=None, ...))])),
                              ('LGBM',
                               Pipeline(steps=[('columntransformer',
                                               ColumnTransformer(transformers=[('passthrough',
                                                                                'passthrough',
                                                                                ['data_source',
...
                                                                                 'DL',
                                                                                 'EA', ...])])),
                                               ('lgbmregressor',
                                                LGBMRegressor())])],
                  final_estimator=LinearRegression())
```

Modeling
model comparison

	RF	XGB	LGBM	KNN	ensemble	ensemble_2
R2	0.569736	0.578673	0.597628	0.521262	0.565163	0.572461
best_params	{'max_depth': 25, 'n_estimators': 250}	{'learning_rate': 0.1, 'n_estimators': 200}	{'learning_rate': 0.05, 'n_estimators': 250, '...	{'n_neighbors': 10, 'weights': 'distance'}	{'kNN__kneighborsregressor__weights': 'distanc...	{'XGB__xgbregressor__n_estimators': 100, 'XGB_...
MSE	61.813245	60.529299	57.806156	68.777118	62.470221	61.421818
MAE	5.714213	5.77645	5.564431	5.943662	5.855617	5.798808

Modeling

NN

another thought is to use NN for prediction consider the sophisticated structure of protein sequence, with another way of feature engineering

	pH	tm	data_source	0	1	2	3	4	5	6	...	211	212	213	214	215	216	217	218	219	220
0	7.0	49.7		1	1	1	2	3	4	5	6	...	0	0	0	0	0	0	0	0	0
1	7.0	45.1		1	1	1	11	11	3	15	3	...	1	3	3	3	14	7	0	0	0
2	7.0	62.8		1	1	1	9	11	9	11	9	...	0	0	0	0	0	0	0	0	0
3	7.0	36.3		1	1	1	10	6	11	18	1	...	0	0	0	0	0	0	0	0	0
4	7.0	83.0		1	1	1	12	1	12	11	12	...	0	0	0	0	0	0	0	0	0

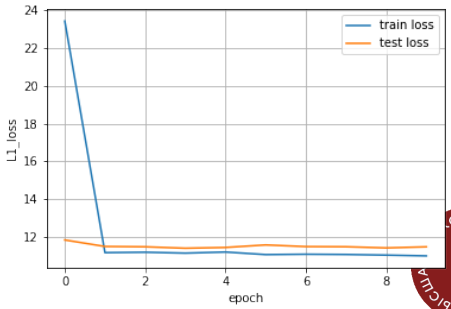
5 rows × 224 columns

Modeling

NN, structure and loss

criterion = nn.L1Loss() optimizer = torch.optim.SGD(model_nn.parameters(), lr=1e-3)

```
NESP(
  (embedding): Embedding(222, 256)
  (lstm1): LSTM(256, 128, batch_first=True, bidirectional=True)
  (fc1): Linear(in_features=256, out_features=128, bias=True)
  (fc2): Linear(in_features=128, out_features=1, bias=True)
  (act): ReLU()
  (drop): Dropout(p=0.2, inplace=False)
)
```



Submission

- nn: 0.02, basically capture nothing at all, need to be modified in the future
- lgbm: 0.16, do have some boosts comparing with ver.1.5(midterm report)

Another approach

physics principle based methods

There are also other physics principle based methods which can predict protein stabilities such as ESM, EVE and Rosetta etc., without using the provided training set.

What's interesting is these approach are more likely to get higher scores over 0.2, better than models trained based on given train_dataset, which would be discussed later.

How to get high scores in real test data ”easily”

with the help of previous works, one can try to combine several and try to contribute to the overall score, e.g. if one ensemble rosetta, rmsd, thermonet, plddtdiff, sasaf, plddt, demask, ddG, blosum (basically physic based theorems) and with proper weight, can easily reach a score of over 0.5

- stack former predictions
- transfer learning, etc.

for example, I can get a score of 0.6 through weighting other’s preds, which reaches the top 5% on leaderboard, however maybe the previous analyse is more valuable somehow

126 TianxiongYu 0.603 16 1d

Results and Discussion
submission

- nn: 0.02, basically capture nothing at all, need to be modified in the future
- lgbm: 0.16, do have some boosts comparing with ver.1.5(midterm report)

Results and Discussion

model comparison

	RF	XGB	LGBM	KNN	ensemble	ensemble_2
R2	0.569736	0.578673	0.597628	0.521262	0.565163	0.572461
best_params	{'max_depth': 25, 'n_estimators': 250}	{'learning_rate': 0.1, 'n_estimators': 200}	{'learning_rate': 0.05, 'n_estimators': 250, '...	{'n_neighbors': 10, 'weights': 'distance'}	{'kNN__kneighborsregressor__weights': 'distanc...	{'XGB__xgbregressor__n_estimators': 100, 'XGB_...
MSE	61.813245	60.529299	57.806156	68.777118	62.470221	61.421818
MAE	5.714213	5.77645	5.564431	5.943662	5.855617	5.798808

Why model performs bad in the submission part

- lacking of further discussion for the protein_sequence, **checked**
- lacking of more features (external dataset) conducted by the biologists and computer scientists, **checked**
- the model itself is poorly constructed, **checked**
- there’re some tricks beneath the test dataset. **exactly!**

Why model performs bad in the submission part

the test data contains only protein sequences with many small mutations in amino acid, so the number of individual amino acids changes very little

- only one pH value in test data
- all test data contain and from the same data source
- the length in test data is very similar
- the protein sequence in test dataset is very alike, or otherwise, only differs form mutation or deletion point

Pros and Cons

- pros
  - detailed data preprocessing procedure, feature engineering is interesting and somewhat inspiring
  - abundant models with comparison, with NN as well
- cons
  - models evaluation should be more specific
  - lots of further discussion should be on NN
  - lacking of deeper biological field knowledge which could inspire model construction
  - cannot get price, 1st Place - $ 12,000!

Bottomline

Thanks for your attention.

Q&A

Appendix - support material

1. Data source: https://www.kaggle.com/competitions/novozymes-enzyme-stability-prediction/data
2. For each part in specific, see in: https://github.com/Lecter314/MLDM_2022_ExamProject_YuTianxiong_EEP

Return to presentation