Enzyme Stability Prediction

Prakash Kumar

210107103

Submission Date: April 26, 2024

Final Project submission

Course Name : Applications of Al and ML in chemical engineering

Course Code: CL653

Contents

1. Executive Summary:

Overview: Develop computational model to predict thermostability of enzyme variants from sequence/structure data to enable broader industrial use of enzymes as green biocatalysts.

Approach: Multidisciplinary combining biochemistry, molecular biology, and computational science. Train ML models (SVM, Random Forest, XGBoost, deep learning) on melting temperature data.

Data: Novozymes Kaggle dataset with protein sequences, pH, data sources.

Methodology: Data preprocessing, EDA, model training/tuning, evaluation (Spearman correlation), cross-validation.

Deployment: Integrate with LIMS/production systems via APIs. User interfaces. Cloud scaling. GPU acceleration.

Applications: Biofuels, pharmaceuticals, food production - improve efficiency, reduce costs/environmental impact.

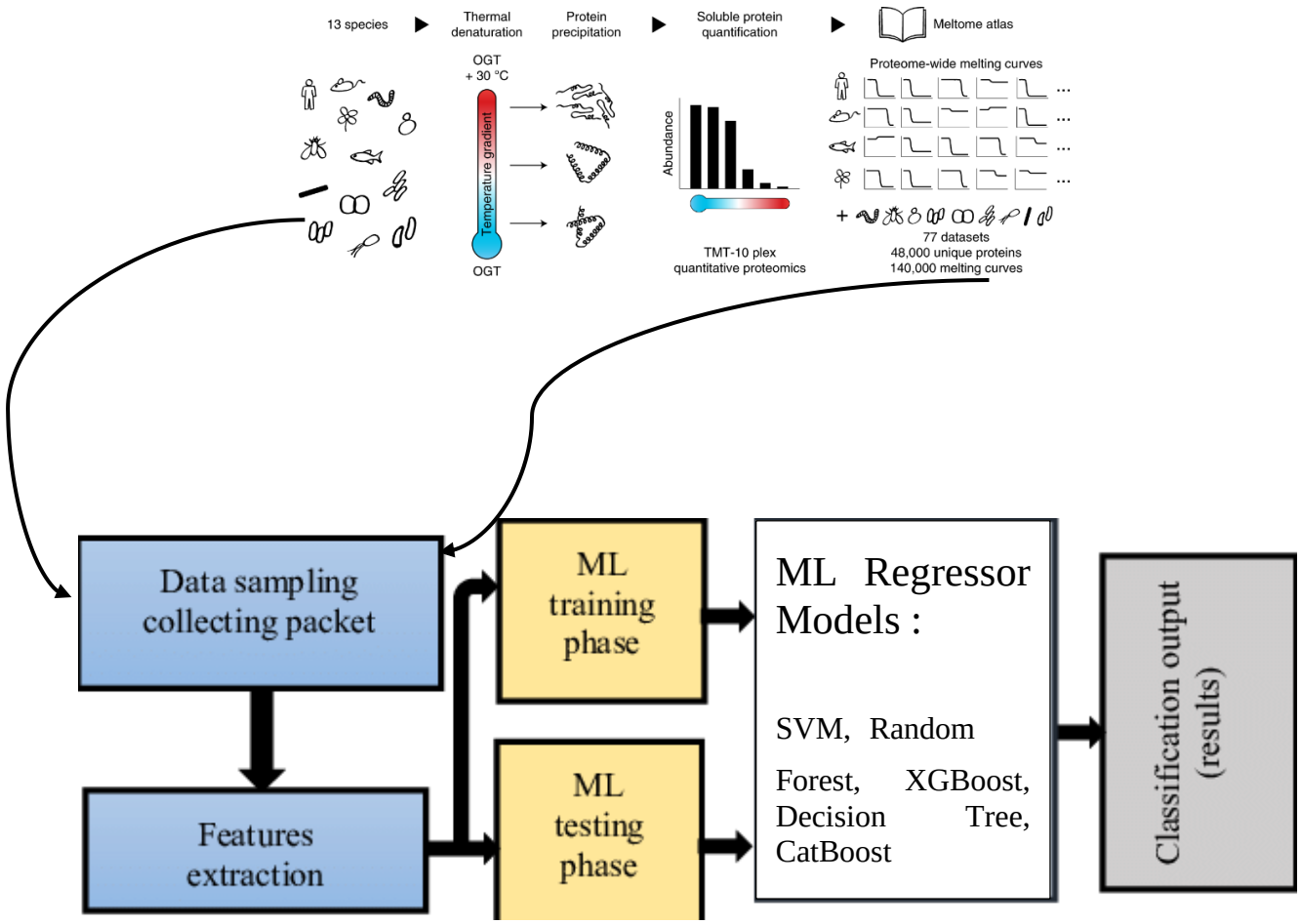
Impact: Enable sustainable industrial enzyme catalysis. Accelerate enzyme engineering innovation.

2. Introduction:

In the domain of chemical engineering, enzymes play a pivotal role as ***biocatalysts***, driving reactions that are essential for the production of a wide array of chemicals, materials, and pharmaceuticals. Their unique ability to operate under mild conditions—lower temperatures, neutral pH, and atmospheric pressure—contrasts sharply with traditional chemical catalysts that often require harsh environmental conditions. This distinction not only makes enzymes a cornerstone of ***green chemistry and sustainable industrial processes*** but also presents a compelling case for their integration into chemical engineering workflows. However, the application of enzymes in chemical engineering is ***frequently challenged by their inherent stability, particularly under the rigorous conditions*** of industrial processes. Thermostability, or the ability of enzymes to maintain functionality at elevated temperatures, is a critical attribute that can dictate the feasibility of enzymatic processes in chemical engineering.

3. Methodology:

a. Flow chart for data collection and model development



b. Data Preprocessing:

i. Data Source: Novozymes Enzyme Stability Prediction on Kaggle.

ii. Data Cleaning: Handling of missing data points, which may involve imputation techniques such as mean substitution or deletion of records with missing values.

iii. Feature Engineering: Explore potential interactions or combinations of features that may provide additional predictive power.

iv. Exploratory Data Analysis (EDA): Conduct exploratory data analysis to gain insights into the relationships between features and the target variable. Visualize distributions, correlations, and patterns in the data using techniques like histograms, scatter plots, or correlation matrices. Identify any potential trends or outliers that may impact model training and interpretation.

c. Model Development:

For the project of predicting enzyme thermostability, the AI/ML models under consideration include: ***Support Vector Regressor (SVR), Decision Trees, Random Forest***, ***Gradient Boosting Machine (XGBoost) and Catboost .***

The rationale for choosing these models lies in their ***versatility, interpretability***, and ability to handle both numerical and categorical features present in the dataset. Ensemble methods like Random Forest, XGBoost and Catboost is particularly well-suited for handling complex interactions between features.

d. Training:

i. Splitting data: Divide the dataset into training and testing sets, typically using a 70-30 or 80-20 split ratio.

ii. Model training: Fit the selected machine learning models (XGBoost, Random Forest, SVM) on the training data using appropriate libraries such as scikit-learn in Python.

iii. Hyperparameter tuning: Optimize model hyperparameters using techniques like grid search or random search to improve model performance.

iv. Cross-validation: Perform k-fold cross-validation on the training data to assess model generalization and stability.

e. Evaluation and Validation:

i. Evaluation Metrics: For evaluating the model use ***spearman correlation***. Spearman correlation: The Spearman correlation coefficient is calculated by first ranking each variable, then applying the Pearson correlation coefficient formula to these ranks. The Spearman correlation coefficient can range from -1 to +1. A coefficient of +1 indicates a perfect positive association of ranks, -1 indicates a perfect negative association of ranks, and 0 indicates no association between ranks.

4. Implementation:

```
[29] train_df.isna().sum()

seq_id                0
protein_sequence      0
pH                    0
data_source        3347
tm                    0
dtype: int64
```

```
from scipy import stats

def remove_outliers_zscore(column):
    z_scores = stats.zscore(column)
    outlier_indices = abs(z_scores) > threshold
    column[outlier_indices] = column.mean()
    return column
```

```
[23] threshold = 3
# Apply the function to all numeric columns in the DataFrame
train_df['tm'] = remove_outliers_zscore(train_df['tm'])
#check after outlier detection and removal some times while dropping or replacing nan values may be created.
#change to iqr code and check urself
train_df.isna().sum()
```

```
] temp = train_df['pH']
  temp = np.where(temp > 14, temp.mean, temp)
  train_df['pH'] = temp
```

```
[40] train_df["protein_length"] = train_df["protein_sequence"].apply(lambda x: len(x))
     (variable) train_df: DataFrame
train_df.head(3)
```

	seq_id	protein_sequence	pH	tm	protein_length
0	0	AAAAKAAALALLGEAPEVVDIWLPAGWRQPFRVFRLERKGDGVLVG...	7.0	75.7	341
1	1	AAADGEPLHNEEERAGAGQVGRSLPQESEEQRTGSRPRRRRDLGSR...	7.0	50.5	286
2	2	AAAFSTPRATSYRILSSAGSGSTRADAPQVRRLHTTRDLLAKDYYA...	7.0	40.5	497

Next steps: Generate code with train_df | View recommended plots

```
[42] def return_amino_acid_df(df):
      # Feature Engineering on Train Data
      amino_acids=['A', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'V', 'W', 'Y']
      for amino_acid in amino_acids:
        df[amino_acid]=df['protein_sequence'].str.count(amino_acid,re.I)/df['protein_length']
        #df[amino_acid]=df['protein_sequence'].str.count(amino_acid,re.I)
      return df
```

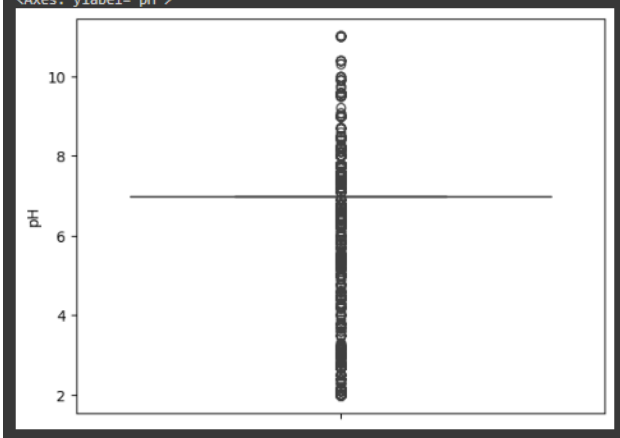
```
[43] train_df = return_amino_acid_df(train_df)
```

```
[45] #Aromaticity
     def calculate_aromaticity(row):
       sequence = str(row[1])
       X = ProteinAnalysis(sequence)
       return "%0.2f" % X.aromaticity()

     #Molecular Weight
     def calculate_molecular_weight(row):
       sequence = str(row[1])
       X = ProteinAnalysis(sequence)
       return "%0.2f" % X.molecular_weight()

     #Instability Index
     def calculate_instability_index(row):
       sequence = str(row[1])
       X = ProteinAnalysis(sequence)
       return "%0.2f" % X.instability_index()

     #Hydrophobicity
     def calculate_hydrophobicity(row):
       sequence = str(row[1])
       X = ProteinAnalysis(sequence)
       return "%0.2f" % X.gravy(scale='KyteDoolitle')

     #Isoelectric Point
     def calculate_isoelectric_point(row):
       sequence = str(row[1])
       X = ProteinAnalysis(sequence)
       return "%0.2f" % X.isoelectric_point()

     #Charge
     def calculate_charge(row):
       sequence = str(row[1])
       X = ProteinAnalysis(sequence)
       return "%0.2f" % X.charge_at_pH(row[2])
```

```
[46] train_df['Aromaticity'] = train_df.apply(calculate_aromaticity, axis=1)
     train_df['Molecular Weight'] = train_df.apply(calculate_molecular_weight, axis=1)
     train_df['Instability Index'] = train_df.apply(calculate_instability_index, axis=1)
     train_df['Hydrophobicity'] = train_df.apply(calculate_hydrophobicity, axis=1)
     train_df['Isoelectric Point'] = train_df.apply(calculate_isoelectric_point, axis=1)
```

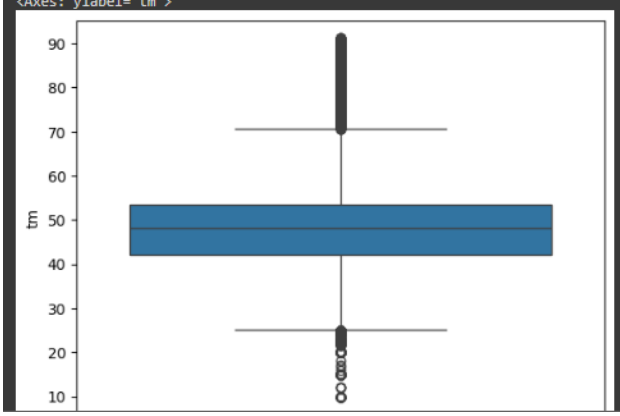
```
sns.boxplot(train_df['pH'])
```

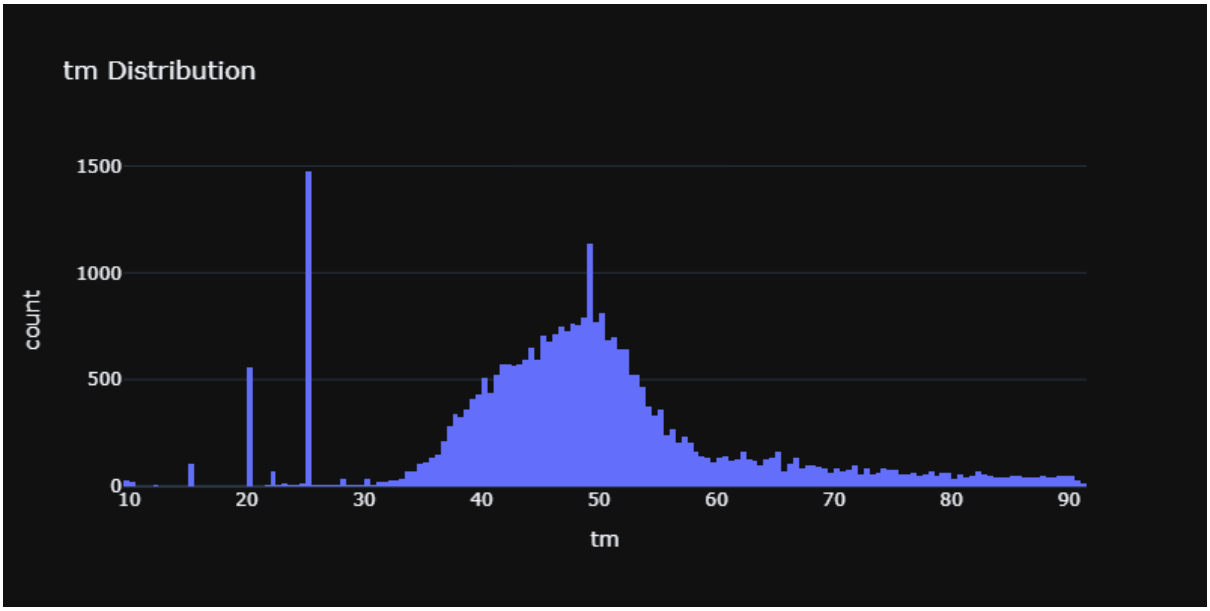
```
<Axes: ylabel='pH'>
```

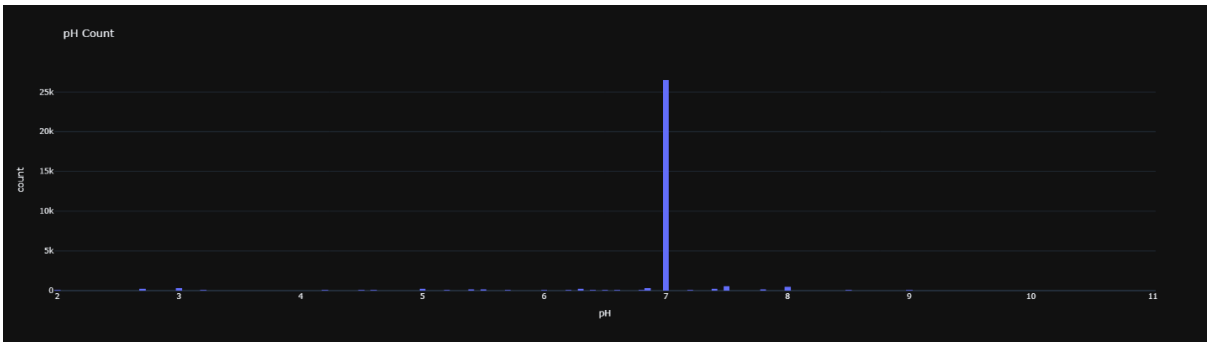


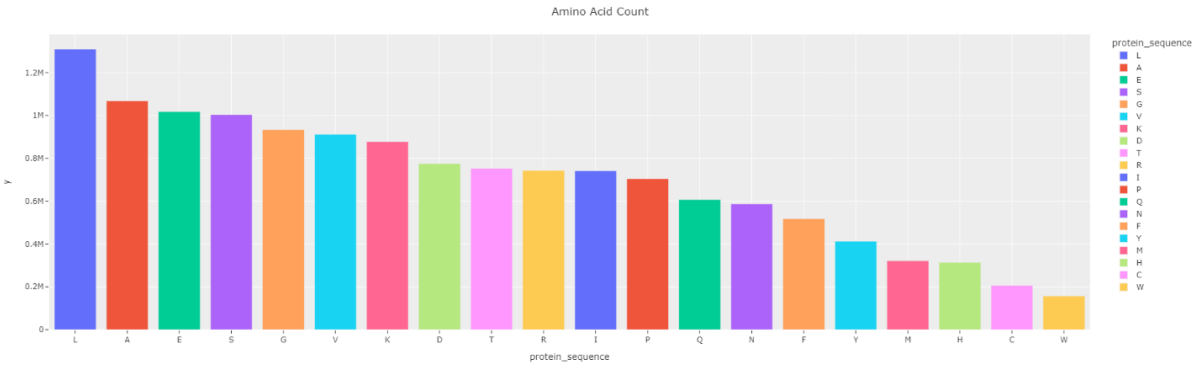

```
sns.boxplot(train_df['tm'])
```

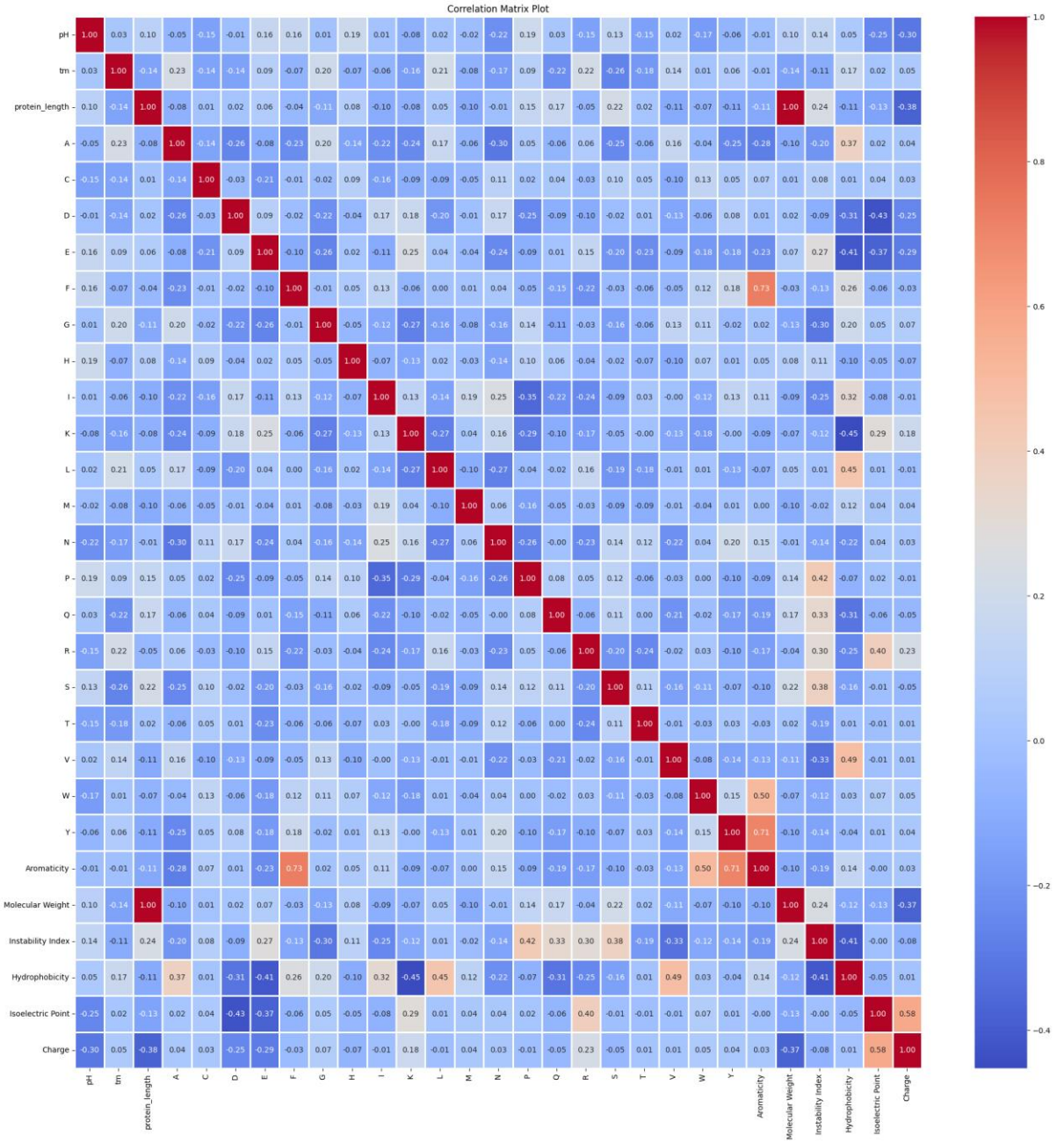
```
<Axes: ylabel='tm'>
```




tm Distribution
1500
1000
500
0
count
10
20
30
40
50
60
70
80
90
tm


pH Count
25k
20k
15k
10k
5k
0
count
2
3
4
5
6
7
8
9
10
11
pH


Amino Acid Count
1.2M
1M
0.8M
0.6M
0.4M
0.2M
0
y
L
A
E
S
G
V
K
D
T
R
I
P
Q
N
F
Y
M
H
C
W
protein_sequence
protein_sequence


Correlation Matrix Plot
pH
tm
protein_length
A
C
D
E
F
G
H
I
K
L
M
N
P
Q
R
S
T
V
W
Y
Aromaticity
Molecular Weight
Instability Index
Hydrophobicity
Isoelectric Point
Charge
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

5. Testing And Deployment:

a. Testing Strategy: Testing was done with train-test split.

b. Deployment Strategy:

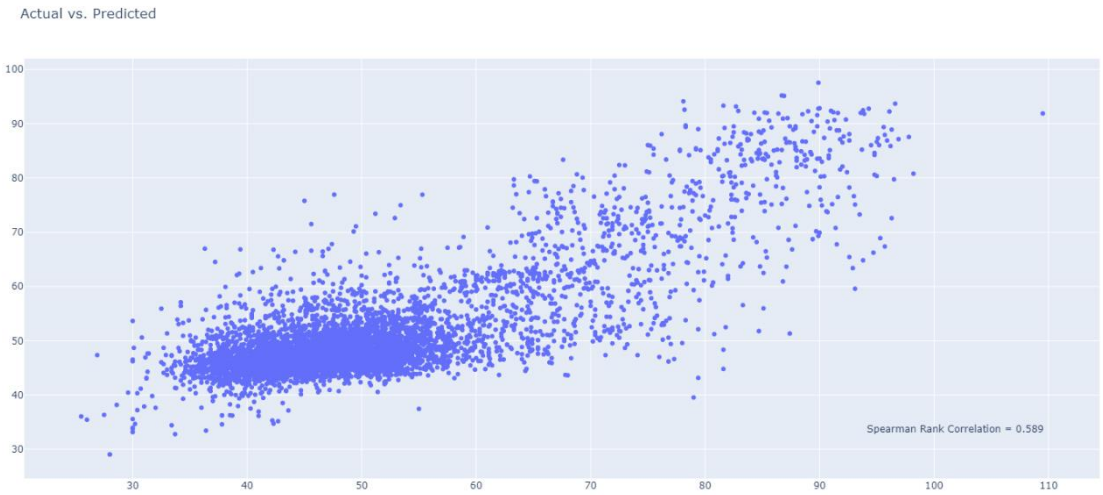
i. Integration with systems: Enable easy access to the model's predictions through APIs or web services integrated with laboratory information management systems (LIMS) or production control system.

ii. User Interface: Design a user-friendly interface that allows scientists, engineers, and technicians to interact with the model. Incorporate visualization tools within the interface to help users interpret the model's predictions, such as graphs showing enzyme stability under different conditions. Provide training and documentation to ensure that users can effectively utilize the model and interface.

iii. Maintenance and Updates: Set up a system for continuous learning, where the model can be updated with new data from ongoing enzyme stability experiments and production processes.

iv. Security: Ensure compliance with data privacy regulations and industry standards for data protection.

6. Result and Discussion:



```
998:    learn: 5.5969028        total: 26.8s    remaining: 26.9ms
999:    learn: 5.5947248        total: 26.9s    remaining: 0us
Best parameters: {'depth': 8, 'l2_leaf_reg': 1}
SignificanceResult(statistic=0.5936646072111422, pvalue=0.0)
```

Got a spearman coeff. of 0.54.

Challenges and Limitations:

- Data Quality: Ensuring the quality and representativeness of the dataset posed challenges, particularly in obtaining comprehensive data covering a wide range of operating conditions and scenarios.
- Feature Engineering: Extracting informative features from raw data and determining their relevance to CHF prediction required careful consideration and domain expertise.
- Model Complexity: Balancing model complexity and interpretability was a challenge, as more complex models tended to provide better predictive performance but were harder to interpret and explain.

7. Conclusion and Future Work

The enzyme stability prediction project represents a significant innovation in the field of biotechnology and chemical engineering. By leveraging advanced AI/ML models, the project aims to accurately predict the thermostability of enzyme variants, which is a critical factor in their industrial application. The key points of the project can be summarized as follows:

- Addressing a Critical Need: The project focuses on solving the challenge of enzyme instability under industrial conditions, particularly at high temperatures, which is a major bottleneck in the widespread use of enzymes in various industries.
- Advanced Computational Models: A range of AI/ML models, including Support Vector Machines, Random Forest, XGBoost, and deep learning architectures, have been considered for their ability to handle the complex patterns in enzyme sequence and structure data.
- Real-World Application: The predictive model has practical applications in industries such as biofuels, pharmaceuticals, and food processing, where it can guide the selection and engineering of enzymes for improved stability and efficiency.
- Environmental and Economic Impact: By enabling the use of more stable enzymes, the project contributes to more sustainable industrial processes, reducing energy consumption and waste, and leading to cost savings.
- Accelerating Innovation: The ability to predict enzyme stability rapidly and accurately can significantly speed up the enzyme engineering process, facilitating the development of new and improved biocatalysts.

In conclusion, the enzyme stability prediction project stands out for its potential to transform industrial biocatalysis. It combines cutting-edge computational techniques with practical applications, offering a pathway to more sustainable and efficient manufacturing processes. The project's success could have a lasting impact on multiple sectors, driving innovation and sustainability in equal measure.

8. References

- Anna Jarzab, Nils Kurzawa, Thomas Hopf, Matthias Moerch, Jana Zecha, Niels Leijten, Yangyang Bian, Eva Musiol, Melanie Maschberger, Gabriele Stoehr, Isabelle Becher, Charlotte Daly, Patroklos Samaras, Julia Mergner, Britta Spanier, Angel Angelov, Thilo Werner, Marcus Bantscheff, Mathias Wilhelm, Martin Klingenspor, Simone Lemeer, Wolfgang Liebl, Hannes Hahne, Mikhail M. Savitski & Bernhard Kuster,*" Meltome atlas—thermal proteome stability across the tree of life"*, Nature Methods (13 April 2020)

9. Auxiliaries

Data taken from Kaggle.

10. Appendices

DataSource:https://www.kaggle.com/c/novozymes-enzyme-stability-prediction

Python file:

https://colab.research.google.com/drive/1fpM8VCaUQUv0yXTyMAQ5Kx5i5DLs4Mnm?usp=sharing