CMPE-257-Group-11 Project Proposal

Public Repo Link: https://github.com/akashmat/CMPE-257-Group-11

Group Member Name (Github Username)

- **Akash Mattaparthy** (akashmat)
- **Dodda Sai Venkata Phanith** (phanithdsv)
- **Himaja Narina** (himajanarinaa)
- **Shravani Naikoti** (shravaninaikoti)

1. Novozymes Enzyme Stability Prediction

This is an ongoing competetion on Kaggle

2. Dataset

Dataset taken from the Competetion site here

3. Problem Description

The competetion involves the prediction of thermostability of enzyme variants (hence using regression methods). The experimentally measured thermostability (melting temperature) data includes natural sequences, as well as engineered sequences with single or multiple mutations upon the natural sequences.

Biotechnology has a basic challenge in trying to understand and reliably predict protein stability. Enzyme engineering can be used to address issues like sustainability, carbon neutrality, and other global challenges. Enzyme stability improvements may save expenses and speed up concept iteration for scientists.

4. Potential Methods

1. This regression problem can be solved with ensemble methods like Random Forest, XGBoost.
2. An attempt to solve it using transformer models like BeRT will also be used.

5. Create GitHub repository with a file called README.md

Public Repo Link: https://github.com/akashmat/CMPE-257-Group-11

6. Preprocessing & Initial Findings

Do some intial investigation of the data (basic statistics and plots). What do you see? Any patterns? What challenges do you forsee.

In [18]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [11]:
```
train_df = pd.read_csv('./train.csv')
test_df = pd.read_csv('./test.csv')
```

In [12]:
```
print(f"Train dataset Shape: {train_df.shape}")
print(f"Test dataset Shape: {test_df.shape}")
```

```
Train dataset Shape: (31390, 5)
Test dataset Shape: (2413, 4)
```

In [13]:
```
train_df.head()
```

Out[13]:

	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

The columns in the training dataset:

1. seq_id: unique identifier of each protein variants
2. protein_sequence: amino acid sequence of each protein variant. The stability of a protein is determined by this protein sequence.
3. pH: measures the acidity of an aqueous solution in which the stability of protein was measured. This is important to as the stability of the same protein varies with different pH levels.
4. data_source: source where the data was published
5. tm: target column to measure the stability of protien. The higher the tm, the more statble.

In [17]:
```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31390 entries, 0 to 31389
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   seq_id            31390 non-null  int64  
 1   protein_sequence  31390 non-null  object 
 2   pH                31104 non-null  float64
 3   data_source       28043 non-null  object 
 4   tm                31390 non-null  float64
dtypes: float64(2), int64(1), object(2)
memory usage: 1.2+ MB
```

In [19]:
```
import pandas_profiling
```

In [20]:
```
pandas_profiling.ProfileReport(train_df)
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Pandas Profiling Report — Overview | Variables | Interactions | Correlations | Missing values | Sample

Missing cells	3633
Missing cells (%)	2.3%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	1.2 MiB
Average record size in memory	40.0 B

Variables

Select Columns

seq_id
Real number ($\mathbb{R}_{\geq 0}$)

UNIFORM
UNIQUE

Distinct	31390
Distinct (%)	100.0%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	15694.5

Minimum	0
Maximum	31389
Zeros	1
Zeros (%)	< 0.1%
Negative	0
Negative (%)	0.0%
Memory size	245.4 KiB

Toggle details

protein_sequence
Categorical

Distinct	MNIFEMLRIDERLRLKIYKDTEG…	14
Distinct	MRNTGAGPSPSVSRPPPSAAP…	13

Out[20]:

From the report, the following can be noted:

- variables protein_sequence and data_source have high cardinality, i.e., have a lot of unique values. The protein_sequence obviously unique (92.3%) due to nature of problem and variety.
- data_source has some missing values; since it has no value in the dataset, it can be deleted (also being the only variable with missing values)
- seq_id and protein_sequence are uniformly distributed

What are some of the preprocessing techniques did you use?

- Only missing column was data_source which can be simply not selected by index selection

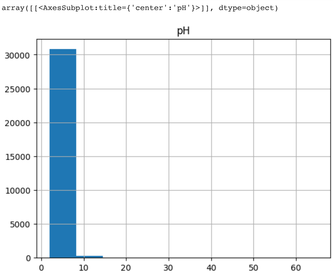
In [21]:
```
train_df = train_df[["seq_id", "protein_sequence", "pH", "tm"]]
train_df.head()
```

Out[21]:

	seq_id	protein_sequence	pH	tm
0	0	AAAAKAAALALLGEAPEVVDIWLPAGWRQPFRVFRLERKGDGVLVG...	7.0	75.7
1	1	AAADGEPLHNEEERAGAGQVGRSLPQESEEQRTGSRPRRRRDLGSR...	7.0	50.5
2	2	AAAFSTPRATSYRILSSAGSGSTRADAPQVRRLHTTRDLLAKDYYA...	7.0	40.5
3	3	AAASGLRTAIPAQPLRHLLQPAPRPCLRPFGLLSVRAGSARRSGLL...	7.0	47.2
4	4	AAATKSGPRRQSQGASVRTFTPFYFLVEPVDTLSVRGSSVILNCSA...	7.0	49.5

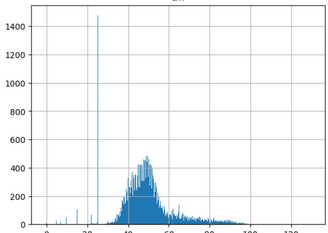
In [27]:
```
train_df.hist("pH")
```

Out[27]:
```
array([[<AxesSubplot:title={'center':'pH'}>]], dtype=object)
```



In [25]:
```
train_df.hist("tm", bins=500)
```

Out[25]:
```
array([[<AxesSubplot:title={'center':'tm'}>]], dtype=object)
```



Both pH and tm are skewed; in particular tm is skewed towards the left, showing most protiens have melting points at higher tm.

Challenges:

1. The high uniformity of protein_sequences can be a problem.
2. The skewed data set in terms of pH and tm can also lead to a model which is un-representative of the general data. Other datasets can be added to compensate for this.

In [ ]:
```
```