Chapter 1: Exploration of Dimensionality Reduction Techniques

Author: Venkatesh Manikantan R0825919

Feature expansion of amino acids sequences results in the extraction of 8,974 features comprising of amnio acids frequencies, molecular parameters and amino acid autocorrelating descriptors. The ratio between the number of observations to the number of features is approximately 1:4, and therefore, the data set classifies to be a high dimensional data.

High dimensional data generally encompasses a certain degree of noise, which could impact the performance of a ML model and easily lead to overfitting, poor performance if not corrected using regularization parameters in the model.

Dimensionality reduction techniques such as PCA, K-PCA and denoising autoencoders compress the data into lower dimensions. The impact of these techniques will be studied with respect to the performance they provide in predicting enzyme catalytic optimum temperature T_{OPT}. [1] Furthermore, techniques such as these also reveal a decision boundary between distinct values, the nature of such decision boundaries are vital in determining the type of model that would yield the optimum predictive performance.

1.1 Principal Component Analysis:

PCA a linear dimensionality reduction technique is used to understand the eigenvalue decomposition with respect to feature loadings and also act as a lower dimension input data for our model.[1][1] PCA is a tool used in analyzing enzymes based amino acid frequencies, in this field it is often used to extract strong patterns and declutter the data from noise. [2][2] PCA has been widely been used and has been proven successful in extracting inference in multiple domain such as physics, biostatistics and social science.

The methodology that PCA is used in our application is to provide featured reduced dataset which does not contain colinear independent features for our models.

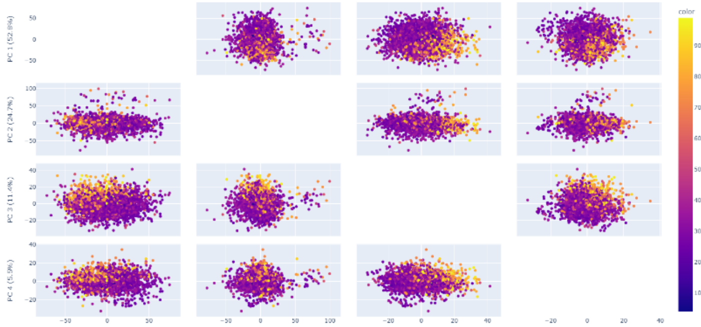


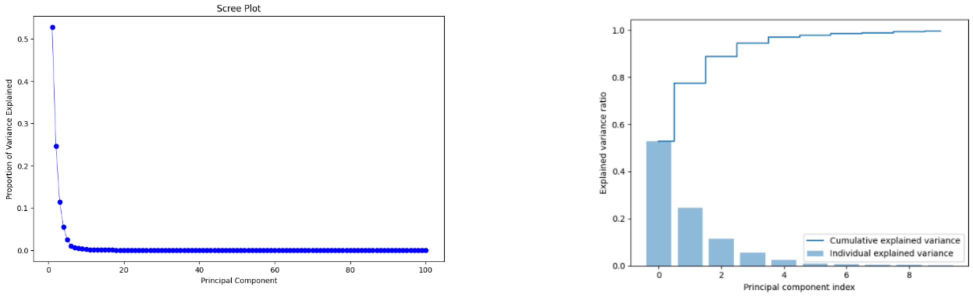
Figure 1: Grid Plot of the first four principal components explaining a significant variance

From the figure set above it can be observed that higher T_{OPT} values are separable from the lower temperatures, especially temperatures above 80˚C are further more separated from the main cluster. This

[1] L. H. Brito, A. L. C. V. Lara, L. E. Zárate and C. N. Nobre, "Improving the quality of enzyme prediction by using feature selection and dimensionality reduction," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8851882.

[2] Lee-Wei Yang, Eran Eyal, Ivet Bahar, Akio Kitao, Principal component analysis of native ensembles of biomolecular structures (PCA_NEST): insights into functional dynamics, Bioinformatics, Volume 25, Issue 5, March 2009, Pages 606–614, https://doi.org/10.1093/bioinformatics/btp023

trend can be observed the most in the third principal component, where the higher temperature values are further separated from the rest.



(a)

(b)

Figure 2: (a) PCA scree plot showcasing the proportional variance explained per component, (b) Cumulative variance explained by components

The first four principal components explain 94.4% variance in our data set which comprises of 8,974 independent features. This indicates that majority of the features present in our data set were colinear in nature. The results of PCA are intriguing as it significantly reduces the complexity of the models employed to estimate T_{OPT}.

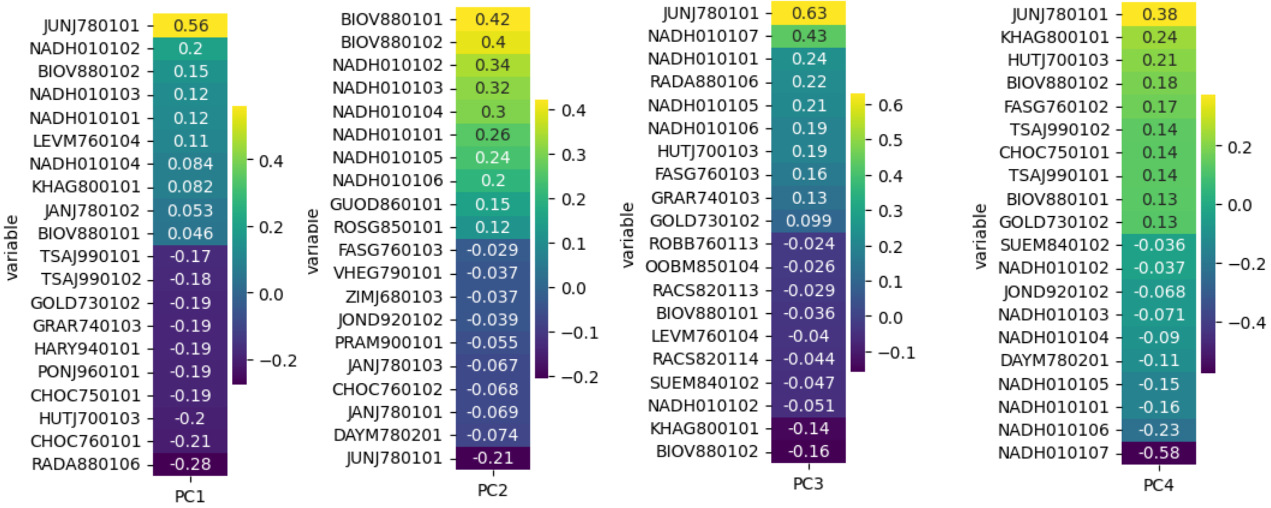


Figure 3: PCA feature loading value showcasing the top ten most weighted features with respect to first four components

The loadings showcase the features that are the most prominent in terms of compressing the initial data structure. Figure 3 showcases variables have significant contribution to the loading weight of their respective component. From the loading it can be interpreted that feature JUNJ780101, is an influential feature as it is represented in the top four principal components.

The results from PCA reduced data is used to train models to predict values for T_{OPT}. The inference gathered from the fit of the model and the list of influential features w.r.t. PCA are important to fine tuning and choosing an appropriate model.

Table 1: Model results from PCA dimensionality reduced data set and feature selection

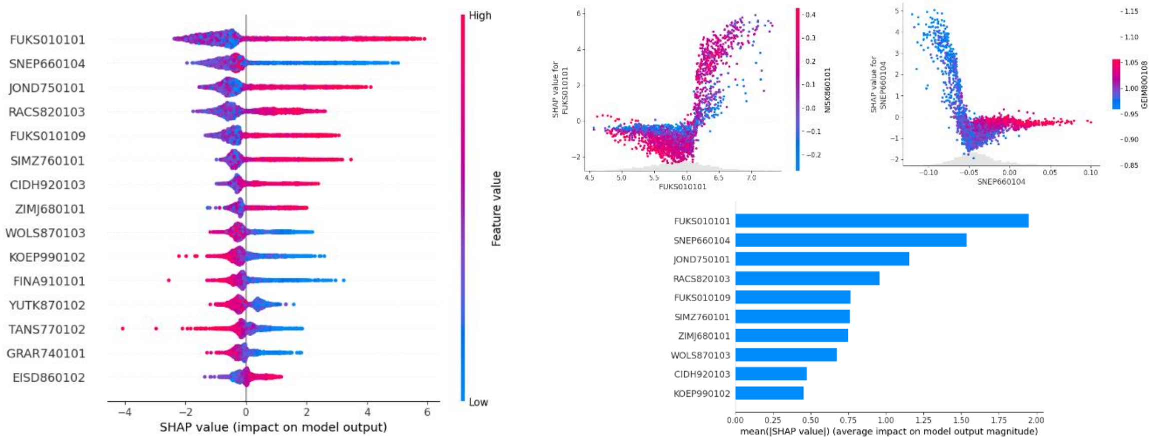
DIMENSIONALITY REDUCTION EXPLORATION					
REGRESSION: TEMPRATURE OPTIMA					
Model:	10-K Cross Validation (RMSE)				
	3-D PCA	10 D PCA	100 D PCA	1000 D PCA	PCA Feature Selection
XGBOOST	17.16	14.2	12.68	13.26	11.45
XGBOOST FOREST	16.45	13.98	12.53	12.84	10.60
LIGHTGBM	16.77	14.42	13.12	13.54	12.20
CATBOOST	17.2	14.22	13.3	13.43	11.96

PCA reduced data set has not been effective in increasing the performance of the models. The model using the 3-dimensional compressed data set explaining 88.9 % of the variance of our input data has not been affective in training an effective model to estimate T_{OPT}.

Models trained using more principal components resulted in significantly increasing the performance of our models. Amongst all the models, XGBOOST forest has been successful in having the best performance, as it combines the advantage of boosting technique of XGBOOST and the stratified feature and data sampling methods of a random forest.

Interesting trend to notice is that changing the dimension from 100 to 1000 input reduced PCA data has not yielded improved results compared to the lower feature model. As the amount of explained variance by each component is lower than its previous iteration and thus results in including features that are non-descriptive of the original data.

Top 200 features with the highest absolute value of loadings in the first 4 principal components were taken as input feature data set for the models, duplicate independent features were removed from the list of 800 features extracted from the first four principal components, resulting in the final feature list of 325 independent features to estimate T_{OPT}.



(a)

(b)

Figure 4: (a) (PCA-Feature Selection Model) SHAPLY bee-swarm plot showcasing feature importance and interaction with Topt, (b) Top two feature scatter plots with their respective co-dependent feature, followed by the impact of those feature on the predictive performance.

PCA feature selection yielded the best results when employed with XGBOOST forest resulting in model with root mean squared value of 10.60 and R-squared value of 0.71. This further supported by the feature importance gathered from the performance of the model showcased in Figure 4.

Feature "FUKS010101" is the most influential feature to estimate T_{OPT}, from Figure 4 (a) it can be deduced that lower values of this feature leads to a lower T_{OPT} value, whereas higher value of the feature leads to higher T_{OPT} value.

The second most influential feature "SNEP660104", has the inverse relationship with T_{OPT} compared to "FUKS010101". Furthermore, the feature with highest loading weight "JUNJ780101" is not part of the top 50 important features to estimate T_{OPT}

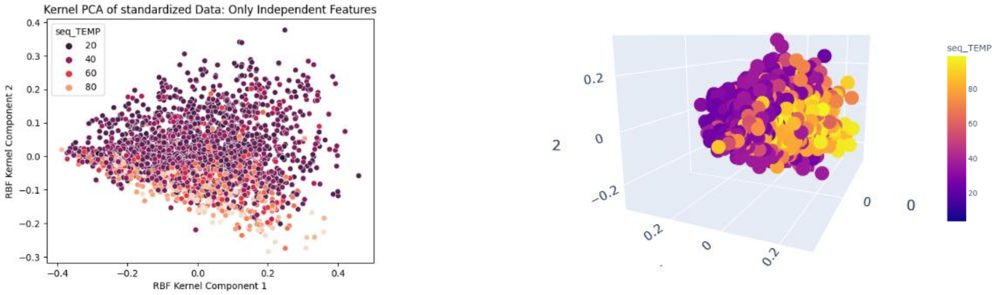
1.2 Kernel Principal Component Analysis:

K-PCA technique is a powerful dimensionality reduction technique used to on data set which represent the feature set in non-linear fashion. [3] [3]The dataset which do not confide to linear separation, can be separated by projecting the data into higher dimension using mapping functions. Kernel functions are the resulting dot product of two mapping functions of the sample dataset.

$$K(X_i, X_j) = \emptyset(X_i) \cdot \emptyset(X_J)^T$$

K-PCA employed in the exploration is the Gaussian-Radial Bias function implementation, which is a native function in the *sklearn's* decomposition library.

$$\mathrm{K}(X_i, X_j) = \exp(-\gamma \parallel X_i - X_j \parallel_2^2)$$



(a)

(b)

Figure 5: (a) K-PCA RBF (gamma:1/(number of features) showcasing 2 dimensions, (b) Identical KPCA parameters showcasing 3 dimensions

[4]From the figure above, a clear split between the lower and higher temperature can be observed in both the graphs. This is a stark difference from the results obtained in the process of PCA, where the higher and lower values were not so distinctively separated. [5]

[3] Jorgensen, Palle E. T., et al. "An Infinite Dimensional Analysis of Kernel Principal Components." *arXiv.Org*, 8 Sept. 2022, arxiv.org/abs/1906.06451.

[4] Antoniou D, Schwartz SD. Toward Identification of the reaction coordinate directly from the transition state ensemble using the kernel PCA method. J Phys Chem B. 2011 Mar 17;115(10):2465-9. doi: 10.1021/jp111682x. Epub 2011 Feb 21. PMID: 21332236; PMCID: PMC3058940.

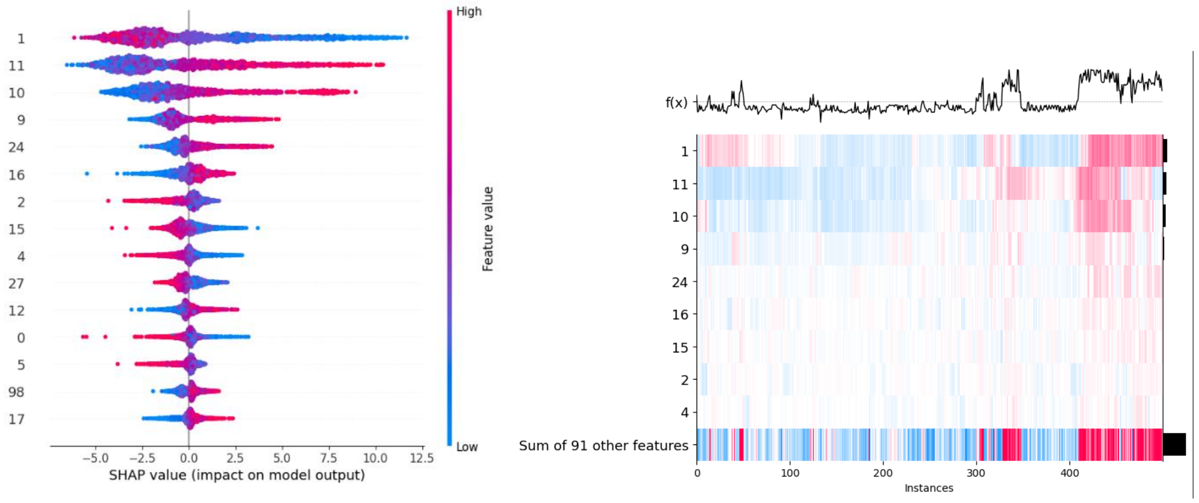
Table 2: Model results from K-PCA dimensionality reduced data set

DIMENSIONALITY REDUCTION EXPLORATION: K-PCA (RBF)				
REGRESSION: TEMPRATURE OPTIMA				
Model:	10-K Cross Validation (RMSE)			
	3-D K-PCA	10 D K-PCA	100 D K-PCA	1000 D K-PCA
XGBOOST	15.76	12.73	11.24	15.45
XGBOOST FOREST	15.50	12.31	10.55	15.05
LIGHTGBM	15.87	12.91	11.65	15.72
CATBOOST	16.20	12.95	11.72	16.13

RBF kernel PCA reduction has produced better results in comparison to PCA. The reduced dimensional space is yielding better results in all dimension reduced feature except the 1000 feature reduced PCA has better performance compared to K-PCA

Model trained using 100 D K-PCA, yielded marginally better performance in comparison to the PCA featured selected model, R-squared value of the new model is approximately 0.72.

The dataset used to train these models does not include a key feature which was the most influential feature in the [6] *(Li et al., 2019)*, that is organism growth temperature (T_{OGT}). Boosted forest with dimensionality reduction methods have proven to be reliable method to estimate T_{OPT}, using only the features extracted from the amino acid sequences.



(a)

(b)

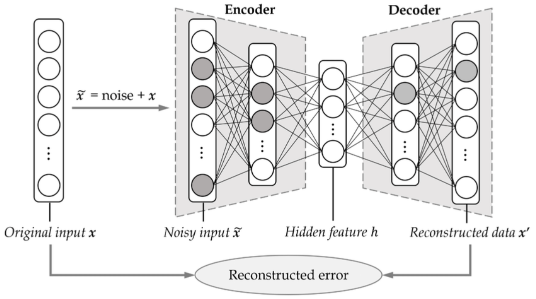
Figure 6: (a) 100 D K-PCA RBF SHAPLY Feature Importance with feature interaction with T_{OPT} (b) Shapley Heatmap plot showcasing hierarchical clustering of samples with respect to their explanation similarity.

[5] Rensi SE, Altman RB. Shallow Representation Learning via Kernel PCA Improves QSAR Modelability. J Chem Inf Model. 2017 Aug 28;57(8):1859-1867. doi: 10.1021/acs.jcim.6b00694. Epub 2017 Aug 7. PMID: 28727421; PMCID: PMC5942586.

[6] Li G, Rabe KS, Nielsen J, Engqvist MKM. Machine Learning Applied to Predicting Microorganism Growth Temperatures and Enzyme Catalytic Optima. ACS Synth Biol. 2019 Jun 21;8(6):1411-1420. doi: 10.1021/acssynbio.9b00099. Epub 2019 Jun 7. PMID: 31117361

From the feature reduced K-PCA structure, the first component of the K-PCA has the most contribution to the predictive performance in estimating T_{OPT}. The first component of the reduced K-PCA has an inverse relationship to models' predictive performance to estimate T_{OPT}, and moreover the effectiveness of K-PCA employing RBF kernel is validated as the first component of the new reduced representation of our data is consistently the most important feature used by the XGBOOST forest model.

1.3 Denoising Autoencoder Dimensionality Reduction:


Encoder
Decoder
$\tilde{x}$ = noise + x
Original input $\boldsymbol{x}$
Noisy input $\tilde{x}$
Hidden feature h
Reconstructed data $\boldsymbol{x}'$
Reconstructed error

Chapter 2: Modeling Strategies/Tuning and Performance Evaluation to Estimate T_{OPT}

Modelling strategies to formulate the best fit models, requires several pre-processing methods followed by rigorous fine tuning of hyperparameters used by the models to regularize its learning process. In this chapter, we will dive into the depths of the models working, training and performance in detail. From *Li et al., 2019,* the best performing model was random forest with R squared of 0.51 employing 5 cross validations. As large decision tree-based models yielded the best results, our exploration will be aimed towards using process intensive and expensive decision tree based boosted trees and forests. The boosting algorithm employed in this chapter are as follows: XGBOOST, XGBOOST Forest, Catboost, LightGBM. boosting models unlike random forest repeatedly initialize to rebalance the weights every iteration to minimize the objective function.

Having a deep insight of the model's hyperparameters is important to understand the steps taken to tune the model for the best performance while implementing regularization to curb over-fitting.

XGBOOST:

- N-Estimators: Number of boosting iterations, Early stopping is used to curb overfitting, large number of boosting iterations leads to overfitting
- Max-Depth: Determines the maximum Depth a tree can be in a boosting Iteration. Higher value of Max-Depth leads to overfitting.
- Learning Rate: Determines the rate at which a model adapts to minimize the objective function. Lower learning rate produces better fit models
- Gamma: Regularization parameter
- Subsample: Number of samples of data used by each iteration/tree in the boosting process
- Sampling Method: Uniform random selection is used to study the impact of sampling techniques
- Lambda: L2 regularization
- Alpha: L1 regularization
- Max Bin: The parameter is used to set bin size and tuning this parameter has impact in regression

XGBOOST Forest:

- Identical features to XGBOOST:
- Number of Parallel Trees: To train parallel trees similar to random forest

CATBOOST:

- Identical features to XGBOOST
- Random Strength
- l2_leaf_reg

LightGBM:

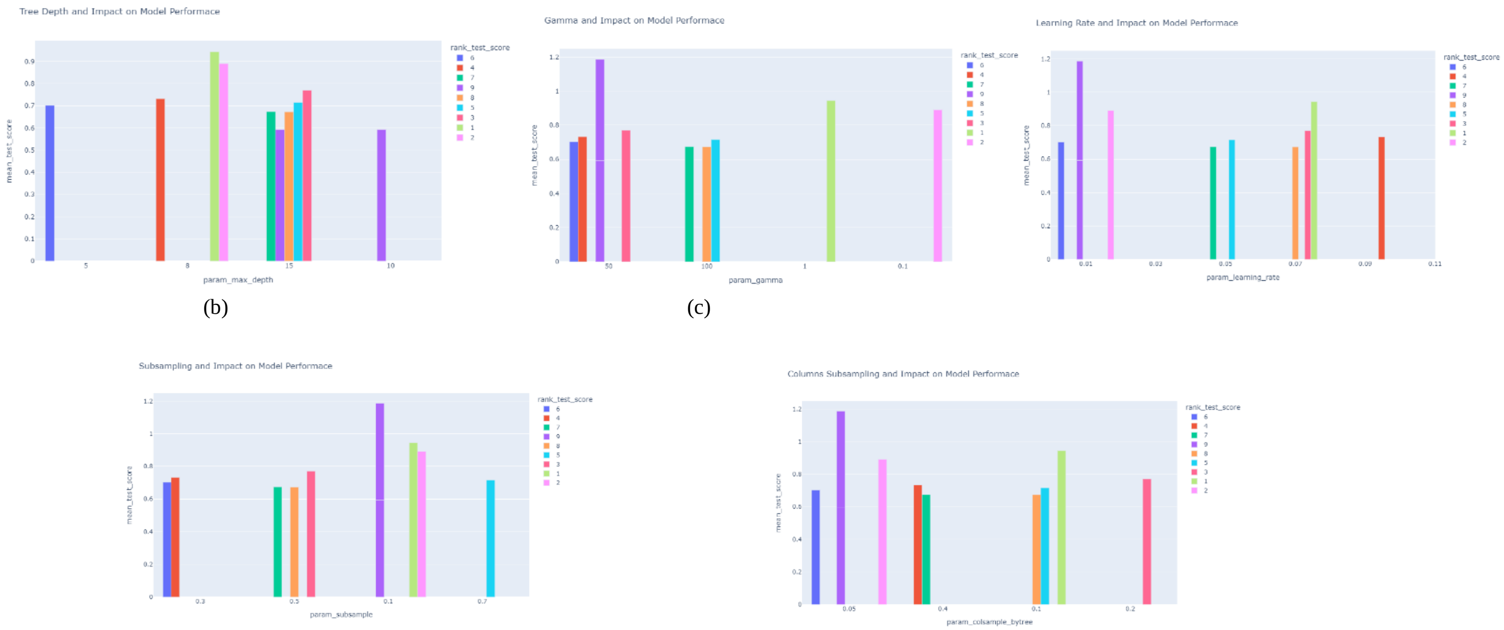
- Data Sample Strategy
- Force Column Wise
- Identical features to XGBOOST

2.1: Boosted Decision Tree Classification Models to Estimate Varied Bin classes of T_{OPT}

Multiclass classification models are employed to estimate bin ranges of T_{OPT}, the purpose of the exploration is to provide further inference with respect to feature selection, test several preprocessing techniques and implement hyperparameter tuning methods to train the best fit model.

The values of T_{OPT} range from 0˚C to 100˚C, classifier model is trained to identify between five ranges of classes equally binning T_{OPT} by 20˚C. The binning of T_{OPT} to five classes causes a heavy class imbalance, and therefore techniques of sampling such as over, under and over-under sampling methods are used to train the models.

Furthermore, hyperparameter tunning is vital step to the process of building the best fit model and this process is performed for all the models showcased in this chapter. Therefore, the process of hyperparameter tunning is further elaborated.



(b)

(c)

(d)

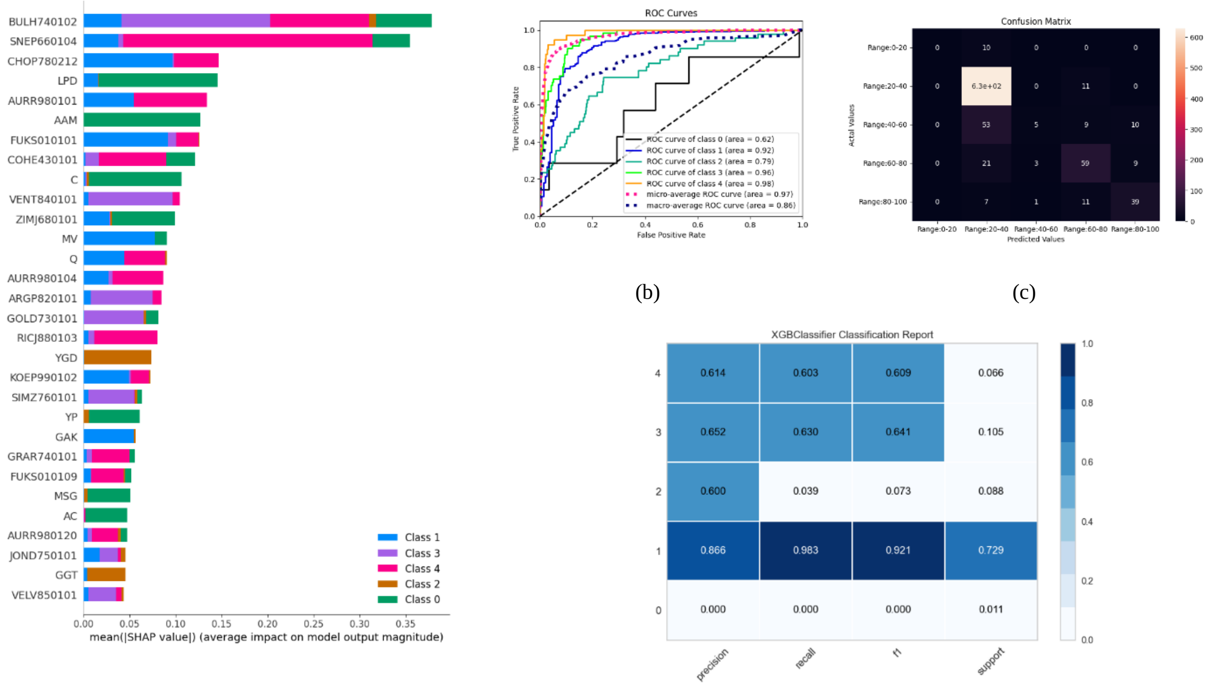
(e)

Figure 6: (a) 3 K-Fold cross validation AUC score of models with varied tree depth size (b) 3 K-Fold cross validation AUC score of models with varied gamma regularization's value (c) 3 K-Fold cross validation AUC score of models with learning rates (d) 3 K-Fold cross validation AUC score of models with varied sampling size taken by an individual tree in the iteration (e) 3 K-Fold cross validation AUC score of models with varied sampling percentage of the number of features per iteration. (*Green index showcases the best performing model- Rank 1,* Rank 9 (Purple) – two iterations share the same rank and hence are stacked on top of each other.)

Random search cross validation is a method often used to search the hyperparameter space for the best performing model. The results showcased above are from training a XGBOOST classifier model. Furthermore, it can be observed that increasing the depth of the boosted decision tree models does not necessarily yield the best results, rather when maximum tree depth is set to 8 has achieved the best result of 0.92 AUC. Similarly higher gamma regularization values have also underperformed compared to default value of 1 which yielded the highest value of 0.9 AUC. As expected, a learning rate lower than 0.1 have yielded better results and similarly lower values bagging proportion and lower percentage of random selection of features per iteration has also yielded positive results.

Hyperparameter tuning process resulted in the following value for the parameters to be the best performing over 300 model fits.

- Max Depth: 8
- Gamma: 1
- Learning Rate:0.7
- Subsample Ratio: 10 %
- Random Feature Selection Ratio: 10%



(b) (c)

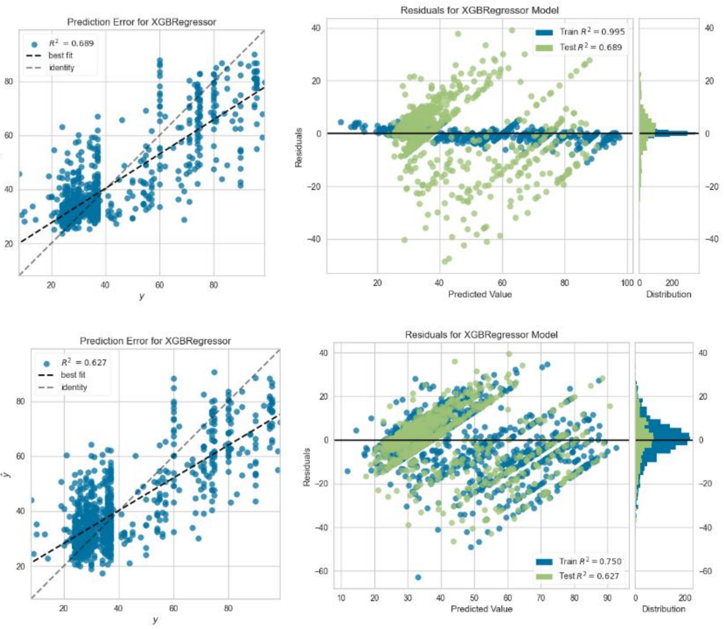
(a) (d)

Figure 7: (a) Classification model's SHAPLY Feature Importance (b) ROC-AUC curves showcasing the best model performance metrics (c) Confusion Matrix (d) Classification Models Report showcasing precision, recall, F1 and support.

Table 3: Model results from K-PCA dimensionality reduced data set

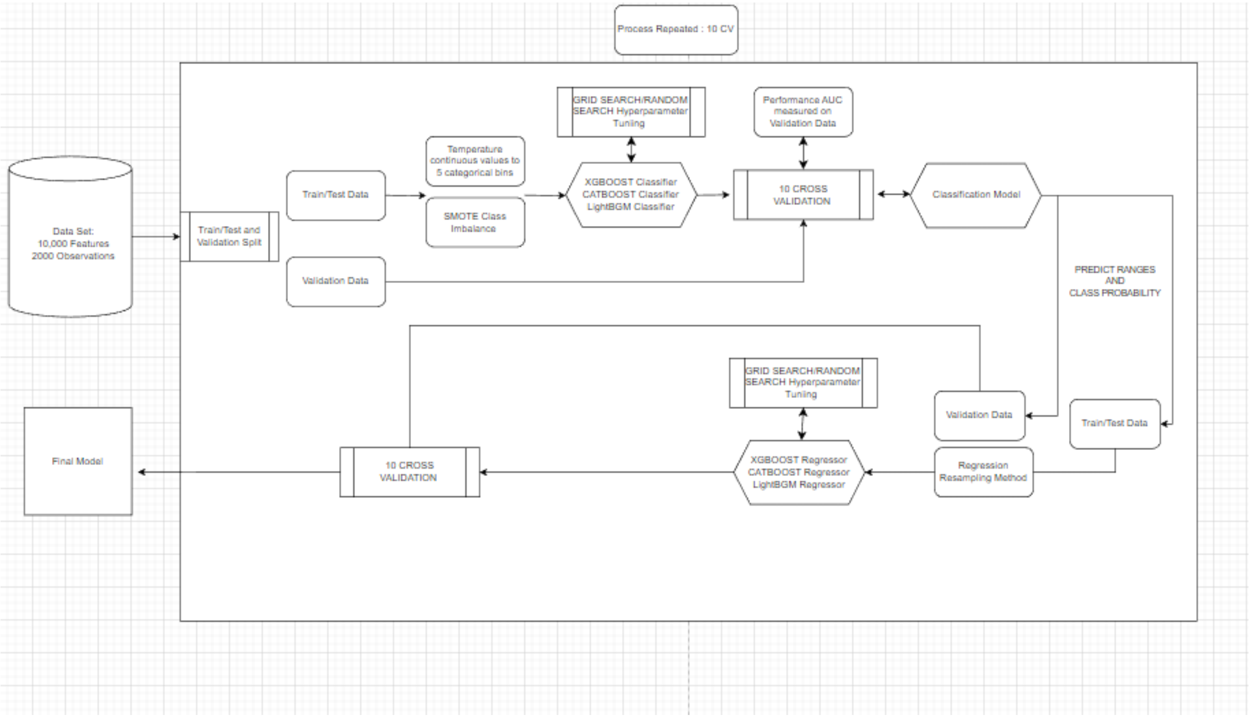
CLASSIFICATION MODEL: PREDICT T_{opt} Bins				
Bins: Range 1 (0-20), Range 2 (20-40), Range 3(40- 60), Range (60-80), Range(80-100)				
Model:	10-K Cross Validation			
	Micro -AUC	Macro-AUC	M-Error	M-LogLoss
XGBOOST	0.91	0.82	0.21	0.68
XGBOOST FOREST	0.96	0.86	0.18	0.65
LIGHTGBM	0.90	0.81	0.23	0.75
CATBOOST	0.91	0.82	0.21	0.67

2.2: Boosted Forest Regression Models to Estimate T_{OPT}



CLASSIFICATION MODEL: PREDICT T_{opt} Bins				
Bins: Range 1 (0-20), Range 2 (20-40), Range 3(40- 60), Range (60-80), Range(80-100)				
Model:	10-K Cross Validation			
	Micro -AUC	Macro-AUC	M-Error	M-LogLoss
XGBOOST	0.	12.73	11.24	15.45
XGBOOST FOREST	0.96	0.86	0.18	0.65
LIGHTGBM	15.87	12.91	11.65	15.72
CATBOOST	16.20	12.95	11.72	16.13

2.3: Combining Classification and Regression Model:


Process Repeated : 10 CV
GRID SEARCH/RANDOM SEARCH Hyperparameter Tuning
Performance AUC measured on Validation Data
Temperature continuous values to 5 categorical bins
Train/Test Data
XGBOOST Classifier CATBOOST Classifier LightBGM Classifier
10 CROSS VALIDATION
Classification Model
SMOTE Class Imbalance
Data Set: 10,000 Features 2000 Observations
Train/Test and Validation Split
Validation Data
PREDICT RANGES AND CLASS PROBABILITY
GRID SEARCH/RANDOM SEARCH Hyperparameter Tuning
Validation Data
Train/Test Data
Final Model
10 CROSS VALIDATION
XGBOOST Regressor CATBOOST Regressor LightBGM Regressor
Regression Resampling Method

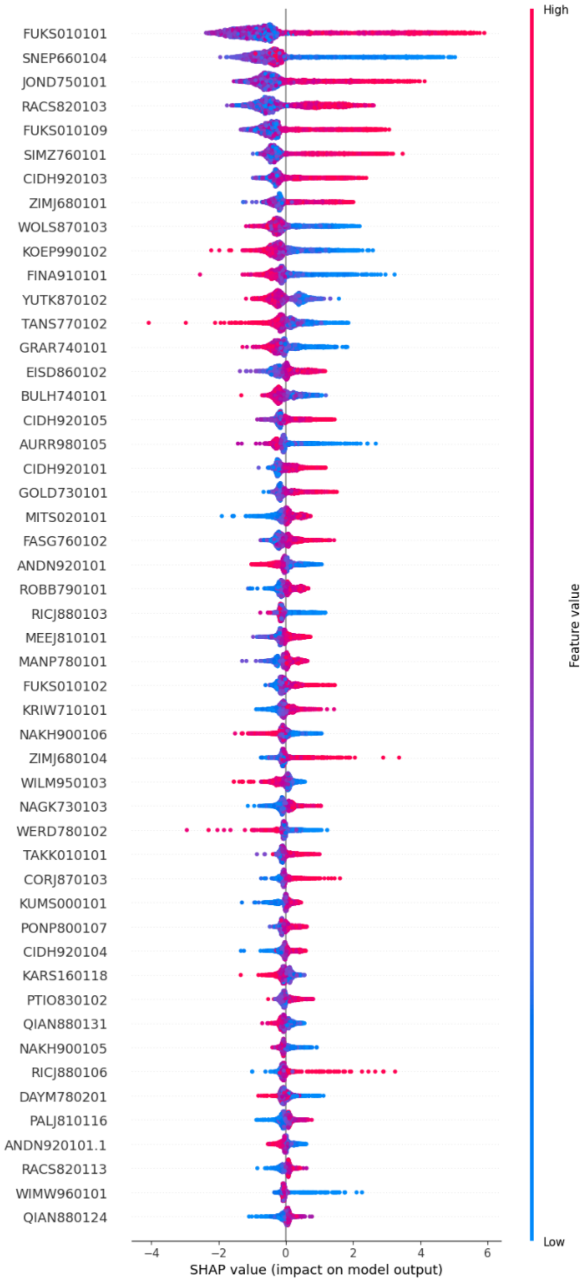
Chapter 3: New Model Development for Expanded Data Set.

Chapter 4: pH

Chapter 5: Stream lit Web application

APPENDIX:

Table 1: Model results from PCA dimensionality reduced data set


FUKS010101
SNEP660104
JOND750101
RACS820103
FUKS010109
SIMZ760101
CIDH920103
ZIMJ680101
WOLS870103
KOEP990102
FINA910101
YUTK870102
TANS770102
GRAR740101
EISD860102
BULH740101
CIDH920105
AURR980105
CIDH920101
GOLD730101
MITS020101
FASG760102
ANDN920101
ROBB790101
RICJ880103
MEEJ810101
MANP780101
FUKS010102
KRIW710101
NAKH900106
ZIMJ680104
WILM950103
NAGK730103
WERD780102
TAKK010101
CORJ870103
KUMS000101
PONP800107
CIDH920104
KARS160118
PTIO830102
QIAN880131
NAKH900105
RICJ880106
DAYM780201
PALJ810116
ANDN920101.1
RACS820113
WIMW960101
QIAN880124
SHAP value (impact on model output)
High
Feature value
Low