Machine Learning

Course Project Report

(Draft - 04, Team No: 6)

Title of the project: Water Quality Prediction Dataset

Student 1 : Abbinav Sankar Kailasam, abbinav.k-25@scds.saiuniversity.edu.in
Student 2 : Sounak Saha, sounak.s-25@scds.saiuniversity.edu.in

ML Category: Regression

---

1. Introduction

The quality of unfiltered water can be well described using the "power of hydrogen"(pH) value which indicates the acidity or basicity of any solution. The ideal pH of water is between 6.5 and 7.5 but this number can deviate when the quality of water is poor. As such, pH of water serves as a viable indicator to measure the quality of a given sample of water.

The objective is to forecast the water quality in terms of its pH value for the next day based on the input data. The input data consists of daily samples of 37 sites for a period of 423 days providing measures which can help predict pH values in Georgia, USA. The input consists of 11 features which includes volume of dissolved oxygen, temperature, and specific conductance (min, max and mean). The output to predict is the measurement of pH value of water (median).

- Problem Statement: Predict the spatio-temporal water quality in terms of pH value of unfiltered water.

2. Dataset and Features

The 11 features used to make the prediction include: (Inputs)

1. pH, water, unfiltered, field, standard units (Minimum)
   - Minimum pH of unfiltered water in standard units.
2. pH, water, unfiltered, field, standard units (Maximum)
   - Maximum pH of unfiltered water in standard units.
3. Specific conductance, water, unfiltered, microsiemens per centimetre at 25 degrees Celsius (Minimum)
   - Minimum microsiemens specific conductance of unfiltered water at 25 degrees Celsius
4. Specific conductance, water, unfiltered, microsiemens per centimetre at 25 degrees Celsius (Maximum)
   - Maximum microsiemens specific conductance of unfiltered water at 25 degrees Celsius
5. Specific conductance, water, unfiltered, microsiemens per centimetre at 25 degrees Celsius (Mean)
   - Mean microsiemens specific conductance of unfiltered water at 25 degrees Celsius
6. Dissolved oxygen, water, unfiltered, milligrams per litre (Minimum)
   - Minimum milligrams dissolved oxygen in unfiltered water per litre

7. Dissolved oxygen, water, unfiltered, milligrams per litre (Maximum)
    - Maximum milligrams dissolved oxygen in unfiltered water per litre
8. Dissolved oxygen, water, unfiltered, milligrams per litre (Mean)
    - Mean milligrams dissolved oxygen in unfiltered water per litre
9. Temperature, water, degrees Celsius (Minimum)
    - Minimum water temperature (degrees celsius)
10. Temperature, water, degrees Celsius (Maximum)
    - Maximum water temperature (degrees celsius)
11. Temperature, water, degrees Celsius (Mean)
    - Mean water temperature (degrees celsius)

The Dataset was taken from 37 different sites for 705 days giving a total of 26,085 samples. The resultant data is a (26,085 x 11) matrix with 26, 085 rows and 11 features.
This above input data and related features (independent variables) will help us make a prediction of the pH value of water (median) (dependent variable) thus forecasting the water quality.

2.1 Exploratory Data Analysis (EDA)

The Exploratory Data Analysis of the dataset will help us make some preliminary conclusions about the data and will help in visualising the data better. The dataset has 26,085 rows and 11 columns which is then split into 60% training and 40% testing data. EDA is conducted to both training and testing sets to check the similarity between the sets while at the same time making initial inferences.
(Note : the dataset was already split between training and testing in the matlab file provided, as such EDA was conducted to both sets separately)

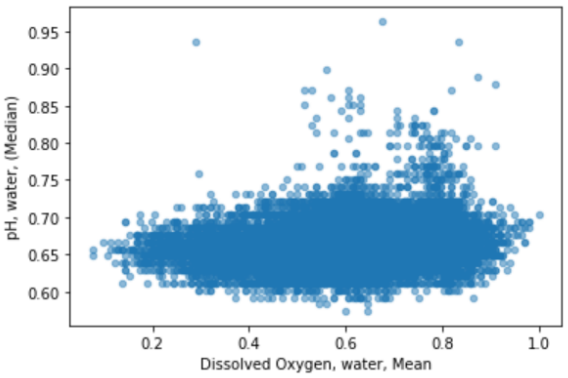
	0	1	2	3	4
Specific condunctance, water, Max	0.001131	0.001170	0.001326	0.014094	0.088109
pH, water, Max	0.884615	0.871795	0.884615	0.858974	0.858974
pH, water, Min	0.001120	0.001159	0.001198	0.001238	0.010766
Specific Conductance, water, Min	0.001113	0.001152	0.001250	0.003926	0.029297
Specific Conductance, water, Mean	0.677632	0.703947	0.677632	0.697368	0.684211
Dissolved Oxygen, water, Max	0.841463	0.829268	0.853659	0.829268	0.853659
Dissolved Oxygen, water, Mean	0.765152	0.772727	0.750000	0.772727	0.765152
Dissolved Oxygen, water, Min	0.787402	0.795276	0.755906	0.771654	0.755906
Temp, water, Mean	0.293750	0.293750	0.300000	0.296875	0.296875
Temp, water, Min	0.298077	0.301282	0.298077	0.294872	0.291667
Temp, water, Max	0.276163	0.276163	0.287791	0.279070	0.281977
pH, water, (Median)	0.648148	0.648148	0.648148	0.648148	0.648148

All the columns of the above data are type float64 with no null values. The training set has 15,651 samples (15,651 x 11) and the testing set has 10,434 samples (10,434 x 11).

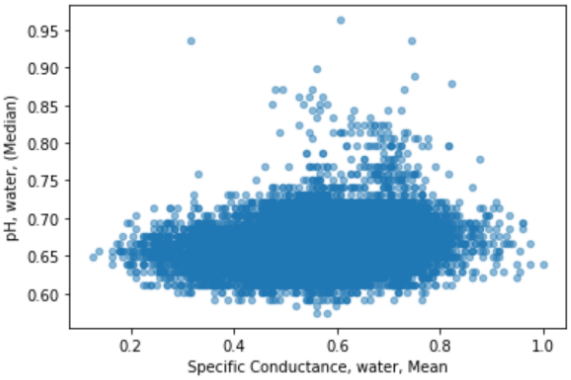
To understand the relationship between output - "pH, water, median" and its various inputs we create scatter plots and correlation matrices .

Sample EDA from training set :

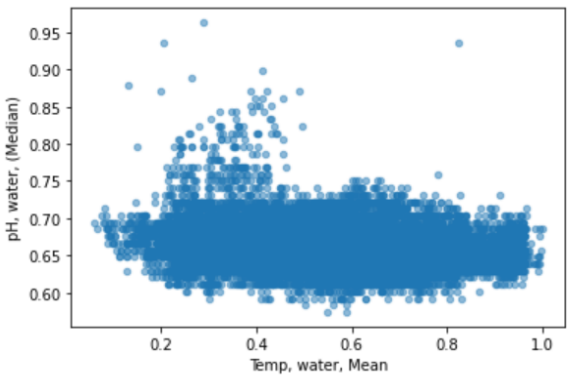
[1]Highest Correlation variable of 0.193960 (Direct correlation)



[2]Second highest correlation variable of 0.188911 (Direct correlation)

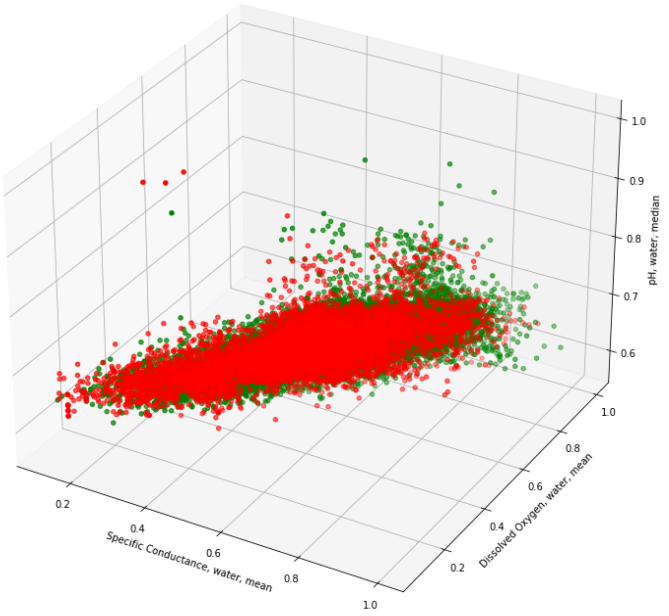


[3]Lowest correlated variable of -0.225402 (Inverse correlation)



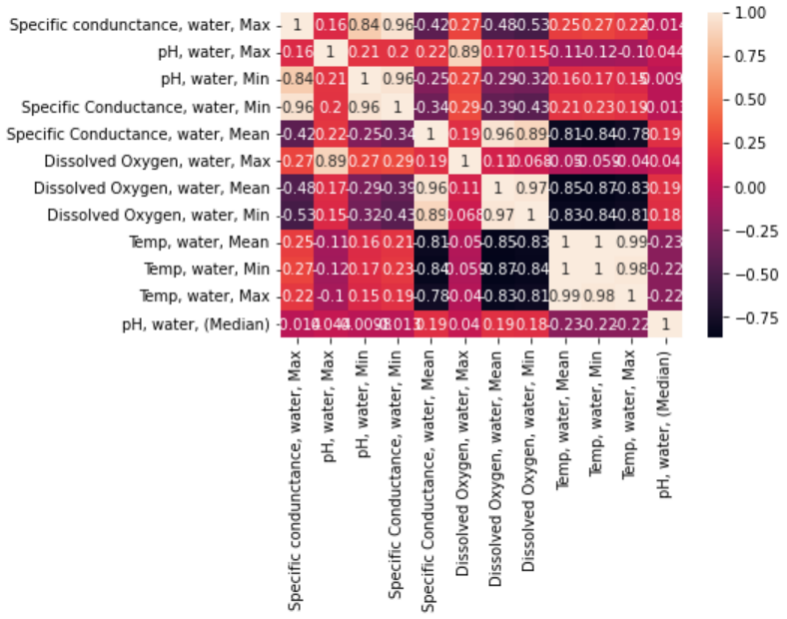
[4] 3D plot :

Relationship between inputs (Specific Conductance, mean and Dissolved Oxygen, mean) and output (pH of water, median)



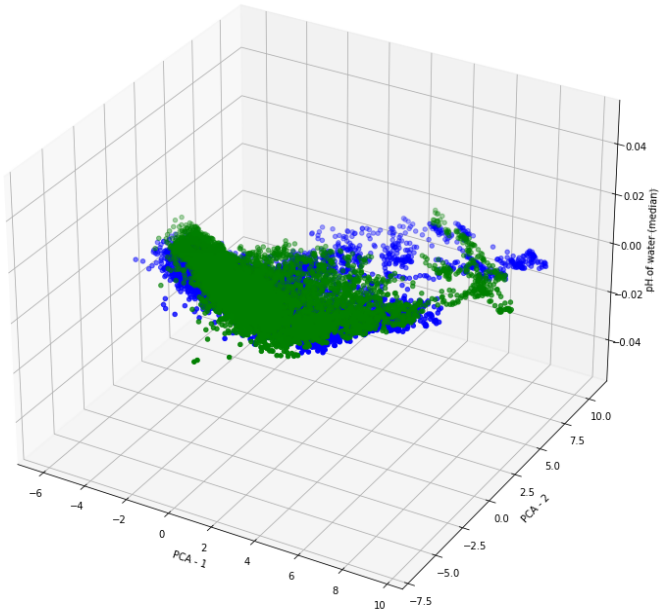
From the above plots, the relationship between “pH, water, median”(output) and the respective inputs is found to be linear but not directly proportional, which is a trend reflected in the correlation matrix (heatmap) shown below.

The last column of the heatmap represents the relationship b/w inputs and output

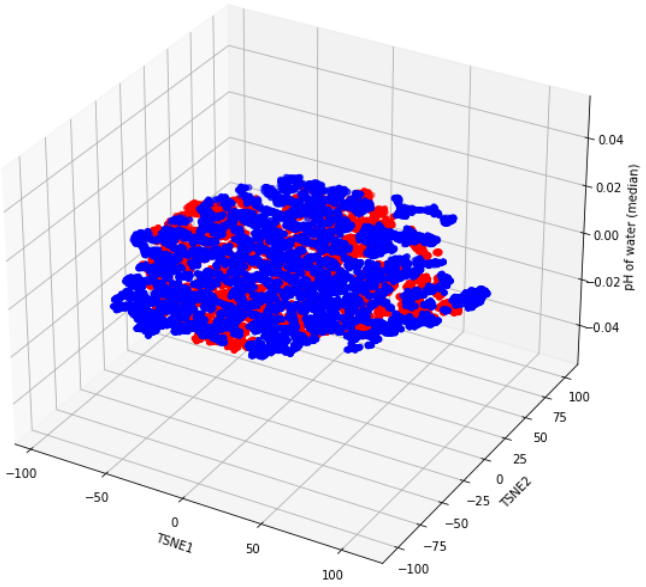


After Principal Component Analysis (PCA), the input set is converted to a 2 dimensional dataset (from a 11 dimensional data) enabling to visualise better :

[5] PCA : Relation b/w independent and dependent variables (training and testing)



[6] TSNE : Relation b/w independent and dependent variables (training and testing)



[1], [2], [3] and [4] showcase a linear relationship between the input and output variables.

3. Methods

The performance of all methods are measured by :

- Root mean square error (RMSE)
- Normalised root mean squared error (NRMSE)
- Mean absolute error (MAE)
- Normalised mean absolute error (NMAE)

3.1 Baseline - Linear Regression

- Brief description of the method :
  The baseline model that was implemented to the above dataset is the multivariate linear regression model (MLR). The MLR model learns a linear mapping from each multivariate input to each prediction.
  Linear Regression algorithm was implemented to the data producing its respective input parameters and the line of best fit (regression line). The MLR model ensures that the error between actual and predicted outputs are minimised.

3.2 Polynomial regression

- Brief description of the method :
  Polynomial Regression learns a non-linear mapping between the input and output variables by fitting a non-linear regression line between the point cloud.
  Polynomial Regression model was implemented to the data producing the input parameters of a n-degree polynomial. The resultant curve (or surface) of best fit ensures that the error between the actual and predicted outputs are minimised.

3.3 Regularisation

- Description for L1 / LASSO Regression :
  Lasso is a multivariate linear regression model with L1 regularisation over the input parameters. It is an algorithm that helps in the elimination of irrelevant parameters by reducing their weights to zero, thus helping in selection and regularisation of features. This algorithm helps prevent overfitting of the training set and generalises the model better.
  Lasso removed irrelevant features in the dataset by reducing their weights to zero, helping in more accurate pH (median) prediction.

$$J(\theta) = MSE(\theta) + 2\alpha \, \Sigma abs(\theta i)$$

- Description for L2 / Ridge Regression :
  Ridge is a multivariate linear regression model with L2 regularisation over the input parameters. It is an algorithm that keeps the model weights as small as possible, thus mitigating multicollinearity and prevents overfitting the training dataset.

Ridge kept the parameters as small as possible, producing more precise pH (median) prediction.

$$J(\theta)=MSE(\theta) + (\alpha/m)\ \Sigma(\theta(i))^2$$

- Description for ElasticNet Regression
  Elastic Net is a multivariate linear regression model that combines both the Lasso and Ridge method, to overcome the limitations of each method. This algorithm also helps mitigate the overfitting of the training dataset.
  Elastic Net combined the methods of Lasso and Ridge to provide a better prediction than Lasso or Ridge respectively.

$$J(\theta) =MSE(\theta) + r(2\alpha\ \Sigma abs(\theta i)) + (1-r)((\alpha/m)\ \Sigma(\theta(i))^2)$$

3.4 Support Vector Machine Regression

- Description for Linear SVM Regression :
  Linear SVM Regression learns a linear mapping between each input and output values. Linear SVM aims to find a hyperplane that best fits the data while minimising the error between the predicted and actual values. The hyperplane lies between a set of training data points, called support vectors, while allowing a certain margin of error or tolerance to avoid overfitting (maximised margin and minimised overfitting).

- Description for Kernel SVM Regression :
  Kernel SVM is used in regression when the point cloud is non-linear. As SVM is natively linear, the point cloud should be converted such that a hyperplane can be fitted to make predictions.

  - Description for Linear Kernel SVM :
    Linear Kernel SVM converts the 11 dimensional dataset to a higher (11+i) dimensional space using a linear function. SVM is operated in this (11+i) dimensional plane and fits a hyperplane in the point cloud such that the margin is maximised without overfitting with the help of support vectors.

  - Description for Polynomial Kernel SVM :
    Polynomial Kernel SVM converts the 11 dimensional dataset to a higher (11+i) dimensional space using a polynomial function (n-degree). SVM is operated in this (11+i) dimensional plane and fits a hyperplane in the point cloud such that the margin is maximised without overfitting with the help of support vectors.

  - Description for Radial Basis Function Kernel SVM :
    Radial Basis Function kernel SVM converts the 11 dimensional dataset to a higher (11+i) dimensional space by applying the gaussian function (-(||x(i) - x(j)||)/2(sigma)^2). SVM is operated in this (11+i) dimensional plane and fits a hyperplane in the point cloud such that the margin is maximised without overfitting with the help of support vectors.

3.5 Decision Tree Regression

- Description for Decision Tree Regression :
  Decision tree regression model breaks down a complex dataset into smaller subsets, forming a binary tree with each branch being a possible outcome. The decision tree is formed such that a pure node (mse=0) arrives as soon as possible. The decision tree algorithm stops when all leaf nodes are pure.

3.6 Ensemble Learning Models

- Description for Random Forest Regression :
  Random forest regression is an ensemble learning algorithm where decision trees are applied to random subsets of the original data. The mean value of all outputs produced, is taken as the output for random forest. This algorithm provides for a much stronger prediction than a single decision tree for the whole dataset.

- Description for AdaBoost Regression :
  AdaBoost Regression is an ensemble learning algorithm that combines several weak learners to obtain a strong learner. AdaBoost is a sequential learning algorithm where each new predictor corrects its predecessor's underfitting instances. Thus, each new learner focuses on harder instances than its predecessor.

- Description for Gradient Boost Regression :
  Gradient Boost Regressor is an ensemble learning algorithm that combines several weak learners to obtain a strong learner. Gradient Boost is a sequential learning algorithm where each new predictor corrects its predecessor's residual errors. Thus, each new learner minimises their residual errors compared to their predecessor.

4. Experiments & Results

4.1 Protocol

- Details about splitting into training and testing datasets :
  The dataset was split randomly into 60% training and 40% testing sets to implement the respective machine learning models.

- Preprocessing was done to the dataset :
  - Feature Scaling
    Standardisation : Experiments were conducted with a standardised input data, where the features were transformed to have mean=0 and std=1.

    $$X' = (X - \text{mean}) / \text{std}$$

  - Feature Selection :
    Select K Best : The input dataset retained the user specified highest scoring K best features (5). The score is determined by the correlation between the feature and output data.

- Feature Reduction :
  Principal Component Analysis (PCA) : The 11 dimensional input data was converted to 2 dimensional data using PCA. PCA reduces the dimensions of the input data while maintaining the spread of the data. Experiments were performed with this dataset.

4.2 Results

- Baseline Results using Multivariate Linear Regression:

MLR	**Raw Data**	Std Data	PCA Data	KBest Data
RMSE	**0.0293**	0.0293	0.0294	0.0292
NRMSE	**0.0443**	0.0442	0.0444	0.0442
MAE	**0.0211**	0.0211	0.0214	0.0211
NMAE	**0.0319**	0.0319	0.0324	0.0319
CV : MAE	**0.022 +/- 0.006**	0.022 +/- 0.006	0.022 +/- 0.005	0.022 +/- 0.006

- Polynomial Regression :

Polynomial	Raw Data	Std Data	PCA Data	KBest Data
RMSE	0.0309	0.0309	0.0295	0.0304
NRMSE	0.0467	0.0466	0.0445	0.0459
MAE	0.0215	0.0215	0.0215	0.0213
NMAE	0.0325	0.0325	0.0324	0.0321
CV : MAE	0.023 +/- 0.005	0.023 +/- 0.005	0.021 +/- 0.005	0.022 +/- 0.005

- Lasso Regression :

Lasso	Raw Data	Std Data	PCA Data	KBest Data
RMSE	0.0295	0.0295	0.0295	0.0295
NRMSE	0.0446	0.0446	0.0446	0.0446
MAE	0.0214	0.0214	0.0214	0.0214
NMAE	0.0323	0.0323	0.0323	0.0323
CV : MAE	0.022 +/- 0.006	0.022 +/- 0.006	0.022 +/- 0.006	0.022 +/- 0.006

- Ridge Regression :

Ridge	Raw Data	Std Data	PCA Data	KBest Data
RMSE	0.0293	0.0293	0.0294	0.0293
NRMSE	0.0442	0.0442	0.0444	0.0442
MAE	0.0211	0.0211	0.0214	0.0211
NMAE	0.0319	0.0319	0.0323	0.0319
CV : MAE	0.022 +/- 0.006	0.022 +/- 0.006	0.022 +/- 0.006	0.022 +/- 0.006

- Elastic Net Regression :

Elastic Net	Raw Data	Std Data	PCA Data	KBest Data
RMSE	0.0295	0.0295	0.0295	0.0295
NRMSE	0.0446	0.0446	0.0446	0.0446
MAE	0.0214	0.0214	0.0214	0.0214
NMAE	0.0323	0.0323	0.0323	0.0323
CV : MAE	0.022 +/- 0.006	0.022 +/- 0.006	0.022 +/- 0.006	0.022 +/- 0.006

- Linear SVM Regression

Linear SVM	Raw Data	Std Data	PCA Data	KBest Data
RMSE	0.0389	0.0377	0.0455	0.0385
NRMSE	0.0588	0.0569	0.0688	0.0581
MAE	0.0303	0.0286	0.0390	0.0297
NMAE	0.0458	0.0432	0.0589	0.0448
CV : MAE	0.034 +/- 0.013	0.035 +/- 0.013	0.038 +/- 0.008	0.033 +/- 0.013

- Kernel SVM Regression with raw data :

Kernel SVM (Raw)	RBF	Linear	Polynomial
RMSE	0.0399	0.0373	0.0394

NRMSE	0.0603	0.0563	0.0595
MAE	0.0307	0.0282	0.0299
NMAE	0.0463	0.0426	0.0453
CV : MAE	0.034 +/- 0.013	0.033 +/- 0.013	0.035 +/- 0.011

- Kernel SVM Regression with Standardised data :

Kernel SVM (Std)	RBF	Linear	Polynomial
RMSE	0.0411	0.0373	0.0501
NRMSE	0.0621	0.0563	0.0756
MAE	0.0314	0.0283	0.0433
NMAE	0.0474	0.0427	0.0655
CV : MAE	0.035 +/- 0.012	0.033 +/- 0.012	0.042 +/- 0.008

- Kernel SVM Regression with PCA data :

Kernel SVM (PCA)	RBF	Linear	Polynomial
RMSE	0.0439	0.0462	0.0499
NRMSE	0.0664	0.0698	0.0754
MAE	0.0357	0.0398	0.0440
NMAE	0.0539	0.0601	0.0665
CV : MAE	0.035 +/- 0.011	0.038 +/- 0008	0.042 +/- 0.008

- Kernel SVM Regression with Select K Best data :

Kernel SVM	RBF	Linear	Polynomial
RMSE	0.0416	0.0383	0.0494
NRMSE	0.0629	0.0579	0.0746
MAE	0.0328	0.0298	0.0431
NMAE	0.0495	0.0450	0.0651
CV : MAE	0.037 +/- 0.008	0.033 +/- 0.013	0.043 +/- 0.009

Hyperparameter Tuning on Kernel SVM regressor :

- Since Kernel SVM with raw data performed the best, hyperparameter tuning was performed on RBF kernel SVM with parameters C and gamma resulting in values 1 and 0.6 respectively.
  Results of RBF Kernel SVM with hyperparameter tuning : {Raw Data}
  RMSE : 0.0388
  NRMSE : 0.0586
  MAE : 0.0300
  NMAE : 0.0454
  CV : 0.033 +/- 0.011

- Decision Tree Regression :

Decision Tree	Raw Data	Std Data	PCA Data	KBest Data
RMSE	0.0411	0.0412	0.0371	0.0398
NRMSE	0.0621	0.0622	0.0560	0.0600
MAE	0.0288	0.0289	0.0272	0.0289
NMAE	0.0435	0.0436	0.0412	0.0437
CV : MAE	0.031 +/- 0.007	0.031 +/- 0.007	0.028 +/- 0.005	0.031 +/- 0.006

- Random Forest Regression :

Random Forest	Raw Data	Std Data	PCA Data	KBest Data
RMSE	0.0303	0.0303	0.0311	0.0309
NRMSE	0.0458	0.0458	0.0470	0.0467
MAE	0.0220	0.0220	0.0230	0.0224
NMAE	0.0333	0.0333	0.0347	0.0339
CV : MAE	0.024 +/- 0.006	0.024 +/- 0.006	0.024 +/- 0.005	0.024 +/- 0.006

Hyperparameter Tuning on random forest regressor :

- Since random forest with raw data performed the best, hyperparameter tuning was performed with parameters n_estimators and max_depth resulting in values 300 and 2 respectively.
  Results of random forest with hyperparameter tuning : {Raw Data}
  RMSE : 0.0293
  NRMSE : 0.0443
  MAE : 0.0211

NMAE : 0.0319
CV : 0.022 +/- 0.006

- AdaBoost Regression :

AdaBoost	Raw Data	Std Data	PCA Data	KBest Data
RMSE	0.0376	0.0374	0.0530	0.0381
NRMSE	0.0567	0.0564	0.0800	0.0576
MAE	0.0289	0.0288	0.0442	0.0294
NMAE	0.0427	0.0435	0.0667	0.0444
CV : MAE	0.033 +/- 0.010	0.033 +/- 0.011	0.041 +/- 0.016	0.034 +/- 0.011

- Gradient Boost Regression :

Gradient Boost	Raw Data	Std Data	PCA Data	KBest Data
RMSE	0.0350	0.0351	0.0345	0.0352
NRMSE	0.0529	0.0530	0.0521	0.0531
MAE	0.0258	0.0258	0.0257	0.0253
NMAE	0.0390	0.0390	0.0388	0.0382
CV : MAE	0.028 +/- 0.006	0.028 +/- 0.006	0.026 +/- 0.005	0.028 +/- 0.006

Hyperparameter Tuning on gradient boosting regressor :

- Since gradient boosting with KBest data performed the best, hyperparameter tuning was performed with parameter : n_estimators, resulting in value .
  Results of gradient boost with hyperparameter tuning : {KBest Data}
  RMSE : 0.0296
  NRMSE : 0.0446
  MAE : 0.0213
  NMAE : 0.0321
  CV : 0.023 +/- 0.0057

- After several experiments we can conclude that multivariate linear regression algorithm with raw data produced the best results compared to all other machine learning models. Polynomial regression, random forest and gradient boosting also provided comparable results.

5. Discussion

- The data has been trained using various regression models like : Multivariate linear regression(MLR), Polynomial regression, Lasso, Ridge, ElasticNet, Linear SVM, Kernel SVM, Decision Trees, Random Forest, AdaBoost and Gradient Boost.
- Experiments were conducted with Raw, standardised, PCA and SelectKBest data(6 features selected).
- As correlation between the input features and output variable is around 0 (close to no correlation), feature Reduction/feature Selection does not improve the performance of the model.
- Lasso and Elastic Net Regression converted the weights of all input features to 0 due to its low correlation with the output and provided results similar to multivariate linear regression.
- MLR is the best performing model for this dataset, succeeded by polynomial regression, random forest and Gradient Boosting.
- Hyperparameter tuning with gradient boosting and random forest greatly improved the accuracy of the algorithms but was not better than MLR.
- AdaBoost and SVM performed poorly compared to all other machine learning models as they had greater residual errors.

6. Conclusion

- The goal of the project was to ascertain a regression analysis between the input data - spatial factors, and output values - pH of water (median). Several experiments were conducted to find the regression algorithm with minimised residual errors. The multivariate linear regression algorithm performed the best with an average root mean square error of 0.0293 across all experiments (Raw, Std, PCA and KBest). The algorithm fitted a hyperplane in the point cloud with minimal residual errors between predicted and actual output values. Thus, this analysis proved that there exists linear dependency between spatial factors like dissolved oxygen, temperature, specific conductance (inputs) and pH value of water (output).

7. References

[1] UCI Machine Learning Repository: Water Quality Prediction Data Set

[2] Spatial Auto-regressive Dependency Interpretable Learning Based on Spatial Topological Constraints | ACM Transactions on Spatial Algorithms and Systems