# Report Biomedical Artificial Intelligence Course

**Written by Kayla Queenazima Santoso (20/456076/TK/50206)**

## Introduction

In Indonesia, the issue of ensuring a safe supply of drinking water poses a serious challenge. Many households rely on water sources and must be aware of the crucial mineral content and the potential risks of heavy metal contamination. The limitations in detecting heavy metal contaminants and minerals in water, coupled with constraints in laboratory testing, further complicate the situation. By that, producing self-monitoring tool as the first step of water screening quality are very important and can be one step to increase the quality of life in Indonesia.

## The Goals

This project focuses on exploring the response of an Electronic Tongue (E-Tongue) sensor and developing a machine learning (ML) algorithm for identifying the types of solutions in mineral and heavy metal samples. The mineral samples consist of calcium, potassium, and magnesium, and lead. This project was currently still running under competition and funding of PKM.

## Profile of the Data

- How the Data Gained?

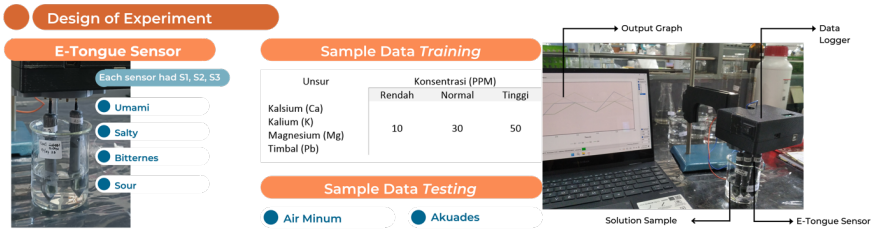



Fig. Design of Experiment and Arrangement of Sampling

Data gained from a set of E-Tongue that consists of 4 sensors taste there are umami, salt, bitterness, and sour. Each sensors had 3 amount of similar designed sensors (S1, S2, and S3). The class of classification case is Ca, K, Mg, and Pb with variety of concentrations are 10, 30, and 50 PPM.

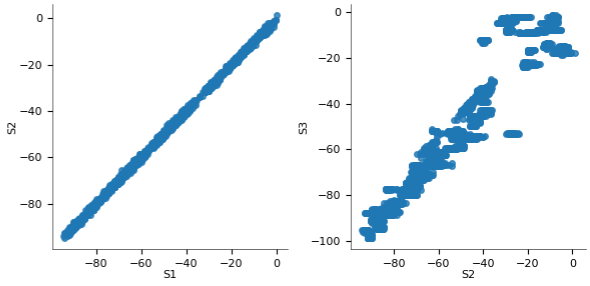
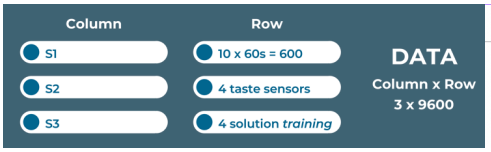



Fig. S1, S2, and S3 Compared to Each Other

In sensor making, each sensor might be inconsistent to each other. That might be happen due to production error or instrument error. Overall, S1 against S2 is more consistent than S3 against the other sensors. This shows that the validity of the S3 data is questionable, with the

hypothesis that there is interference in instrument input of S3 by looking into consistency of inconsistency S3 for every sample and every sensor tastes.

- The Data We Gained




| Nilai | Umami | Asin | Pahit | Asam |
|---|---|---|---|---|
| Mean | -63.668547 | -79.801816 | -21.134530 | -74.570214 |
| STD | 12.273975 | 14.375874 | 8.608179 | 13.243998 |
| Min | -88.000000 | -104.000000 | -60.000000 | -100.000000 |
| Max | -28.000000 | -20.000000 | 4.000000 | -26.000000 |

Fig. Profile of The Data

$$V_{rel} = V_s - V_r$$

$V_{rel}$: *nilai tegangan relatif*

$V_r$ : *nilai tengan reference*

$V_s$ : *nilai tegangan sampel*

Fig. Equation of Relative Value

The data produced using 3 amount of sensors with 4 type tastes that sampled 10 times for 1 minute (60 seconds, 60 point of data) for 4 classes resulted into a data with 3 columns and 9600 rows. Each data then processed to get the relative value of each response as the Fig above. For each class, Fig. Profile of the Data show that all tastes averages, standard deviations, and maximum values are similar except sour sensor (Pahit). Sour sensor show higher mean, lower standard deviation, and much higher maximum value than others. The anomalies of sour data distribution might be a clue of insignificance different response of sour taste sensor. All the mean are negative show that the response voltage had tendency to be much lower than reference voltage. With variety of concentration, its normal to have around 10 standard deviation. Even from that we know in variety of 10 to 50 PPM might not different to each other.

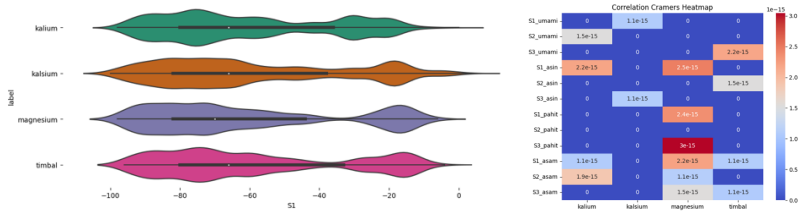
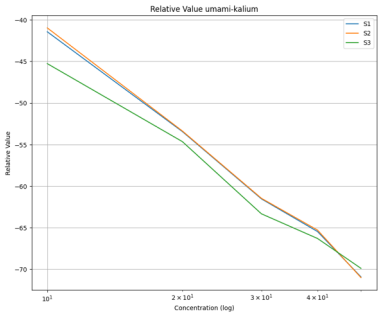
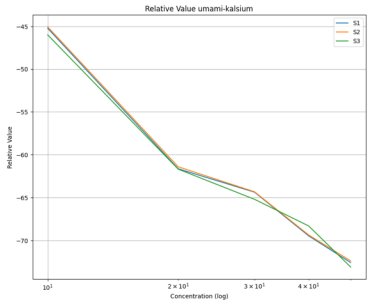
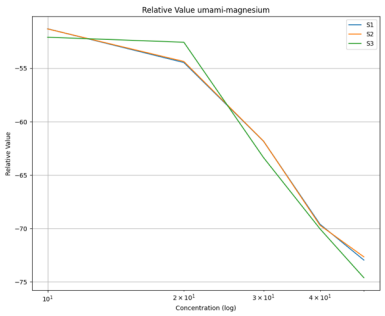
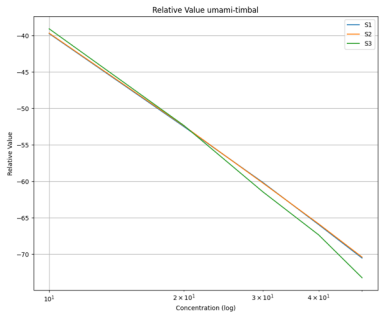


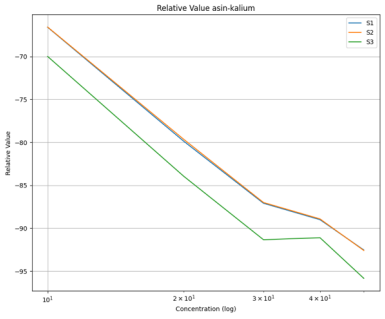
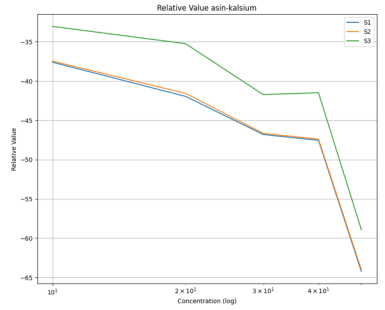

Fig. Violin Graph of Each Class and Correlation Cramers for Each Features

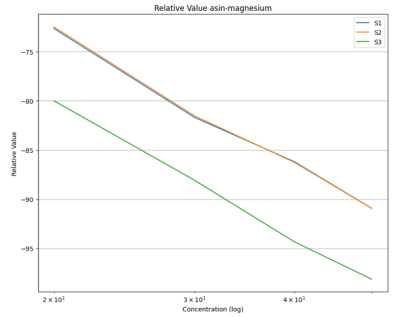
The distributions of data in each class can be show by the violin graph above. From the violin graph, Calcium have the most wide distributions of responses while lead (Timbal) are the less. Despite of that, each class show similar pattern of amount of data density in range of -100 to -50 than decreased significant and got increased density again although in different range for each class. The violin graph not really show different value of each class. While the correlation cramers show correlation of each sensors (S1, S2, and S3) for each tastes to its classes. Highest score is very low (3e-15) indicating that might need another process to get better model of classification.

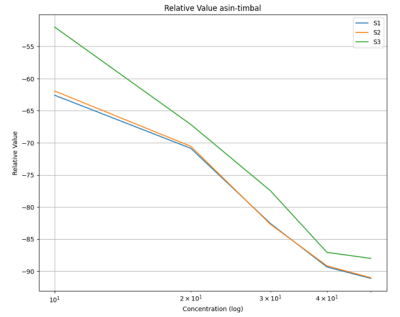
**Umami**

| Kalium | Kalsium |
| --- | --- |
|  |  |
| Magnesium | Timbal |
|  |  |

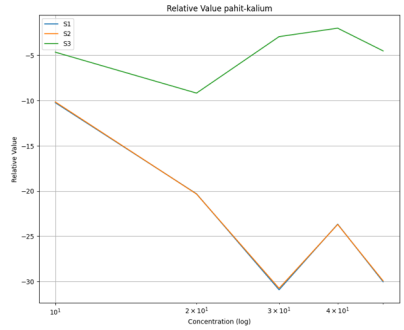
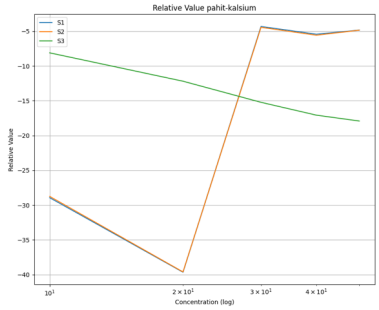
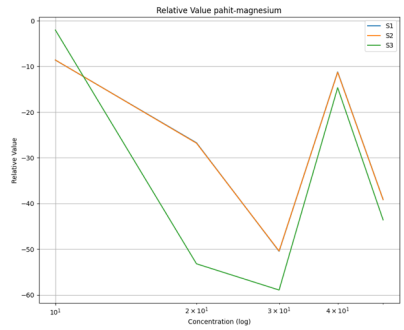
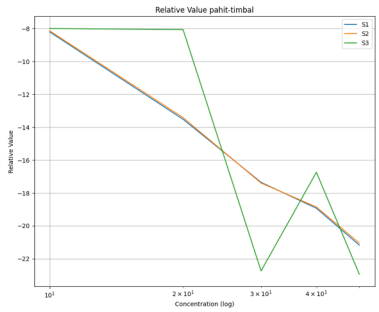
**Salt**

| Kalium | Kalsium |
| --- | --- |
|  |  |
| Magnesium | Timbal |



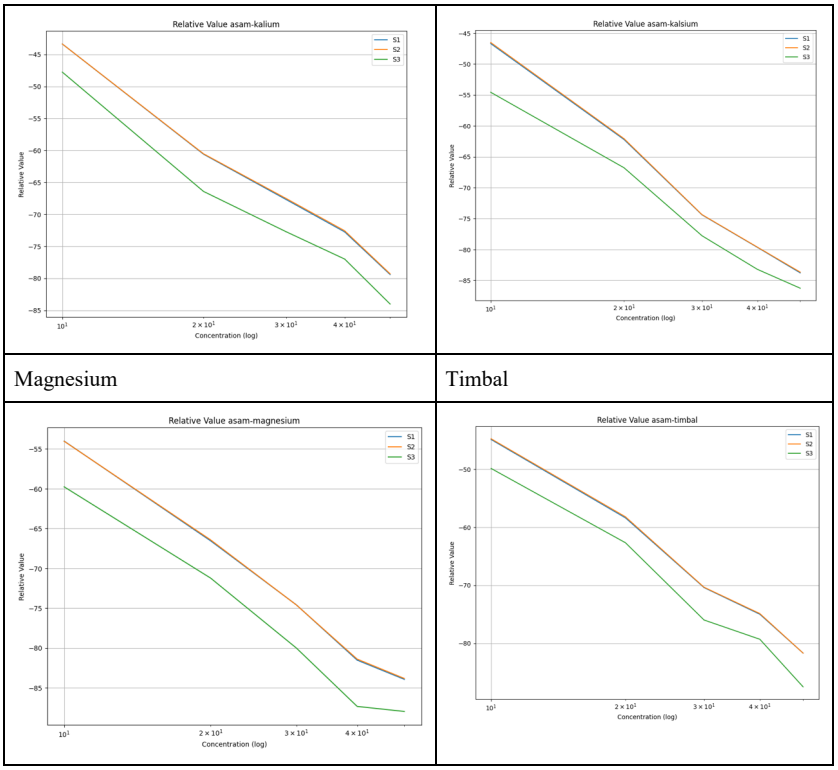

**Bitter**

| Kalium | Kalsium |
|---|---|
|  |  |
| Magnesium | Timbal |
|  |  |



**Sour**

| Kalium | Kalsium |
|---|---|




Tables. Each Tastes for Each Classes Plot Logaritimically by its Concentrations

The Tables above show that it's have different range of response in each classes for its each concentrations.

**Principal Component Analysis**

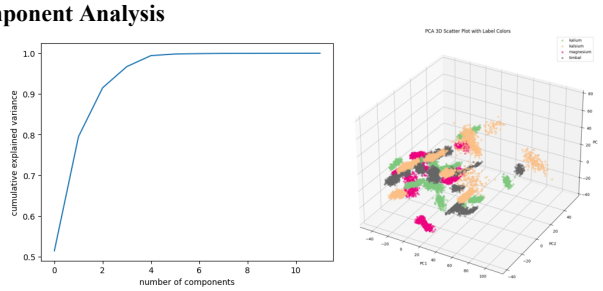


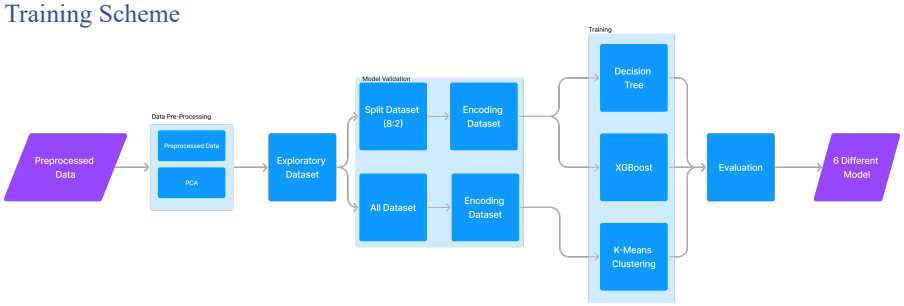

Fig. PCA Cumulative Explained Variance in Number of Components and PCA Plot of 3 Components

PCA stands for Principal Component Analysis. It is a statistical technique used to reduce the dimensionality of a dataset while retaining as much of the original variation as possible. PCA works by identifying the directions of maximum variance in the data and projecting the data onto a lower-dimensional subspace spanned by these directions, called principal components.

The resulting principal components are uncorrelated and ordered by the amount of variance they explain [1]. The PCA Cumulative graph show that 3 components already be good for 90% of coverability of all responses. Besides that, the plot of 3 components show that each class had grouped even the density of each class can be seen as layered into each other.

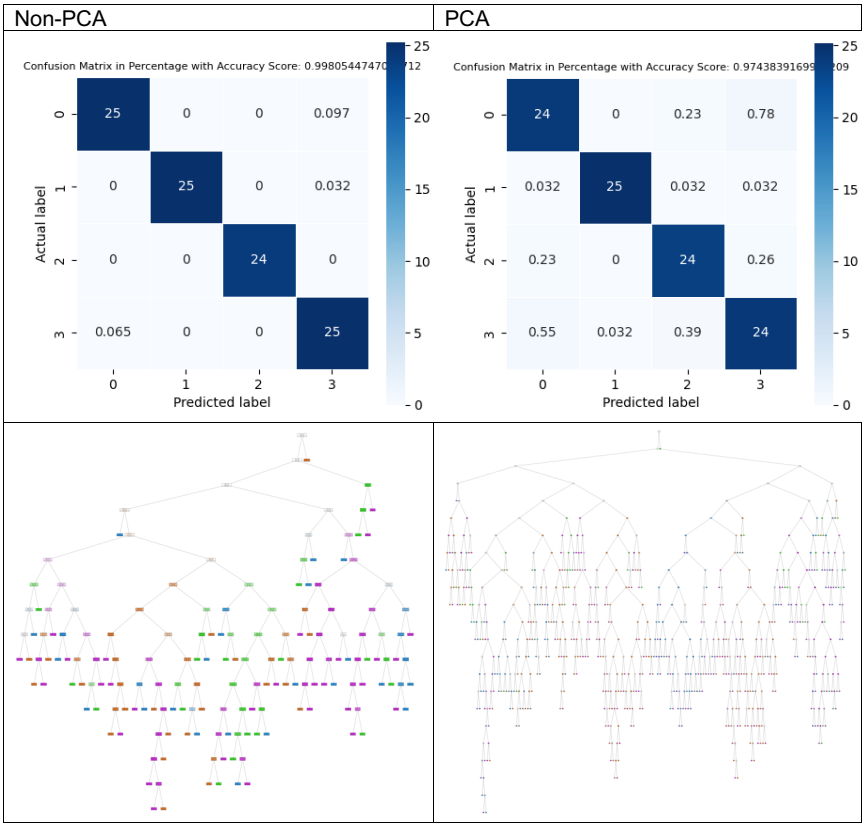
Commented [MP1]: A. M. Martinez and A. C. Kak, "PCA versus LDA," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 228-233, 2001.

## Training Scheme



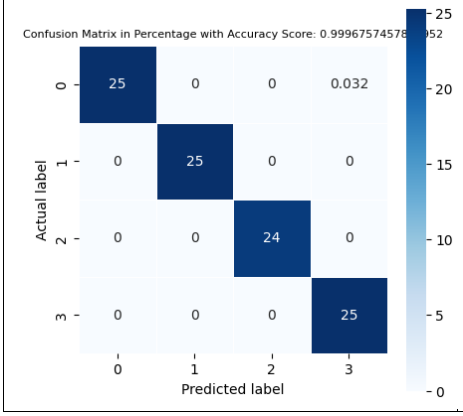
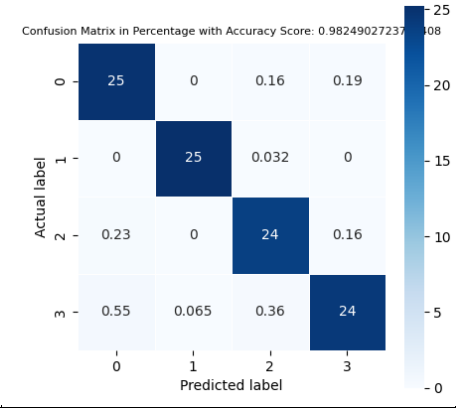

## Result

- Decision Tree

| Non-PCA | PCA |
|---|---|
|  | |

On Decision Tree, data that not yet reduced using PCA seems get better in accuracy and bias by what shown in Confusion Matrix of Non-PCA above. While the reduced data using PCA into 3 components got more biased model and complex decision tree by what shown in Confusion Matrix of PCA and Tree Logic Visualizations of PCA above.

- XGBoost

| Non-PCA | PCA |
|---|---|
|  |  |



While on XGBoost, the accuracy slightly increases from before. But the reduced data (PCA) still seems lower than Non-PCA either accuracy or bias.

**XGBoost and Decision Tree**

**Feature Importance of Model Trained on Data Non-PCA**

| index | Columns | Importance Decision Tree | Importance XGBoost |
|---|---|---|---|
| **0** | S1_umami | 0.09468218195461237 | 0.10778356343507767 |
| **1** | S2_umami | 0.0001921768325967800 5 | 0.01956545002758503 |
| **2** | S3_umami | 0.09169515590943468 | 0.06279657781124115 |
| **3** | S1_asin | 0.16052533197385022 | 0.14708241820335388 |
| **4** | S2_asin | 0.03506250750927733 | 0.04154161363840103 |
| **5** | S3_asin | 0.06322572268954876 | 0.054061148315668106 |
| **6** | S1_pahit | 0.0933218245514886 | 0.0999603122472763 |
| **7** | S2_pahit | 0.004579353005469969 | 0.16506412625312805 |
| **8** | S3_pahit | 0.1877157498374312 | 0.10922998934984207 |
| **9** | S1_asam | 0.2153268449643699 | 0.10940389335155487 |
| **10** | S2_asam | 0.0001857872942572202 6 | 0.009822765365242958 |
| **11** | S3_asam | 0.05348736347766291 | 0.0736882463097572 |

**Feature Importance of Model Trained on Data PCA**

| Index | columns | Importance Decision Tree | Importance XGBoost |
|---|---|---|---|
| **0** | comp1 | 0.3393778807602677 | 0.33891627192497253 |
| **1** | comp2 | 0.3901292650394198 | 0.3400021195411682 |
| **2** | comp3 | 0.27049285420031244 | 0.3210815787315368 |

From 2 tables above, it can be seen that importance value in PCA higher than Non-PCA. Both model shown highest importance feature is S1 Asam or the 1st sensor of sour taste. While accuracy and bias of Decision Tree better at non-PCA than PCA data also better than XGBoost, in feature importance, XGBoost got better than Decision Tree on PCA data.

- Clustering

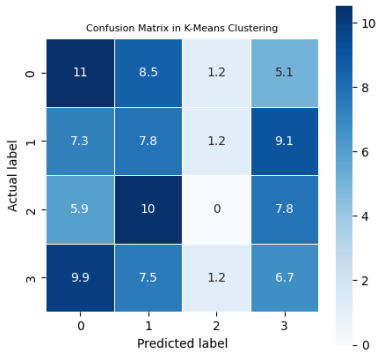



Fig. Confusion Matrix of K-Means Clustering

The K-Means Clustering set to 4 clusters than compared to class that already classified by the label before, result to Fig. Confusion Matrix of K-Means Clustering that show relative distributed bias for each label.

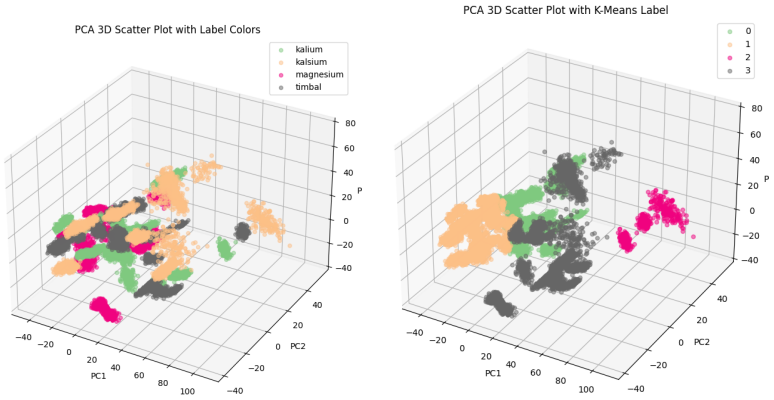



Fig. 3D Scatter Plot of PCA Labeled Classes (left) and PCA Labeled Clusters (right)

The scatter plot above show significant different of colors. The distribution of colors show K-Means clustering not yet got close to the expected value.

## Conclusion

From the results above, it can be concluded that with PCA and only 3 components can get relatively similar metrics to the complete data. Its potential if the case needed low computations or less data is better. While the clustering not yet can classified the case of solution differentiation show it needed another algorithms to try.

## Reference

[1] A. M. Martinez and A. C. Kak, "PCA versus LDA," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 228-233, 2001.

**Repository:** https://github.com/kaylaque/etongue/blob/main/modelling_etongue_AI.ipynb