ML PROJECT REPORT

Model Development Report

1. Preprocessing Steps and Rationale

The data were imported and meticulously examined for consistency. There were no missing data points. Visualization was performed next to see hidden patterns, distribution, and the presence of outliers. Sensor drift data were quantified and viewed through Principal Component Analysis (PCA) to inspect whether there had been any trends or systematic deviations in time impacting the model.

To handle outliers, several detection methods were investigated. Ultimately, the Isolation Forest method was used for outlier removal as it performs best at identifying anomalies in the dataset

2. Insights from Dimensionality Reduction

Dimensionality reduction methods, such as PCA, were utilized to examine variance in the dataset. This facilitated redundant or less informative feature detection and helped to support improved model generalization. PCA visualization informed understanding of clustering trends and sensor drift effects.

3. Model Selection, Training, and Evaluation

Two models were evaluated for vomitoxin prediction:

1. Neural Network
2. XGBoost

Hyperparameter tuning was applied to both models to improve performance. The evaluation criteria for each model are as follows:

Neural Network Performance:

Mean Absolute Error (MAE): 1823.40
Root Mean Squared Error (RMSE): 6394.47
R-squared (R2): 0.528

XGBoost Performance:

Mean Absolute Error (MAE): 125.91
Root Mean Squared Error (RMSE): 164.93
R-squared (R2): 0.9997

4. Key Findings and Suggested Improvements

The XGBoost model outperformed the Neural Network on all performance measures by a wide margin, with an almost perfect R2 score. This indicates that the relationships between features in the dataset were nicely picked up by XGBoost's ensemble method.
To enhance the Neural Network's performance, additional hyperparameter tuning would be worth exploring, including:
Changing the number of hidden layers and neurons
Trying different activation functions
Using sophisticated optimization methods like adaptive learning rates. Using dropout and batch normalization to avoid overfitting. Moreover, feature engineering methods and increasing the dataset with more meaningful attributes can further improve overall model performance.