Corn Mycotoxin Hyperspectral Imaging Data Prediction Report

Data Preprocessing Steps and Rationale:

1. Data Exploration:

- Loaded the dataset and inspected for missing values, outliers, and inconsistencies.
- Checked for missing values using “isnull().sum()”.
- Checked for outliers using box plots.

2. Normalization:

- Separated features (X) and the target variable (DON_concentration).
- Applied normalization using StandardScaler to ensure the features have a **MEAN** of 0 and **STANDARD DEVIATION** of 1.

3. Data Visualization:

- Created a line plot using matplotlib and seaborn to visualize the average reflectance across 100 wavelength bands.
- Generated a heatmap for sample comparisons to understand the Corn Hyper-Spectral data characteristics.

Insights from Dimensionality Reduction of Data:

1. Principal Component Analysis (PCA):

- Applied PCA to reduce dimensionality of the spectral data.
- Explained variance by the top 10 principal components was sufficient to capture significant information from the dimensions.
- Visualized the reduced data using a 2D scatter plot, revealing patterns and relationships within the dataframe.

2. t-SNE (t-Distributed Stochastic Neighbor Embedding)(Optional):

- Implemented t-SNE for a more intuitive representation of the data.

- Visualized the reduced data in a 2D scatter plot, highlighting clustering patterns and separability of samples.

Model Selection, Training, and Evaluation:

1. Model Selection:

- Chose MLPRegressor(Multi-Layered Perceptron Regressor) for it's simplicity performance and ease of use.
- Chose XGBRegressor for its performance and scalability in regression tasks.
- Defined a parameter grid for hyperparameter tuning.

2. Hyperparameter Tuning:

- Used GridSearchCV to find the best hyperparameters (n_estimators, max_depth, learning_rate, subsample, and colsample_bytree).
- Identified the optimal parameter combination for improving the model performance.

3. Model Training:

- Splitted the dataset into training (80%) and testing (20%) sets.
- Trained the XGBRegressor model with the best parameters on the training set.
- Saved the trained model using pickle for future use.

4. Model Evaluation:

- Evaluated the model using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 Score for XGBoost Regressor model.
- Results:
  - MAE: 0.71
  - RMSE: 0.89
  - R^2 Score: 0.65
- Visualized actual vs. predicted values using a scatter plot.
- Out of 3 chosen models only XGBoost Regressor model gave the best results.

Key Findings and Suggestions for Improvement:

1. Key Findings:

- The model performed reasonably well, with an R^2 score of 0.65(Acceptable), indicating that about 65% of the variance in DON concentration could be explained by the model.
- Dimensions which are reduced using PCA and t-SNE provided valuable insights into the data characteristics and improved model training efficiency.

2. Suggestions for Improvement:

- **Model Complexity:** For real-world , we can experiment with more complex models like Convolutional Neural Networks (CNNs) or Long Short-Term Memory networks (LSTMs) for potentially better performance of the model.
- **Hyperparameter Tuning:** We can further refine model hyperparameters using techniques like RandomizedSearchCV or Bayesian Optimization.
- **Regularization:** We can apply techniques like L1 and L2 regularization techniques to prevent overfitting and increase the model performance.