REPORT

**Hyperspectral Imaging Data Analysis for Mycotoxin Prediction in Corn Samples**

**Introduction**

**Hyperspectral imaging (HSI) is a valuable tool for analyzing spectral data across different wavelengths. This study explores how HSI data can be used to predict the concentration of deoxynivalenol (DON), a harmful mycotoxin found in corn. The process involves cleaning and preparing the data, reducing its complexity, training a machine learning model, and evaluating its performance.**

**Tasks-1(Data Exploration and Preprocessing)**

The dataset consists of spectral reflectance values across multiple wavelengths for various corn samples, with DON concentration as the target variable. The following preprocessing steps were performed:

* **Handling Missing Values: Missing data was either filled in using average values or removed if too much information was missing.**
* **Normalization: The data was scaled using Min-Max normalization to keep all values between 0 and 1.**
* **Visualization: Visualizing Spectral Bands for Reflectance Data- refers to creating a graphical representation of how different wavelengths**
  + **Line plots were used to show how reflectance changes across different wavelengths.**
  + **Heatmaps helped compare spectral responses between samples, making it easier to identify patterns and inconsistencies.**

**TASK-2 (Dimensionality Reduction)**

Principal Component Analysis (PCA) was employed to reduce dimensionality and extract the most significant features.

It is a data reduction technique. It helps us simplify complex data while keeping the most important information.

* **The first few principal components captured more than 95% of the total variance, ensuring that the essential characteristics of the dataset remained intact.**
* **Visualization:**
  + **A 2D scatter plot of the top two principal components revealed patterns in the data, allowing for better interpretation and understanding of spectral variations.**
  + **A 3D plot of the first three principal components provided deeper insights into the data structure, helping to visualize clustering trends and relationships between samples.**

**Using PCA, I improved computational efficiency while maintaining the most relevant spectral details. This step helped streamline the machine learning model, making it more effective in predicting DON concentration.**

**TASK-3(Model Training)**

This project focuses on training a deep learning model using an MLP architecture for classification. The dataset was preprocessed, and a model was trained with hyperparameter tuning to optimize performance.

**Data Preprocessing**

* The dataset was loaded from a CSV file.
* Features (X) and labels (y) were separated.
* Features were normalized using **StandardScaler** for better model performance.
* Labels were encoded into numerical values using **LabelEncoder** for compatibility with the neural network.
* The dataset was split into **80% training** and **20% testing** for evaluation.

**Model Selection & Training**

* A **Sequential Neural Network (MLP)** was created using TensorFlow/Keras with the following layers:
* Input layer (features)
* **Hidden Layers:** Two or more layers with ReLU activation
* **Output Layer:** Uses Softmax activation for multi-class classification
* The model was compiled using:
* **Loss Function:** sparse\_categorical\_crossentropy (since labels were integers)
* **Optimizer:** adam (adaptive learning rate for efficient training)
* **Metric:** accuracy (to measure model performance)
* The model was trained for **50 epochs** with a batch size of **32**, and validation accuracy was monitored.

**Hyperparameter Tuning**

* Used **Keras Tuner** (RandomSearch) to find the best model configuration.
* Tuned parameters:
* Number of hidden layers
* Number of neurons per layer
* Dropout rates
* Learning rate
* The best hyperparameters were selected based on validation accuracy.

**TASK-4(Model Evaluation)**

I evaluated the model using three main metrics:

1. **Mean Absolute Error (MAE):** This tells how much the model’s predictions are off on average. A lower number means better predictions.
2. **Root Mean Squared Error (RMSE):** This also measures error but gives more importance to bigger mistakes. A smaller value is better.
3. **R² Score:** This shows how well the model explains the actual data. If it's close to 1, the model is doing well. If it’s low, the model needs improvement.

To **visualize** the results, I plotted actual vs. predicted values on a scatter plot. If most points are near the **diagonal line**, it means the model is predicting well. If they are scattered far away, the model is making mistakes.

**Limitations of the Model:**

* The model might work well on the training data but not on new data (**overfitting**) or might not learn enough patterns (**underfitting**).
* It may be missing some important information (features) that could improve predictions.
* If the data has errors or unusual values (**outliers**), the model might not perform well.
* If the relationship between input and output is complex, a simple model like this may not be the best choice.