Mycotoxin Prediction using Machine Learning

Overview

This project aims to develop a machine learning pipeline to predict mycotoxin (DON) concentration in hyperspectral corn data. The goal is to create a robust, modular, and production-ready solution that can analyze spectral reflectance data and produce accurate predictions.

Approach & Methodology

- **Data Source:** The dataset (MLE-Assignment.csv) contains hyperspectral readings from corn samples with mycotoxin concentration values.
- **Preprocessing:**
    - Standardization of input features using `StandardScaler or you can also use MinMaxScaler`.
    - Dimensionality reduction using `PCA` (Principal Component Analysis) to reduce noise and improve model efficiency.
- **Model Selection:**
    - A **Convolutional Neural Network (CNN)** is used due to its ability to extract spatial patterns from spectral data.
    - Regularization techniques like Dropout and Batch Normalization are incorporated.
- **Evaluation Metrics:**
    - Mean Absolute Error (MAE): 0.20
    - Root Mean Squared Error (RMSE): 0.49
    - R^2 Score: 0.85

Pipeline Implementation

The project is divided into three main components:

1. Streamlit App (`app.py`)

A web-based UI for uploading data, performing predictions, and visualizing results.

- Users upload a CSV file.
- Preprocessing and feature extraction are performed.
- The trained CNN model predicts mycotoxin levels.
- Results are displayed interactively using Matplotlib and Seaborn.

2. Machine Learning Pipeline (`pipeline.py`)

A structured pipeline for training and evaluating the model.

- **Preprocessing:** Standardization, PCA for feature reduction.
- **Modeling:** CNN architecture with Conv1D layers.
- **Training & Evaluation:** Train-test split, Early Stopping, evaluation metrics calculation.
- **Logging:** Logs key steps for debugging and reproducibility.

3. Jupyter Notebook (`predict_mycotoxin_levels.ipynb`)

- Exploratory Data Analysis (EDA) on hyperspectral data.
- Model performance visualization.

Note: Hyperparameter tuning was not implemented to avoid unnecessary complexity and maintain the model's robustness and simplicity.

How to Run the Code

Run the Machine Learning Pipeline:

python pipeline.py /path/to/MLE-Assignment.csv

OR

Run the Unit Tests:

You can test the pipeline implementation using the unit test script:(just change the path of file)

pytest unit_tests.py

Run the Streamlit Application:

streamlit run app.py

Future Enhancements

- **Model Optimization:** Experiment with transformer-based architectures for better performance (eg LSTM, RNN, Transformer). it would perform great on the sequential data
- **Feature Engineering:** Explore alternative dimensionality reduction methods beyond PCA.
- **Deployment:** Convert the pipeline into a deployable API.
- **Integration:** Automate data collection from IoT-enabled sensors.

Conclusion

Through this project, I have successfully built a pipeline to predict DON concentration in corn using spectral data. We started by carefully preprocessing the data, handling its high dimensionality with PCA, and then built a CNN-based predictive model. The results showed strong performance, with an **MAE of 0.20, RMSE of 0.49, and R^2 of 0.85**, indicating that our model does a great job at making accurate predictions.

What makes this approach effective is the combination of dimensionality reduction and deep learning, allowing us to extract meaningful patterns from complex spectral data. Additionally, by wrapping the pipeline in a **Streamlit app**, we made it user-friendly and easy to interact with.

That said, there's always room for improvement. We could experiment with different architectures, explore transformer-based models, or fine-tune hyperparameters to push performance even further. Real-world deployment might also require handling larger datasets and making the system more scalable.

Overall, this project is a strong step toward automating mycotoxin prediction, reducing manual testing efforts, and improving food safety.