Predicting Mycotoxin Levels in Corn Using Hyperspectral Imaging Data

1. Introduction

This assignment focuses on processing **hyperspectral imaging data** to predict **mycotoxin (DON) concentration** in corn samples. Given spectral reflectance data across multiple wavelengths, the goal is to:

- **Preprocess the data** (handling missing values, normalization).
- **Visualize spectral bands** to explore patterns.
- **Apply dimensionality reduction** using **PCA** to extract relevant features.
- **Train regression models** (**Random Forest, XGBoost, CNN**) to predict mycotoxin levels.
- **Evaluate models** and draw actionable insights for future improvements.

2. Dataset Overview

- The dataset contains **hyperspectral reflectance values** across **448 wavelength bands** for various corn samples.
- The target variable is **DON concentration** (a mycotoxin harmful to food safety).
- **Challenge:** The dataset has **high dimensionality (448 features)**, requiring dimensionality reduction for better model performance.

3. Preprocessing Steps and Rationale

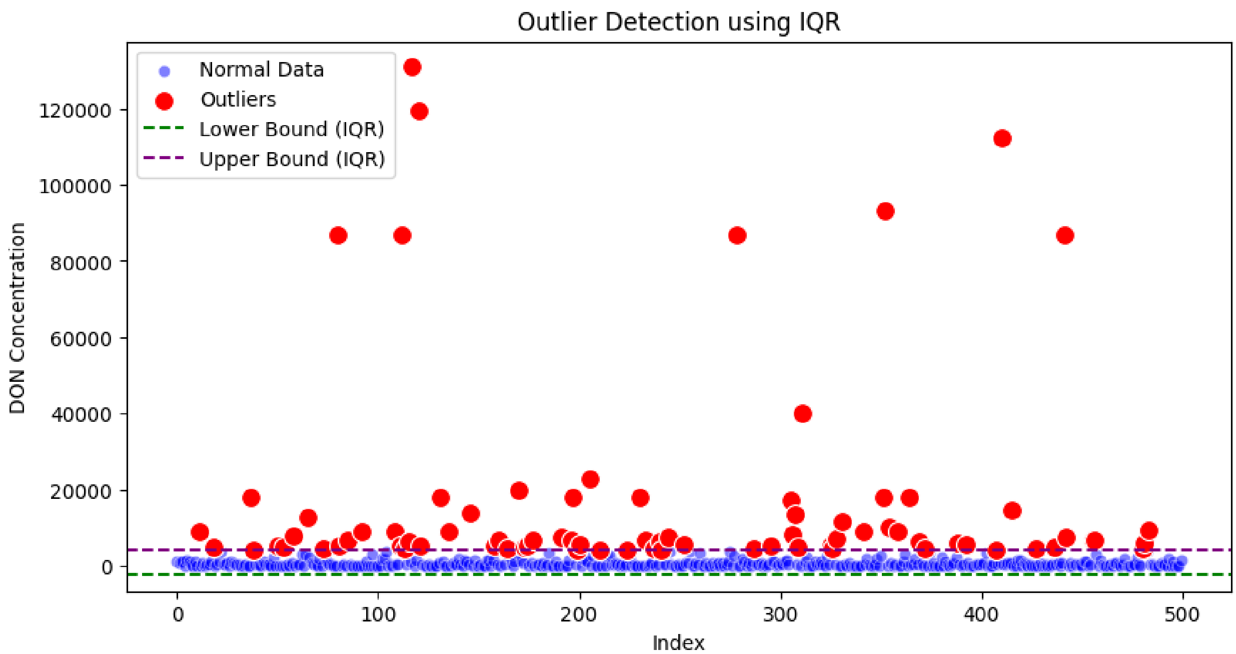
3.1 Handling Missing Values

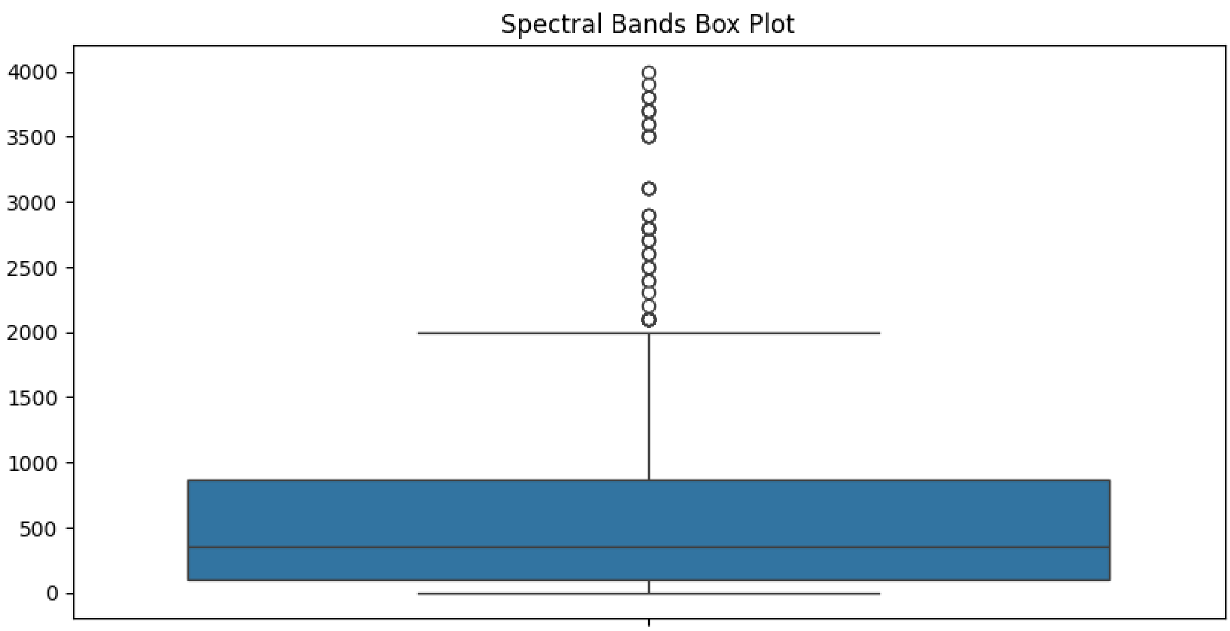
- **Checked for missing data** → No Missing Values found.

3.2 Outlier Detection & Handling

- **Used IQR (Interquartile Range) to detect outliers.**
- **Observation:** Removing outliers **significantly reduced model performance**, so they were retained.

Visualization:
(Outlier Detection: IQR Plot & Box Plot)





3.3 Feature Scaling

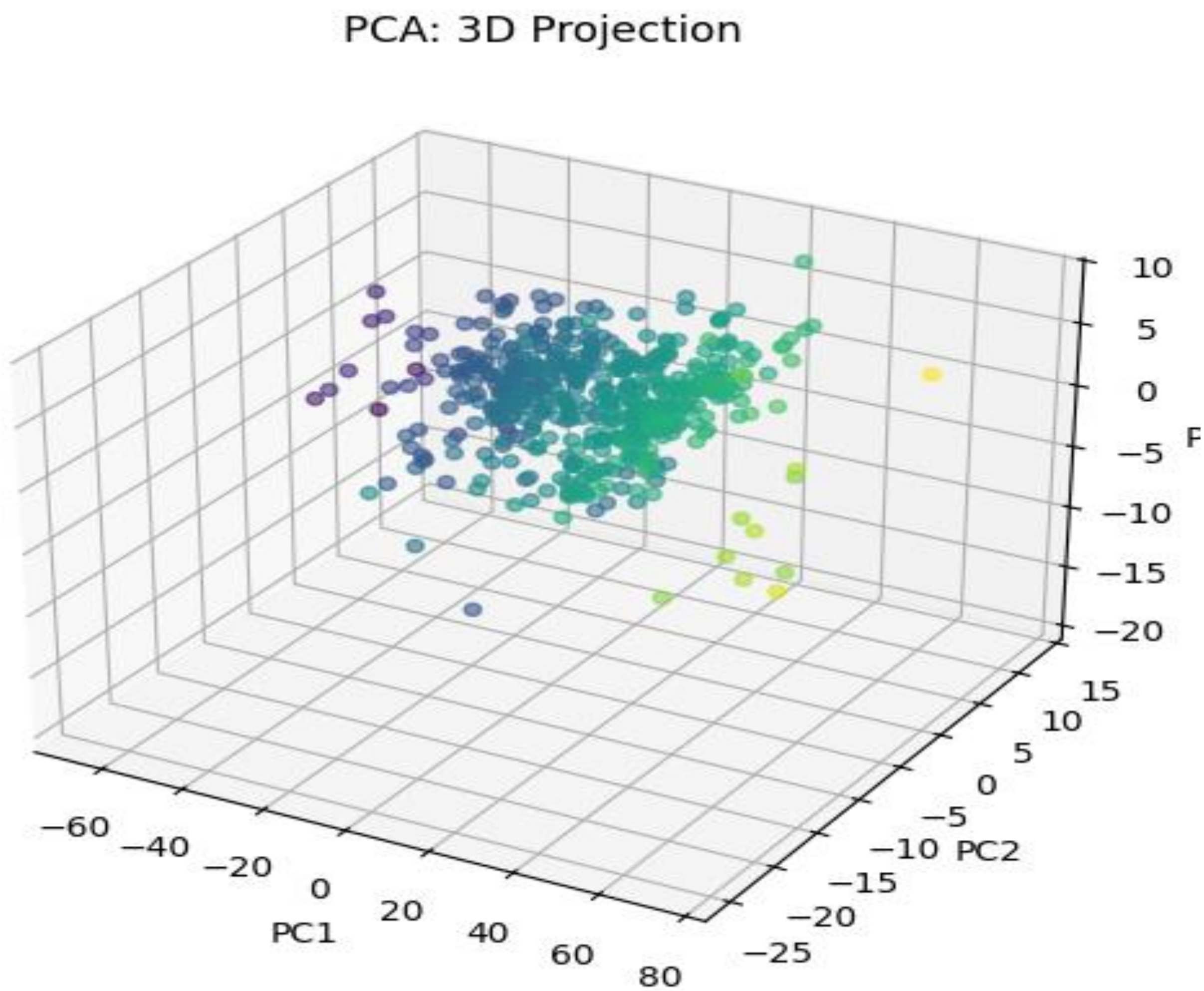
- Applied **Robust Scaling** to make the data **resilient to outliers.**

3.4 Dimensionality Reduction (PCA)

- Applied **PCA** to reduce **448 features → 3 principal components**, retaining **95% variance**.
- PCA helped **reduce overfitting, speed up training, and improve model interpretability.**

Visualization:

(PCA 3D Plot to Show Feature Reduction)



4. Model Selection, Training & Evaluation

4.1 Machine Learning Models Evaluated

Three models were trained for regression:

- **Random Forest**
- **XGBoost**
- **1D CNN (Convolutional Neural Network)**

4.2 Model Architectures & Hyperparameters

Random Forest

- *n_estimators=500, max_depth=20, min_samples_split=5, max_features=0.7*
- **Strengths:** Performs well on structured data, less sensitive to noise.

XGBoost

- *n_estimators=500, learning_rate=0.03, max_depth=9, colsample_bytree=0.8*
- **Strengths:** Handles non-linearity well and optimizes computational efficiency.

CNN (Deep Learning Approach)

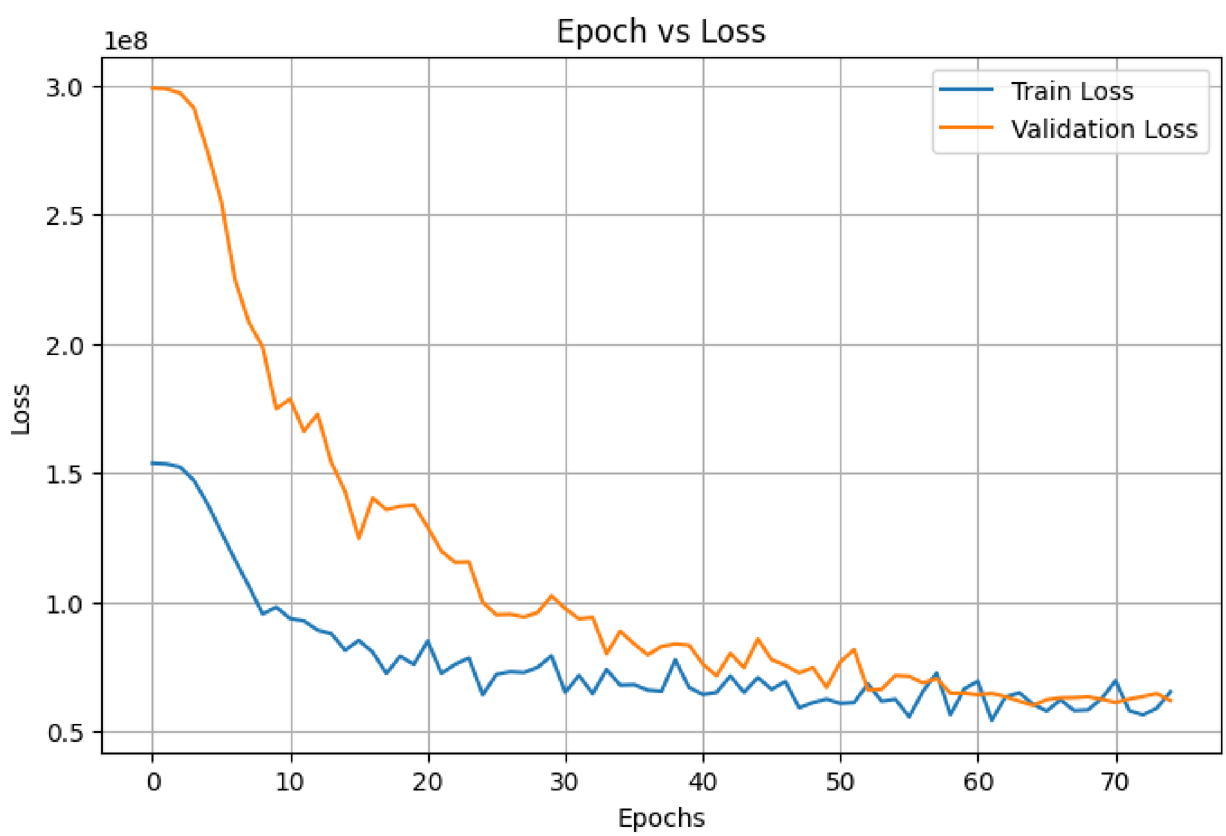
Layers:

- Conv1D layers (256, 128, 64 filters) + ReLU Activation
- Batch Normalization + Dropout for regularization
- Fully Connected Dense Layers (256, 128, 64)
- Adam Optimizer (learning_rate=0.0005), MSE Loss

Issue with CNN:

- **Overfitting observed** when adding extra layers.
- Can be improved with **hyperparameter tuning and regularization techniques.**

Visualization :
(CNN Training: Epoch vs Loss Plot)



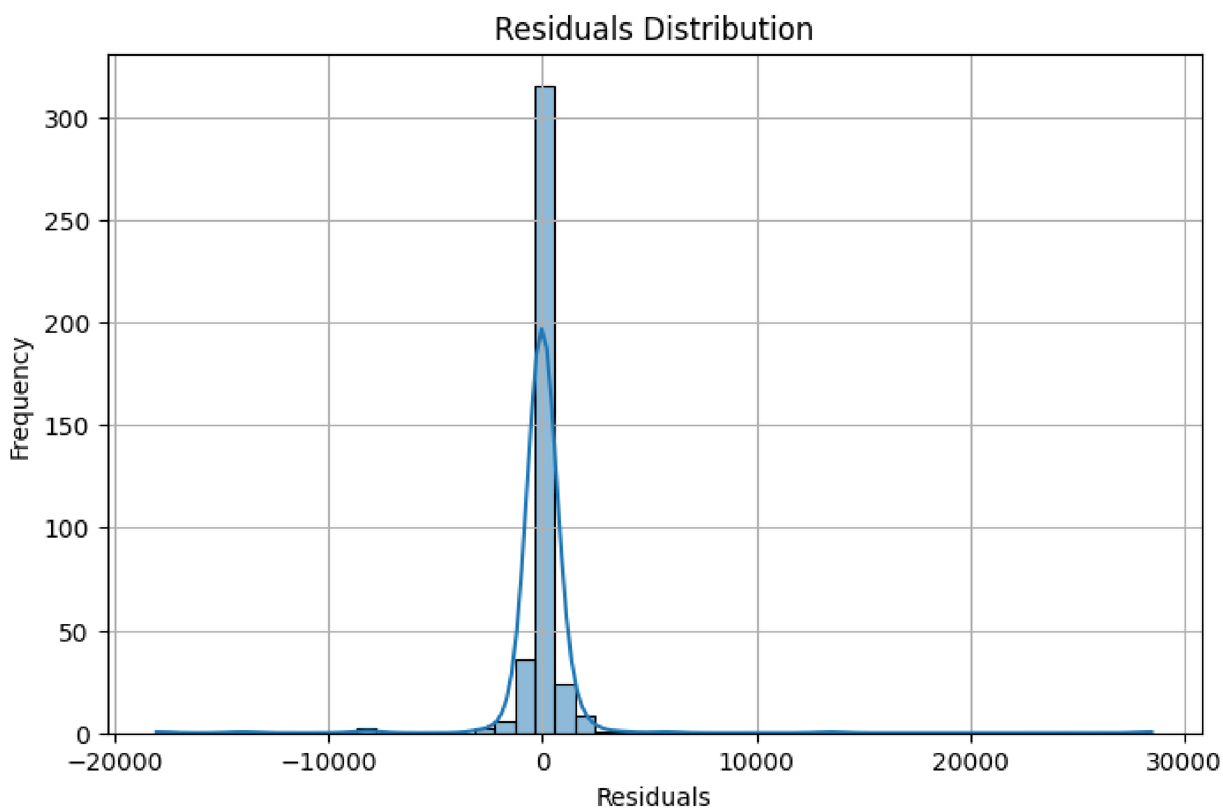
5. **Model Performance & Evaluation**

Model	MSE (↓ Better)	MAE (↓ Better)	R^2 Score (↑ Better)
Random Forest	14,406,387.25	2012.74	0.9485
XGBoost	42,026,314.11	2279.86	0.8497
CNN	48,054,924.62	2555.20	0.8281

Key Findings:

- **Random Forest performed the best** (lowest MSE, highest R^2 = **0.9485**).
- **XGBoost performed decently**, but not as well as Random Forest.
- **CNN had the lowest performance**, likely due to the small number of input features after PCA

Visualization:
(Residual Plot to Check Model Fit)



6. Key Insights & Recommendations

Key Findings:

- **Random Forest is the best-performing model** for this dataset.
- **PCA effectively reduced features from 448 → 3**, improving efficiency.
- **CNN was overfitting**, but could be improved with better tuning.
- **Outliers were present**, but removing them worsened performance, so they were retained.

Suggestions for Improvement:

- **Hyperparameter tuning for CNN** could improve accuracy.
- **Testing LSTMs or hybrid models (CNN + XGBoost)** may improve deep learning performance.
- **Feature engineering (instead of PCA)** could yield better results.
- **Stacking models (RF + XGBoost)** may further improve accuracy.

7. Deployment: Streamlit App for Frontend

A **Streamlit app** was developed to visualize model predictions interactively.

- Users can **upload spectral reflectance data** and **get DON concentration predictions**.
- **Plots** for Training Losses and Different Models Comaprisons are included.

Next Steps:

- Deploy the **Streamlit app** for real-time model evaluation.
- Allow **interactive hyperparameter tuning** to further refine CNN performance.

Final Recommendation: Use Random Forest for Mycotoxin Prediction

- ✓ Best accuracy ($R^2 = 0.9485$)
- ✓ Handles high-dimensional data well
- ✓ Resilient to noise and outliers

Future work: Tune CNN parameters, try feature engineering, and deploy the Streamlit app for real-time use!