DOCUMENTATION ON MODEL CREATION FOR PREDICTION OF MYCOTOXIN LEVELS

CREATED BY: AVINASH SOY

Creator Email: avisoy2002@gmail.com

Creator Contact No: 9752041517

1. INTRODUCTION

This report details the analysis of hyperspectral imaging data to predict mycotoxin levels (DON concentration) in corn samples using machine learning techniques. The primary objective was to preprocess the data, reduce dimensionality, and develop an optimized regression model using XGBoost, Neural Network, Decision Tree and Gradient boosting Regression.

The primary objectives of this report are to:

1. **Analyse Historical Data**: Evaluate past data trends and patterns to understand the underlying structure of the time series.
2. **Develop Forecasting Models**: Build and validate models for next-day and next hour ahead predictions using advanced statistical and machine learning techniques.
3. **Evaluate Model Performance**: Assess the accuracy and reliability of the forecasting models through various performance metrics.
4. **Provide Insights and Recommendations:** Offer actionable insights based on the forecasted data and suggest improvements for future forecasting efforts.

2. WORKFLOW

Here's the workflow of the XGBoost Regression model:

1. **Input the CSV File**: First task is to take the CSV File, which consists of Features of corn samples and the values of mycotoxin levels based on the features.
   The dataset contains 450 columns containing 448 features, hsi_id, vomitoxin_ppb(mycotoxin levels).
2. **Data Inspection and Cleaning**: Check the missing values and outliers present in the dataset. We will handle missing values by imputing the medians. Standardized spectral features using **StandardScaler** for improved model performance.
3. **Dimensionality Reduction**: We will apply Principal Component Analysis(PCA) to reduce the features space while retaining critical information. We can retain the **top n principal components**, which preserved a significant variance of the data. N varies due to different model architecture.
4. **Model Development**:
   XGBoost Regression: It selects XGBoost as Regression model due to it's robustness and ability to handle high dimensional data.
   Hyparameters tuning included:
   n_estimators=500 (no.of trees)
   Learning_rate = 0.05 (controls step size in updates).
   Max_depth = 7 (prevents overfitting)
   Subsamples=0.8 and colsample_bytree = 0.8 (reduce overfitting)
   Random_state = 42 (ensures reproducibility).
5. **Model Training and Testing**: We have used 80% for training and 20% for testing, then we will implement 20 rounds of patience based on Root Mean Square Error and we will use root mean square error to track training progress and prevent overfitting.

6. **Model Evaluation**: The trained model was evaluated using the following metrics:
   Mean Absolute Error (MAE): Measures the average absolute differences between actual and predicted values.
   Root Mean Squared Error (RMSE): Penalizes larger errors more than MAE.
   R² Score: Indicates how well the model explains variance in the data.
7. **Results**:
   Mean Absolute Error: 2046.098
   Root Mean Square Error: 4709.232
   R^2 score: 0.92066
8. **Visualizations**:
   Scatter plot is used for displaying actual vs predicted values of mycotoxin levels, showing a positive correlation. Comparison chart is used for enabling detailed visualization of actual and predicted trends.
9. **Conclusion**:
   - The XGBoost model provided a high R² score, indicating strong predictive capabilities.
   - PCA effectively reduced the feature space without significant information loss.
   - Early stopping and hyperparameter tuning helped optimize the model's performance

Model Development of the Neural Network Models:

We've selected LSTM as regression model due to its capability to capture temporal data as well as long term dependencies in hyperspectral data. Hyperparameters tuning included:

- **PCA**(n_components=30) to reduce dimensionality
- Reshaped data into a 3D format for LSTM processing
- Defined a deep neural network with multiple hidden layers: 128, 64, 32, 16, 8, 4, and 2 neurons in successive layers
- **ReLU** activation for non-linearity
- **Dropout (0.15)** applied after each layer to prevent overfitting
- **Adam optimizer** with a learning rate of 0.005 for efficient gradient updates
- Implemented **early stopping** (patience = 45) to avoid unnecessary training
  For successive steps, we use the same parameters for train-test split, evaluation metrics.
- **Results**:
  Mean Absolute Error: 2956.790
  Root Mean Square Error: 8398.110
  R^2 score: 0.747692
- **Conclusion**:
  1. The optimized LSTM model demonstrated improved predictive performance.
  2. PCA effectively reduced the feature space without significant information loss.

3. Early stopping and hyperparameter tuning helped optimize the model's performance.

Model Development of Gradient Boosting Regression:

I've selected **Gradient Boosting Regressor (GBR)** as the regression model due to its ability to capture complex relationships in the data.

Hyperparameter tuning was performed using RandomizedSearchCV, searching over:

- n_estimators: [200, 300, 400, 500]
- learning_rate: [0.01, 0.05, 0.1, 0.2]
- max_depth: [3, 5, 7, 9]
- min_samples_split: [2, 5, 10]
- min_samples_leaf: [1, 2, 4]
- subsample: [0.8, 0.9, 1.0]

The best model was selected based on cross-validation performance.

Train-Test Split: 80% training and 20% testing.

Evaluation Metric: Used mean_squared_error (MSE), mean_absolute_error (MAE), and r2_score to measure performance.

Randomized Search Optimization: Used 30 iterations of randomized search for optimal hyperparameter selection.

Result:

Mean Absolute Error (MAE): 1565.780

Root Mean Squared Error (RMSE): 3217.378

R^2 Score: 0.9629

Model Development of Decision Tree Regression:

I've selected Decision Tree Regressor as the regression model due to its ability to handle non-linear relationships in hyperspectral data. Hyperparameters tuning included:

- **PCA**(n_components=20) to reduce dimensionality
- **Grid Search** Cross-Validation to optimize hyperparameters:
- max_depth: [5, 10, 15, 20, None]
- min_samples_split: [2, 5, 10]
- min_samples_leaf: [1, 2, 4, 8, 16]

- **Train-Test Split:** 80% training and 20% testing.

- Best Model Selection: Identified using GridSearchCV with 5-fold cross-validation.
- Grid Search Cross-Validation: Performed to find the optimal hyperparameters.
- Evaluation Metric: Used mse for loss function and mae as an additional metric.
- Conclusion:
  1. The optimized Decision Tree Regressor demonstrated improved predictive performance.
  2. PCA effectively reduced the feature space without significant information loss.
  3. Hyperparameter tuning with Grid Search helped optimize the model's performance.
- Results:

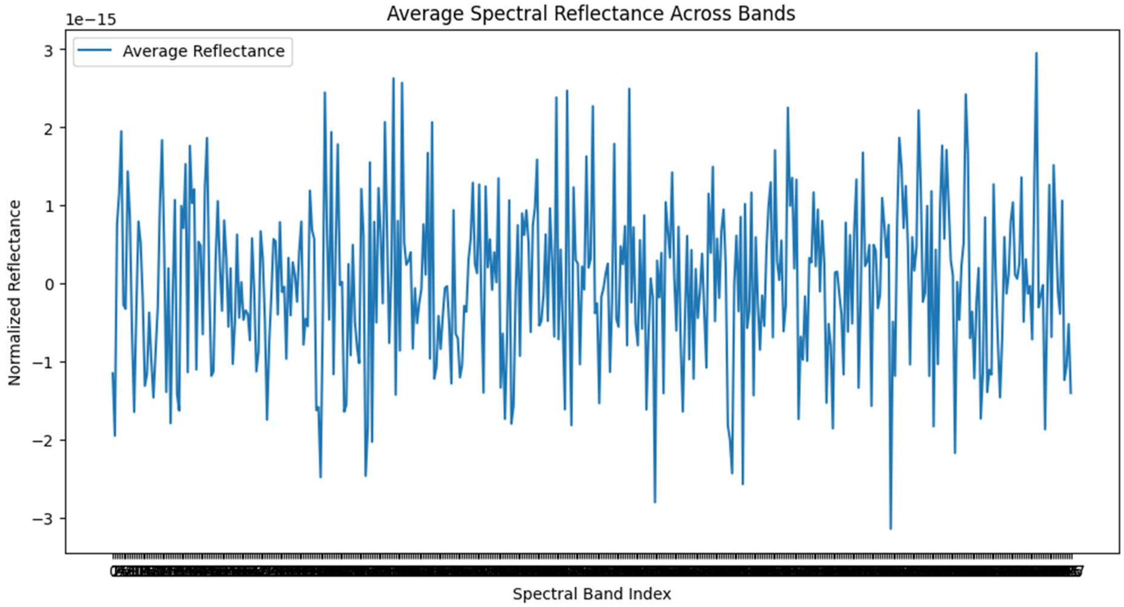
  Mean Absolute Error (MAE): 1984.67344

  Root Mean Squared Error (RMSE): 4013.03536

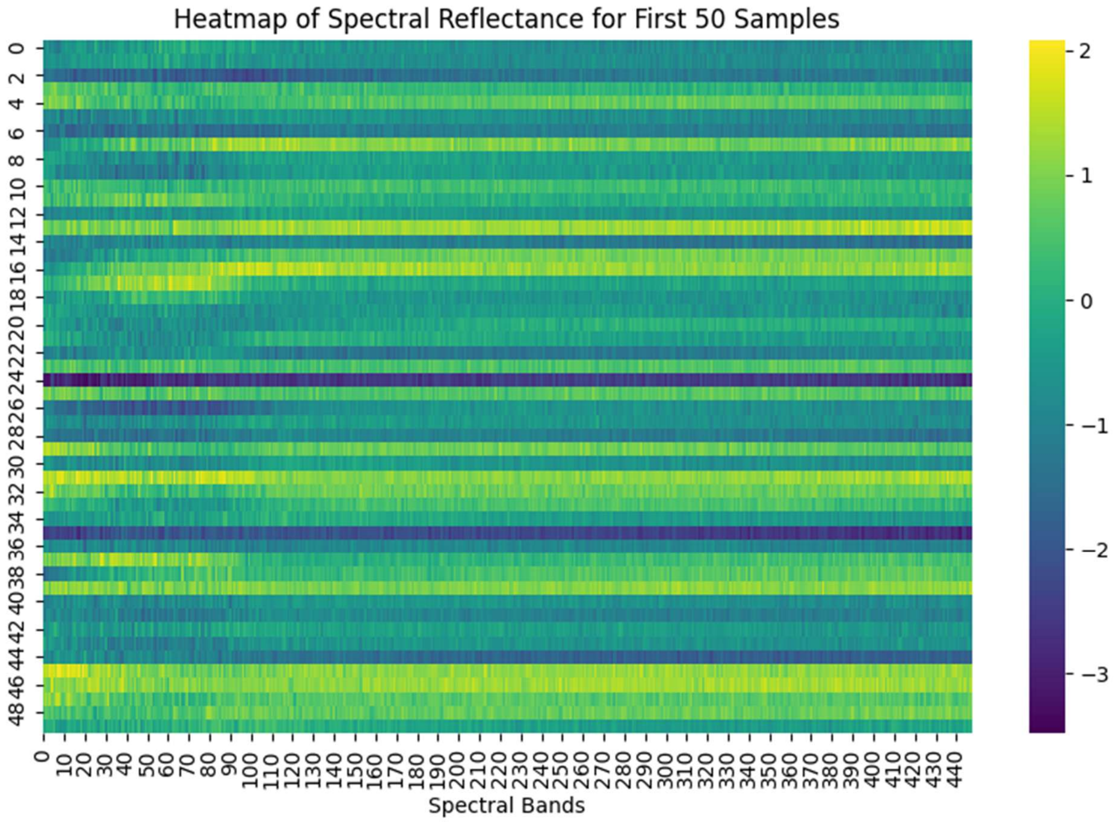
  R² Score: 0.94197

3. RESULTS

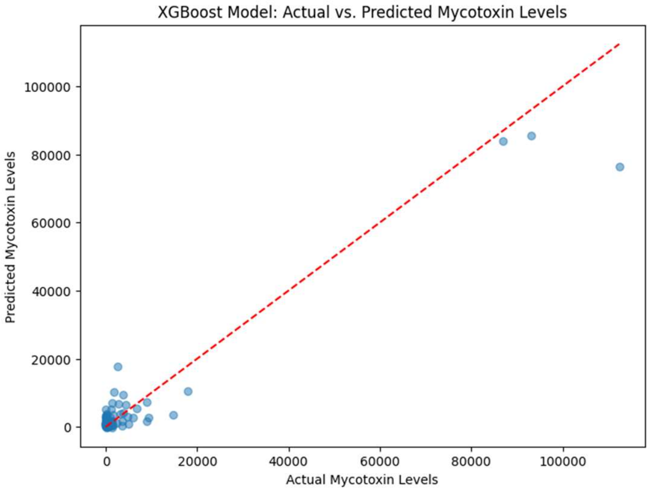
The below figure shows the average spectral reflectance across the bands w.r.t Spectral Band Index. The y-axis shows normalized reflectance.

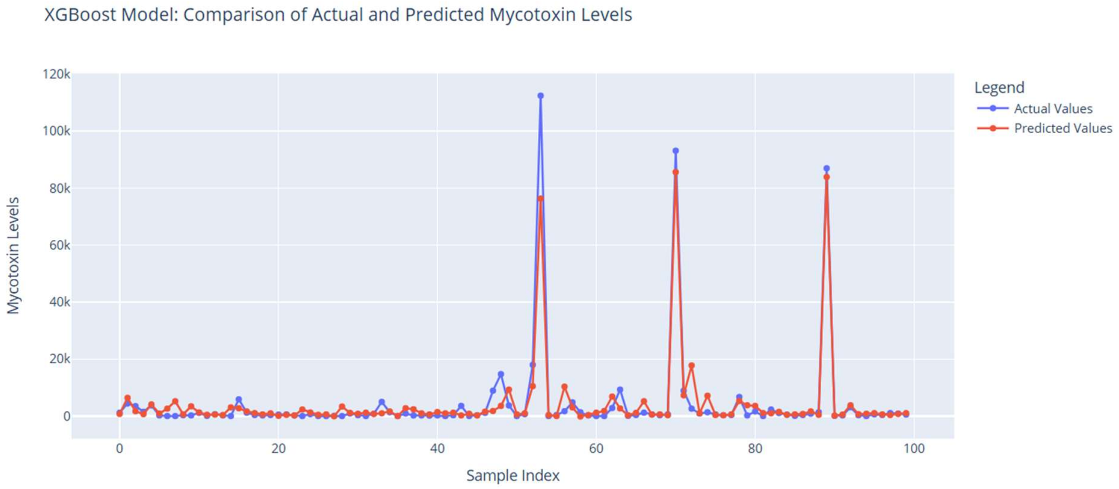


The below Figure shows the heatmap visualization of first 50 corn sample's Spectral reflectance.

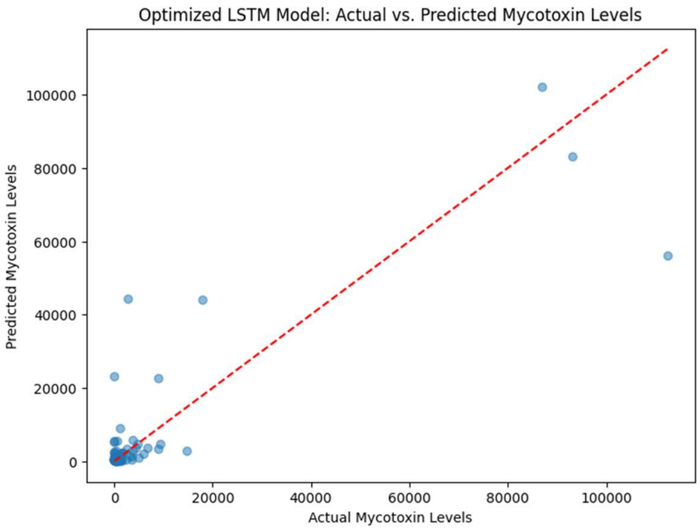


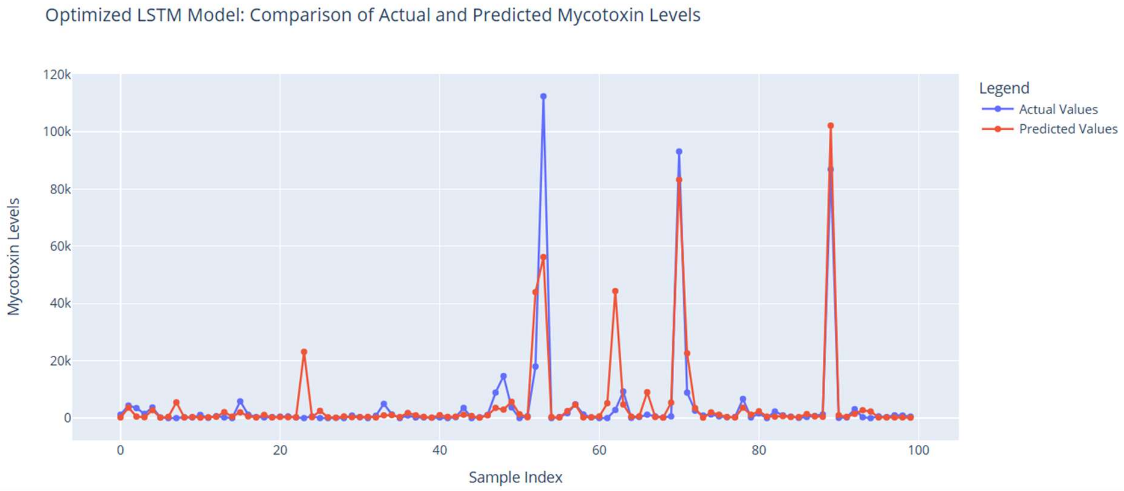
Here's the result developed by the models using the concept of XGBoost Regression Model.



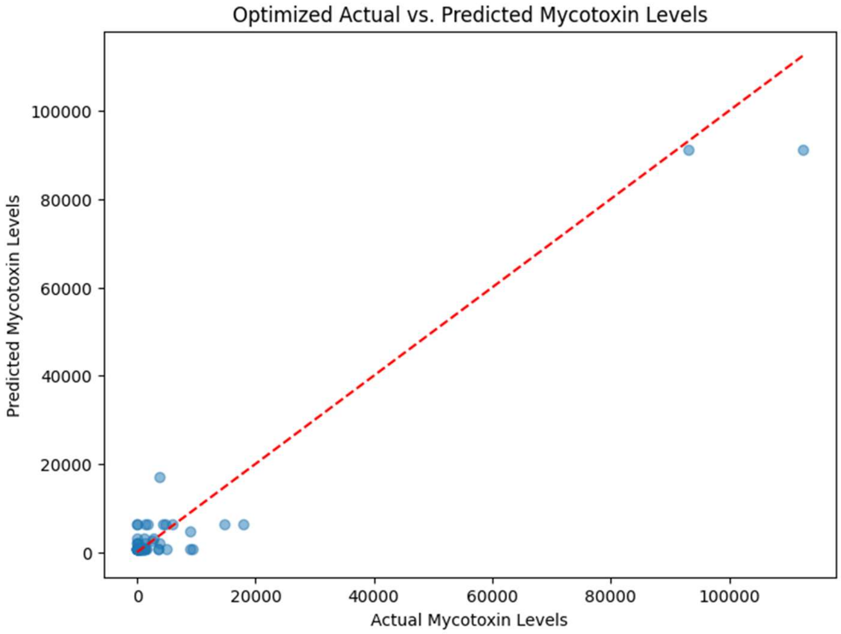

XGBoost Model: Comparison of Actual and Predicted Mycotoxin Levels

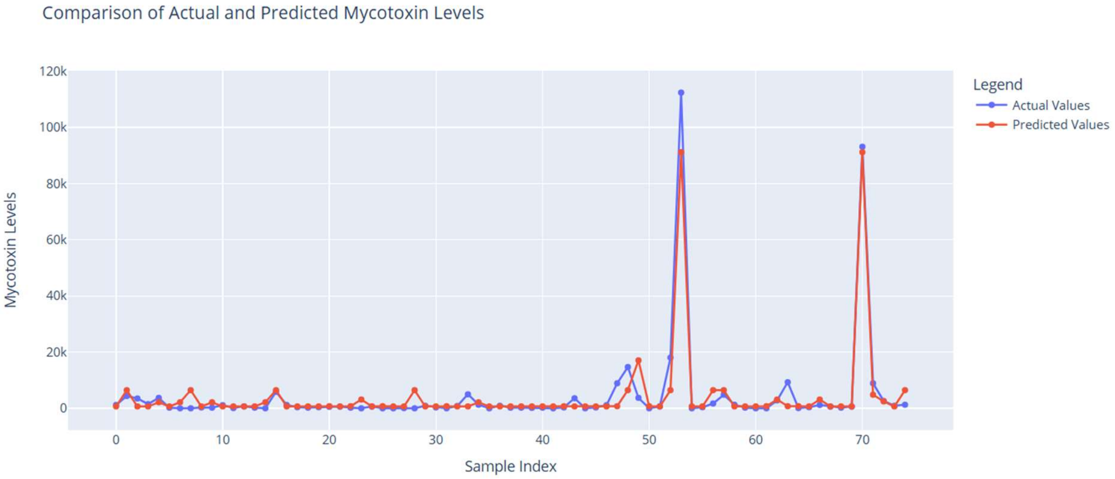
Here’s the result developed by the models using the concept of LSTM Models


Optimized LSTM Model: Actual vs. Predicted Mycotoxin Levels

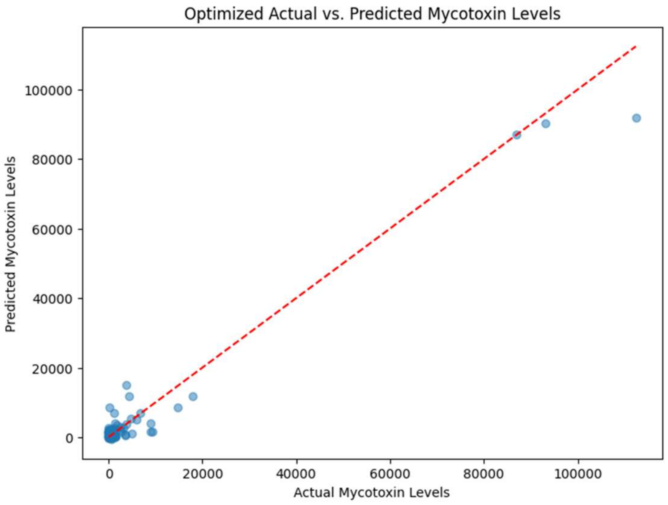

Optimized LSTM Model: Comparison of Actual and Predicted Mycotoxin Levels

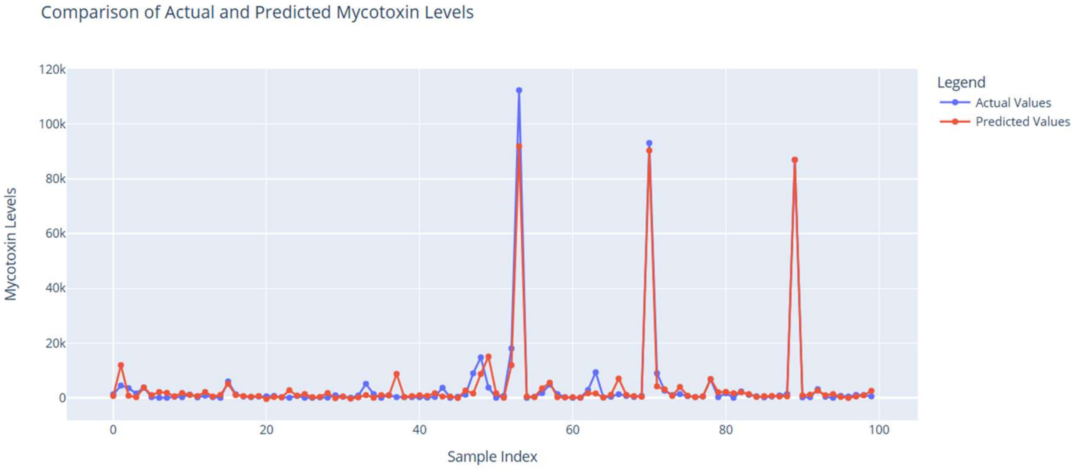
Here’s the result developed by the models using the concept of Decision Tree Regression Model:





Here’s the result developed by the models using the concept of Gradient Boosting Regression Model:


Optimized Actual vs. Predicted Mycotoxin Levels
Predicted Mycotoxin Levels
100000
80000
60000
40000
20000
0
0
20000
40000
60000
80000
100000
Actual Mycotoxin Levels


Comparison of Actual and Predicted Mycotoxin Levels
Mycotoxin Levels
120k
100k
80k
60k
40k
20k
0
0
20
40
60
80
100
Sample Index
Legend
Actual Values
Predicted Values