Predicting mycotoxin levels in corn samples.

1. Preprocessing Steps and Rationale

1.1 Data Loading and Exploration

- The dataset was loaded using pandas and inspected for missing values and basic structure.
- The features were identified, and the target variable (vomitoxin_ppb) was separated.
- hsi_id was removed as it is an identifier and does not contribute to prediction.

1.2 Feature Scaling

- **Standardization** (StandardScaler): Standardization was applied to the input features to ensure they have a mean of 0 and a standard deviation of 1.
    - **Rationale**: Many machine learning models, especially distance-based models and gradient-based optimizers, perform better when input data is standardized.

1.3 Dimensionality Reduction (PCA)

- **Principal Component Analysis (PCA)** was applied to reduce the number of features to 20 principal components.
    - **Rationale**: PCA helps remove redundant information, reduces noise, and speeds up training by reducing the number of features.
    - **Trade-off**: Some variance is lost, which may slightly impact model performance.

1.4 Data Splitting

- The dataset was split into **80% training and 20% testing**.

---

2. Insights from Dimensionality Reduction (PCA)

- PCA transformed the original feature space into orthogonal components.
- The first few components captured most of the variance in the data.
- This reduced the risk of overfitting by eliminating less important variations.

---

3. Model Selection, Training, and Evaluation

3.1 Random Forest and XGBoost

Model Justification

- **Random Forest**: A robust ensemble-based method that reduces overfitting and handles complex relationships in data.
- **XGBoost**: A gradient boosting model that typically outperforms Random Forest in structured data due to its ability to minimize loss efficiently.

Training

- Both models were trained using default hyperparameters initially.
- **Grid Search Optimization** was performed for XGBoost to fine-tune hyperparameters.

Evaluation Metrics

- **Mean Absolute Error (MAE)**: Measures the average absolute error.
- **Root Mean Squared Error (RMSE)**: Penalizes large errors more than MAE.
- **R^2 Score**: Measures how well the model explains the variance in the data.

Results

Model	MAE	RMSE	R^2 Score
Random Forest	2674.7867	7119.3581	0.8187
XGBoost	1729.7213	3395.7421	0.9587
Tuned XGBoost	1738.6320	3422.2255	0.9581

- **XGBoost outperformed Random Forest**

---

4. Transformer Model Performance

4.1 Model Justification

- **Transformers** have shown great success in sequential data and feature-rich structured datasets.
- Unlike XGBoost, **Transformers can learn complex dependencies** between features.

4.2 Transformer Model Architecture

- Input was passed through a series of **self-attention layers**, which capture relationships between features.
- The output was processed through dense layers for regression.

5. Key Findings and Model Comparison

Model	MAE	RMSE	R^2 Score
XGBoost	1729.7213	3395.7421	0.9587
Tuned XGBoost	1738.6320	3422.2255	0.9581
Transformer	4412.3582	17291.6046	-0.0696

5.1 Best Performing Model

- **XGBoost achieved the best R^2 score (0.95)**, showing its strength in capturing complex feature interactions.

XGBoost is designed for structured tabular data and works well on this dataset because:
Handles missing values efficiently.
Captures complex feature interactions.
Less sensitive to data size than deep learning models.
Hyperparameter tuning improved performance.

5.2 Why Transformer Gave Negative R^2?

Negative R^2 means the model performed **worse than a simple mean predictor**.
The reasons:

Challenges with Transformer in Regression:

1. **Transformers excel at sequential data** (e.g., NLP, time series), but DON concentration data **lacks sequential dependencies**.
2. **Data was not large enough** for Transformers to generalize well.
3. **Overfitting issue:**
    - Transformers have millions of parameters, and the dataset **was too small** to train efficiently.
    - The model memorized training data but failed on the test set.

Submission by:

NavyaSri Vallakati