**Machine Learning Model Documentation**

**1. Introduction**

This document provides an overview of the machine learning model, including data preprocessing, model training, evaluation, and API integration.

**2. Data Preprocessing**

**2.1 Loading Data**

* Data is loaded from a CSV file named MLE-Assignment.csv.
* The dataset consists of hsi\_id, spectral features, and the target variable vomitoxin\_ppb.

**2.2 Handling Outliers**

* Outliers in vomitoxin\_ppb are removed using the **best outlier detection method**.
* Various techniques such as IQR, Z-score, and Isolation Forest were considered.
* The chosen method effectively reduces skewness while retaining meaningful data.

**2.3 Log Transformation**

* A log transformation (log1p) is applied to vomitoxin\_ppb to reduce skewness and improve model performance.

**3. Model Training**

**3.1 Feature Engineering**

* Spectral features are selected for training.
* Data is split into training and test sets using an 80-20 split.

**3.2 Model Selection**

* Various regression models were evaluated, including:
  + Linear Regression
  + Decision Tree Regressor
  + Random Forest Regressor
  + XGBoost Regressor
* The model with the best performance is chosen based on RMSE, R², and MAE metrics.

**3.3 Model Evaluation**

* Performance metrics:
  + **R² Score**: Measures the proportion of variance explained.
  + **RMSE**: Root Mean Squared Error for assessing prediction errors.
  + **MAE**: Mean Absolute Error for evaluating absolute differences.

**4. API Integration**

**4.1 FastAPI Implementation**

* An API is developed using **FastAPI** to serve the trained model.
* The API provides an endpoint to predict vomitoxin\_ppb from spectral data.
* Input validation is performed to ensure proper request handling.

**4.2 API Endpoints**

* POST /predict: Accepts JSON input and returns the predicted vomitoxin\_ppb.
* GET /health: Checks if the API is running correctly.

**5. Unit Testing**

* Unit tests are written to ensure model accuracy and API reliability.
* Testing framework: **pytest**.
* Tests include:
  + Checking API response structure.
  + Validating model predictions against test data.
  + Ensuring input validation handles incorrect inputs gracefully.

**6. Deployment**

* The FastAPI application is containerized for easy deployment.

**7. Conclusion**

This documentation outlines the steps taken to preprocess data, train a regression model, integrate it into an API, and deploy the solution. The pipeline ensures efficient and accurate predictions for vomitoxin\_ppb in corn samples.