# Machine Learning Task Report

## 1. Data Exploration and Preprocessing

- **Data Loading:** The dataset was loaded using Pandas read_csv function.
- **ID Column Removal:** The hsi_id column, which is non-informative for prediction, was dropped.
- **Missing Values:** Checked for missing values; if any were found, they would be handled appropriately.
- **Outlier Detection:** Visualized using boxplots to identify potential anomalies in spectral reflectance data.
- **Feature Scaling:** Standardized using StandardScaler to normalize feature distributions, improving model performance.

## 2. Dimensionality Reduction

- **Technique Used:** Principal Component Analysis (PCA).
- **Why PCA?** PCA reduces the dimensionality of the dataset while retaining the most important features (i.e., those with the highest variance). This helps avoid the "curse of dimensionality" and reduces computational complexity.
- **Results:** A significant proportion of the data's variance was captured with fewer components, enabling more efficient model training.

## 3. Model Training and Evaluation

- **Models Trained:**
  - **Random Forest Regressor:** A robust ensemble method that reduces overfitting through bagging.
  - **Neural Network (Keras):** A multi-layer perceptron trained to learn complex patterns in the data.
- **Training Process:** The dataset was split into training and test sets using train_test_split. Models were trained and evaluated on both sets.
- **Evaluation Metrics:**
  - **Mean Absolute Error (MAE):** Measures the average magnitude of errors.
  - **Mean Squared Error (MSE):** Penalizes larger errors more heavily.
  - **$R^2$ Score:** Indicates how well the model explains the variance in the target variable.

## 4. Key Findings and Suggestions for Improvement

- **Model Performance:** The Neural Network model performed better overall, achieving lower MAE and higher $R^2$ compared to the Random Forest.
- **Insights:** PCA significantly reduced computation time without major accuracy loss, confirming the dataset had redundant features.
- **Recommendations:**
  - Fine-tune hyperparameters (e.g., number of trees, learning rate) for improved accuracy.
  - Experiment with other dimensionality reduction techniques like t-SNE or UMAP.
  - Handle outliers more rigorously using statistical methods like IQR filtering.
  - Try ensemble techniques like stacking or boosting for potentially better results.

## Conclusion

This project showcased a complete ML workflow: from data preprocessing and dimensionality reduction to model training and evaluation. The insights gained from PCA and model comparisons provide a solid foundation for further optimization and feature engineering.