1. Introduction

This report focuses on exploring and preprocessing data, applying dimensionality reduction techniques, training a machine learning model for regression, and evaluating its performance. The dataset used in this study contains spectral reflectance data across different wavelengths, with a target variable representing some physical property.

2. Data Exploration and Preprocessing

The initial step in any data analysis pipeline is data exploration and preprocessing. The dataset was loaded and we used df.info and df.describe for a summary of the dataset with a statistics overview.

2.1 Missing Values Inspection

We began by checking for missing values in the dataset, using data.isnull().sum() to determine whether any features were incomplete. This step is essential to ensure data integrity before applying any models. In this case, no missing values were found.

2.2 Outliers and Data Distribution

To detect outliers, we have to find the quartiles and then through the upper and lower bound, we have removed all the outliers. After removing, we print the actual data and the data we get after removing outliers. Through Boxplot, the distribution of the dataset helps to detect outliers.

2.3 Normalization

Normalization was applied to the spectral data to ensure that all features (wavelengths) have the same scale. This was done using StandardScaler from scikit-learn. This step is crucial in machine learning models, particularly those sensitive to the magnitude of the input data, such as neural networks and distance-based models.

2.4 Visualizations

Two key visualizations were created:

1. **Average Reflectance Plot**: This plot shows the average spectral reflectance across all samples for each wavelength band. The curve provides insight into the overall reflectance behaviour, helping us understand the spectral signature of the dataset.
2. **Correlation Heatmap**: A heatmap was generated to assess the correlation between different spectral bands. This helps identify potential redundancies in the data, which might inform feature selection or dimensionality reduction strategies.

3. Dimensionality Reduction

Given the high-dimensional nature of spectral data, dimensionality reduction techniques were applied to reduce the number of features and visualize the data in lower dimensions. Two techniques were explored: Principal Component Analysis (PCA) and t-SNE. We have used the PCA method for reduction.

Principal Component Analysis (PCA)

PCA is a linear dimensionality reduction technique that transforms the data into a set of orthogonal components. We applied PCA to the standardized spectral data, reducing it to two principal components (PC1 and PC2) for visualization.

- **Variance Explained by PCA**: The first two principal components explained approximately 80% of the variance in the data, suggesting that the majority of the information in the dataset can be captured with just these two components.
- **3DScatter Plot**: A 3D scatter plot of the first three principal components was created to visually examine the data. This revealed some clustering of data points, which could represent different classes or groups within the data. The 3D visualization helped identify more complex structures and relationships that might not be evident in a 2D plot.

4. Model Training

In the next phase, a machine learning model was selected and trained to predict the target variable. A deep learning model, specifically a neural network, was chosen for its ability to model complex non-linear relationships in the data.

4.1 Train-Test Split

The dataset was split into training (80%) and testing (20%) sets using train_test_split from scikit-learn. This split ensures that the model is trained on one portion of the data and evaluated on unseen data to assess its generalization ability.

4.3 Model Training

The neural network was trained using the Adam optimizer and mean squared error (MSE) loss function. Hyperparameter optimization was performed using grid search, although in this case, default parameters were sufficient for obtaining reasonable performance.

4.4 Model Evaluation

The model's performance was evaluated using common regression metrics:

- **Mean Absolute Error (MAE)**: Measures the average magnitude of errors in predictions.
- **Root Mean Squared Error (RMSE)**: Highlights large errors more than MAE, as it squares the errors.
- **R^2 Score**: Represents the proportion of the variance in the target variable that is predictable from the features.

4.5 Evaluation Visualizations

To visualize the model's predictions:

1. **Scatter Plot (Actual vs. Predicted)**: A scatter plot was generated to compare the actual target values versus the predicted values. This visualization helps identify if the model is underfitting or overfitting.
2. **Residual Plot**: A plot of residuals (the differences between actual and predicted values) was generated to check if the model exhibits any systematic errors.