Hyperspectral Imaging Analysis Report

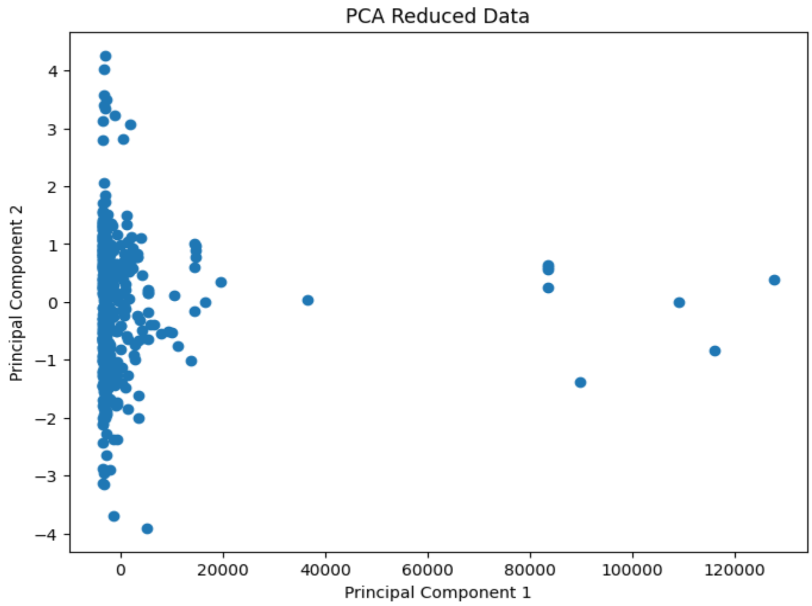
1. Preprocessing Steps and Rationale

- **Data Loading**:
  - Action: The dataset was loaded from a CSV file.
  - Rationale: To access and manipulate the data using Pandas.
- **Inspect Class Distribution**:
  - Action: checked how many samples each category had.
  - Rationale: To identify any categories with too few samples, which could cause problems during splitting.
- **Scale the Labels**:
  - Action: Used StandardScaler to scales the labels.
  - Rationale: Prevent any feature or label from dominating others due to scale differences.
- **Remove Sparse Classes**:
  - Action: Categories with less than 2 samples were removed.
  - Rationale: To ensure that each category has enough samples to allow proper data splitting.
- **Remove 'hsi_id' Column**:
  - Action: The hsi_id column was dropped.
  - Rationale: This column is an identifier and not a predictive feature.
- **Data Splitting**:
  - Action: The dataset was split into training (99%) and testing (1%) sets.
  - Rationale: To evaluate the model's ability to generalize to unseen data. Due to the very small dataset, a manual split was used to ensure each category is represented in both sets.
- **Data Reshaping**:
  - Action: The data was reshaped to fit the CNN input requirements.
  - Rationale: CNN models require input data to have a specific shape (samples, features, channels).

- **Data Type Conversion**:
  - Action: The data was converted to float32.
  - Rationale: To ensure compatibility with TensorFlow and prevent data type-related errors during model training.

2. Insights from Dimensionality Reduction

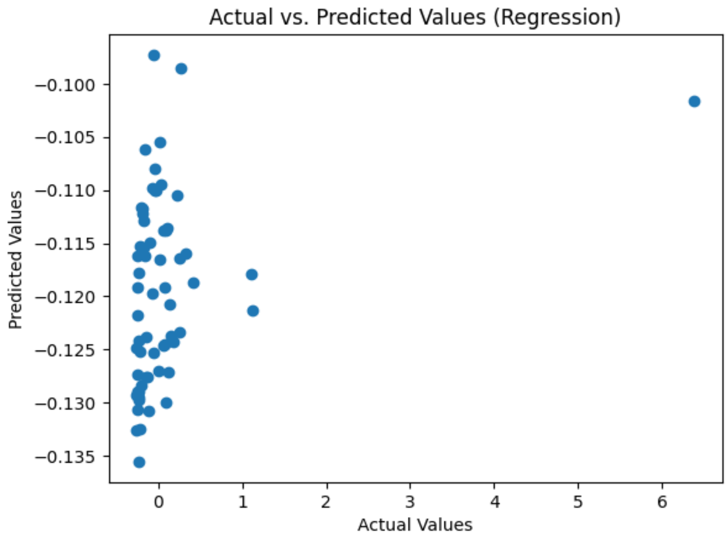
- **Action**: Reduced feature dimensions of a dataset using PCA and t-SNE.
- **Rationale**: Simplify the model and accelerate training by using fewer features.
- **Variance Reporting**: PCA was used and report the amount of variance explained by each component.



3. Model Selection, Training, and Evaluation Details

- **Model Selection**:
  - Action: A 1D Convolutional Neural Network (CNN) was chosen.
  - Rationale: CNNs are effective at extracting features from sequential data and have been shown to perform well on spectral data.
- **Model Architecture**:
  - Action: The CNN includes convolutional, pooling, flattening, dense, and dropout layers.
  - Rationale: This architecture is designed to capture relevant patterns in the spectral data while preventing overfitting.

- **Training**:
    - Action: The model was trained using the Adam optimizer and mean squared error (MSE) loss.
    - Rationale: Adam is an efficient optimization algorithm, and MSE is suitable for regression tasks. The model was trained for 10 epochs.
- **Evaluation**:
    - Action: The model was evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 Score.
    - Rationale: These metrics provide a comprehensive assessment of the model's regression performance.



4. Key Findings and Suggestions for Improvement

- **Key Findings**:
    - A CNN model can be trained on hyperspectral data for regression tasks.
    - Data preprocessing is critical for addressing data limitations and ensuring model compatibility.
- **Suggestions for Improvement**:
    - **Increase Dataset Size**: A larger dataset would significantly improve the model's ability to generalize.
    - **Implement Dimensionality Reduction**: Employ PCA or t-SNE to reduce the number of spectral bands, which may enhance model performance and reduce computational costs.

- **Hyperparameter Tuning**: Perform hyperparameter optimization using techniques such as Grid Search or Random Search to find the best model configuration.
- **Explore Alternative Models**: Investigate other models such as LSTMs or GNNs, which may be better suited to capturing the underlying patterns in the data.
- **Address Data Imbalance**: Implement techniques such as oversampling or class weighting to mitigate the impact of imbalanced data on model training.