**Human Activity Recognition Using Machine Learning and Neural Networks**

This project focuses on **Human Activity Recognition (HAR)** using smartphone sensor data. The objective is to classify human activities into distinct categories by leveraging machine learning and deep learning techniques. The dataset used for this project is sourced from publicly available repositories.

**Environment Setup and Library Imports**:

* Libraries for machine learning, neural networks (Keras/TensorFlow), visualization (Matplotlib, Seaborn), and metrics are imported.
* TensorFlow and Keras-related components are loaded for building and training deep learning models.

**Data Loading**:

* Training and test datasets are loaded from a given URL.

**Data Inspection and Preprocessing**:

* The script inspects for missing values, duplicates, and column inconsistencies.
* Column names are standardized.
* The target variable, Activity, is label-encoded.
* Feature columns are scaled using MinMaxScaler.

**Data Splitting**:

* Features (X) and target variables (y) are separated for both training and testing.
* y is one-hot encoded for use in neural networks.

**Neural Network Model**:

* A sequential neural network with 3 layers is defined using Keras.
* It uses relu activation for hidden layers, softmax for the output layer, and Dropout layers for regularization.
* The model is compiled with Adam optimizer and categorical\_crossentropy loss.
* Training occurs for up to 50 epochs, with early stopping enabled.

**Evaluation**:

* The trained neural network is evaluated on the test set.
* Accuracy, confusion matrix, and classification reports are generated.
* Training and validation metrics (loss and accuracy) are plotted.

**Ensemble Classifiers**:

* Individual machine learning models (RandomForest, Logistic Regression, and SVM) are trained and evaluated.
* A **Voting Classifier** combines predictions from these models using soft voting.
* Performance metrics (accuracy, precision, recall, F1-score) are calculated.

**Hyperparameter Tuning**:

* Grid Search/Randomized Search is started to tune parameters for RandomForestClassifier.