Hyperparams Assignment - ML

Task 1: Hyperparameter Tuning for Each Classifier

KNN (K-Nearest Neighbors):

- **Observation**:
  - The validation curves demonstrate how the choice of 'k' (number of neighbors) impacts the model's accuracy.
  - **Sweet Spot**:
    - Too few neighbors make the model sensitive to noise.
    - Too many neighbors can oversimplify the model.
  - **Result**:
    - Testing accuracy peaks at **k = 5 to k = 10**, indicating a good balance between bias and variance.

SVM (Support Vector Machine):

- **Observation**:
  - Different kernels (linear, RBF, polynomial, sigmoid) were tested.
  - **Result**:
    - **RBF and Polynomial kernels** achieved the highest accuracy, suggesting that non-linear boundaries are more suited for this dataset.
    - **Best Performing Kernel**: The **RBF kernel** captured the complex relationships within the data more effectively.

MLP (Multilayer Perceptron):

- **Observation**:
  - Different batch sizes were tested to observe how they affect training.
  - **Result**:
    - **Small batch size**: Noisier updates, but potentially faster convergence.
    - **Large batch size**: Smoother updates, but slower convergence.
    - **Key Observation**: Accuracy is high across all batch sizes, but loss decreases **rapidly with small batch sizes** and **slowly with large batch sizes**.

Task 2: Impact of Activation Functions in MLP

- **Key Activation Functions**:
  - **ReLU**: Efficient, reduces vanishing gradients.
  - **Tanh**: Output between -1 and 1, can be helpful in some complex scenarios.
  - **Sigmoid**: Output between 0 and 1, useful for binary classification but prone to vanishing gradients.
- **Observation**:
  - Loss curves for each activation function need to be analyzed (graphs not provided here).
  - Focus on how fast the loss decreases and if it flattens too soon (indicating poor training performance).

Task 3: Experiment with Learning Rate in MLP

- **Observation**:
  - The learning rate determines the step size in gradient descent.
  - **Too High**: Can lead to overshooting, instability, and failure to converge.
  - **Too Low**: Results in slow convergence and potential to get stuck in local minima.
- **Key Observation**:
  - Identify the learning rate that provides **the fastest convergence** based on the learning curves for different learning rates (graphs not provided here).

General Observations and Insights

- **Validation Curves**: Extremely useful for visualizing how hyperparameters affect model performance.
- **Grid Search/Random Search**: Efficient strategies for exploring different hyperparameter combinations.
- **Activation Functions**:
  - 'ReLU' is generally a strong default choice, but other functions like 'tanh' and 'sigmoid' may be beneficial in certain cases.
- **Learning Rate**:
  - It's critical to properly tune the learning rate. Adaptive methods like Adam can often work well.

- **Batch Size**:
  - Smaller batch sizes add noise but can improve generalization. Larger batches tend to offer smoother updates but at the cost of slower convergence.