# Gradient Boosting Machines (GBMs) with neural network backbones

**Overview of the Hybrid Approach**

1. **Train a Gradient Boosting Machine (GBM)**:
   * The first step involves training a GBM, which is a powerful ensemble learning technique that builds multiple decision trees sequentially. Each tree aims to correct the errors made by the previous one, focusing on the instances that were previously misclassified. The GBM is particularly good at handling tabular data and capturing complex interactions between features.
2. **Use GBM Predictions as Additional Features**:
   * After training the GBM, the model makes predictions on both the training and test datasets. These predictions are then used as additional features (or inputs) for the next model in the pipeline: the neural network. Essentially, you're augmenting the original dataset with the predictions made by the GBM, which can provide the neural network with higher-level information about the data.
3. **Train a Neural Network**:
   * A neural network is then trained on the augmented dataset. The network can leverage the original features along with the new features generated by the GBM. Neural networks are highly flexible and can capture complex patterns in the data. They can refine the predictions made by the GBM by learning from both the raw input features and the GBM’s outputs.

**Why Use This Hybrid Approach?**

* **Leverage Strengths of Both Models**: GBMs excel in handling tabular data, especially when there are complex interactions between features. Neural networks, on the other hand, are very flexible and can model complex, non-linear relationships. Combining the two allows you to leverage the strengths of both.
* **Improved Generalization**: By stacking these models, you can often achieve better generalization to unseen data. The neural network can learn to correct the errors made by the GBM or refine its predictions.

**Steps in the Code**

1. **Training the GBM**:
   * The GradientBoostingClassifier is trained on the original dataset, and its predictions (probabilities) are generated for both the training and test sets.
2. **Augmenting the Dataset**:
   * The GBM’s predictions are added as new features to the original features, creating an augmented dataset for the neural network.
3. **Building and Training the Neural Network**:
   * A neural network is built and trained on this augmented dataset. The network has two hidden layers with ReLU activation and uses softmax activation in the output layer, indicating it's a multi-class classification task.
4. **Evaluation**:
   * The model’s performance is evaluated using accuracy, classification reports, ROC AUC scores, and a confusion matrix. These metrics provide insights into how well the model performs on different classes.
5. **Saving and Loading the Model**:
   * The GBM model is saved using joblib and can be reloaded later to make predictions without retraining the model.

**Conclusion**

This hybrid approach of combining GBMs with neural networks allows for a powerful and flexible model that can potentially outperform either method when used individually. The GBM serves as a feature generator, providing the neural network with enriched inputs that can lead to better performance on complex datasets.