Key Performance Indicators for all Models

```
% Key Performance Indicators for multiple trained models
models = {trainedModel, trainedModel1, trainedModel2, trainedModel3,
trainedModel4, ...
          trainedModel5, trainedModel6, trainedModel7, trainedModel8,
trainedModelkfold, trainedModelkfold1, trainedModelkfold2};
modelNames = {'Tree', 'Random Forest', 'KNN', 'NaiveBay', ...
              'Narrow Neural', 'Medium Neural', 'Wide Neural', 'Bilayered
Neural', 'Trilayered Neural', 'K-Fold Tree', 'K-Fold NaiveBay', 'K-Fold Random
Forest'};

% Initialize results table
results = table('Size', [numel(models), 5], ...
                 'VariableTypes',
{'string','double','double','double','double'}, ...
                 'VariableNames', {'Model', 'Accuracy', 'Precision', 'Recall',
'F1'});

% Loop through each model
for i = 1:length(models)
    model = models{i};
    name = modelNames{i};

    % Predict on full dataset
    inputTable = array2table(X_full_bal, 'VariableNames',
model.RequiredVariables);
    predictions = model.predictFcn(inputTable);

    % Convert labels
    Y_true = categorical(Y_full_bal(:));
    Y_pred = categorical(predictions(:));

    % Accuracy
    accuracy = mean(Y_pred == Y_true);

    % Confusion matrix
    [confMat, classOrder] = confusionmat(Y_true, Y_pred);

    % Macro-averaged precision, recall, F1
    precisionTotal = 0;
    recallTotal = 0;
    f1Total = 0;
    numClasses = numel(classOrder);

    for j = 1:numClasses
        TP = confMat(j,j);
        FP = sum(confMat(:,j)) - TP;
        FN = sum(confMat(j,:)) - TP;

        precision = TP / (TP + FP + eps);
        recall = TP / (TP + FN + eps);
        f1 = 2 * (precision * recall) / (precision + recall + eps);
        precisionTotal = precisionTotal + precision;
        recallTotal = recallTotal + recall;
        f1Total = f1Total + f1;
    end

    % Average metrics
    macroPrecision = precisionTotal / numClasses;
    macroRecall = recallTotal / numClasses;
    macroF1 = f1Total / numClasses;

    % Store results
    results.Model(i) = name;
    results.Accuracy(i) = accuracy * 100;
    results.Precision(i) = macroPrecision;
    results.Recall(i) = macroRecall;
    results.F1(i) = macroF1;
end

% Display overall KPI summary
disp(results);
```

```
             Model               Accuracy    Precision    Recall      F1
    ______________________    ________    _________    _______    _______

    "Tree"                     84.904       0.85236    0.84904    0.85019
    "Random Forest"            99.045       0.99048    0.99045    0.99045
    "KNN"                      94.559       0.94674    0.94559    0.94536
    "NaiveBay"                 65.805       0.67519    0.65805    0.65502
    "Narrow Neural"            89.385       0.89412    0.89385    0.89392
    "Medium Neural"            93.155        0.9317    0.93155    0.93158
    "Wide Neural"              97.295       0.97296    0.97295    0.97293
    "Bilayered Neural"         90.659       0.90692    0.90659    0.90665
    "Trilayered Neural"        90.534       0.90575    0.90534    0.90532
    "K-Fold Tree"              84.977       0.85377    0.84977    0.85111
    "K-Fold NaiveBay"          65.806       0.67534    0.65806    0.65507
    "K-Fold Random Forest"     99.355       0.99357    0.99355    0.99355
```