data = readtable('synthetic\_emotion\_dataset.csv');

features = data{:,2:end}; % All feature columns

labels = categorical(data{:,1}); % Convert label column to categorical

features = normalize(features);

cv = cvpartition(labels, 'HoldOut', 0.3);

trainIdx = training(cv);

testIdx = test(cv);

X\_train = features(trainIdx, :);

X\_test = features(testIdx, :);

y\_train = labels(trainIdx);

y\_test = labels(testIdx);

svm = fitcecoc(X\_train, y\_train);

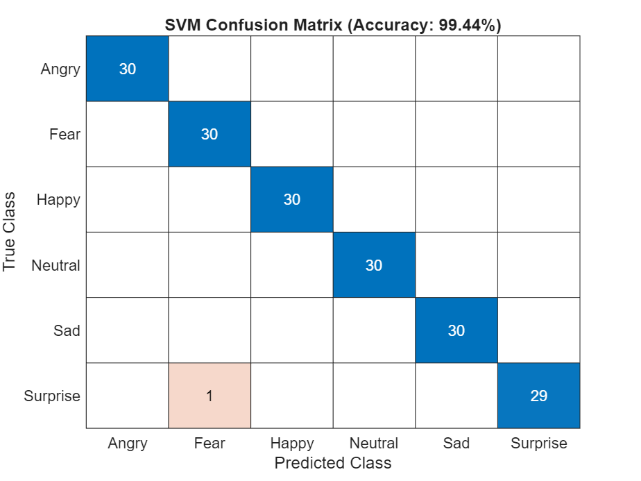
pred\_svm = predict(svm, X\_test);

acc\_svm = mean(pred\_svm == y\_test) \* 100;

figure;

confusionchart(y\_test, pred\_svm);

title(sprintf('SVM Confusion Matrix (Accuracy: %.2f%%)', acc\_svm));



tree = fitctree(X\_train, y\_train);

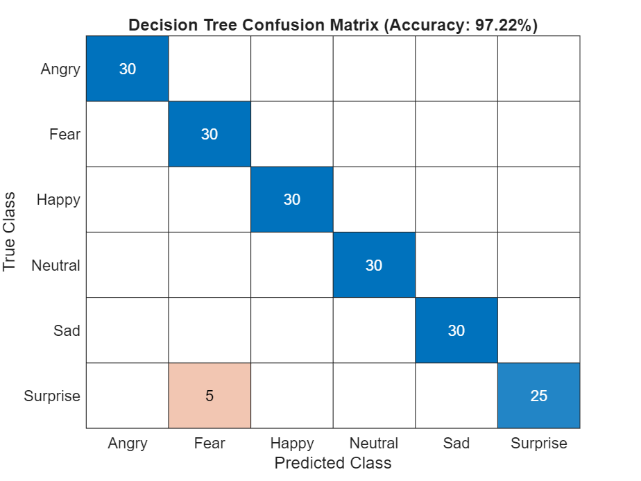
pred\_tree = predict(tree, X\_test);

acc\_tree = mean(pred\_tree == y\_test) \* 100;

figure;

confusionchart(y\_test, pred\_tree);

title(sprintf('Decision Tree Confusion Matrix (Accuracy: %.2f%%)', acc\_tree));



knn = fitcknn(X\_train, y\_train, 'NumNeighbors', 5, 'Distance', 'euclidean');

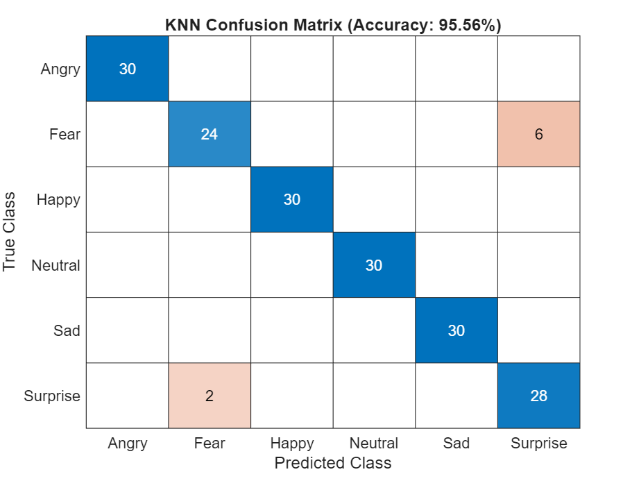
pred\_knn = predict(knn, X\_test);

acc\_knn = mean(pred\_knn == y\_test) \* 100;

figure;

confusionchart(y\_test, pred\_knn);

title(sprintf('KNN Confusion Matrix (Accuracy: %.2f%%)', acc\_knn));



ens = fitcensemble(X\_train, y\_train, 'Method', 'Bag');

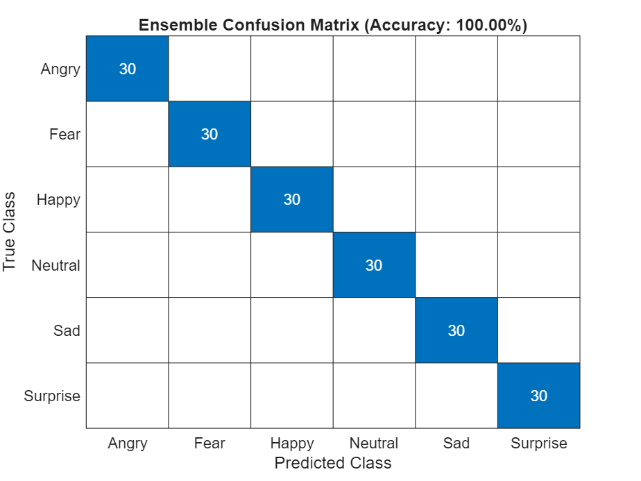
pred\_ens = predict(ens, X\_test);

acc\_ens = mean(pred\_ens == y\_test) \* 100;

figure;

confusionchart(y\_test, pred\_ens);

title(sprintf('Ensemble Confusion Matrix (Accuracy: %.2f%%)', acc\_ens));



[coeff, score] = pca(features);

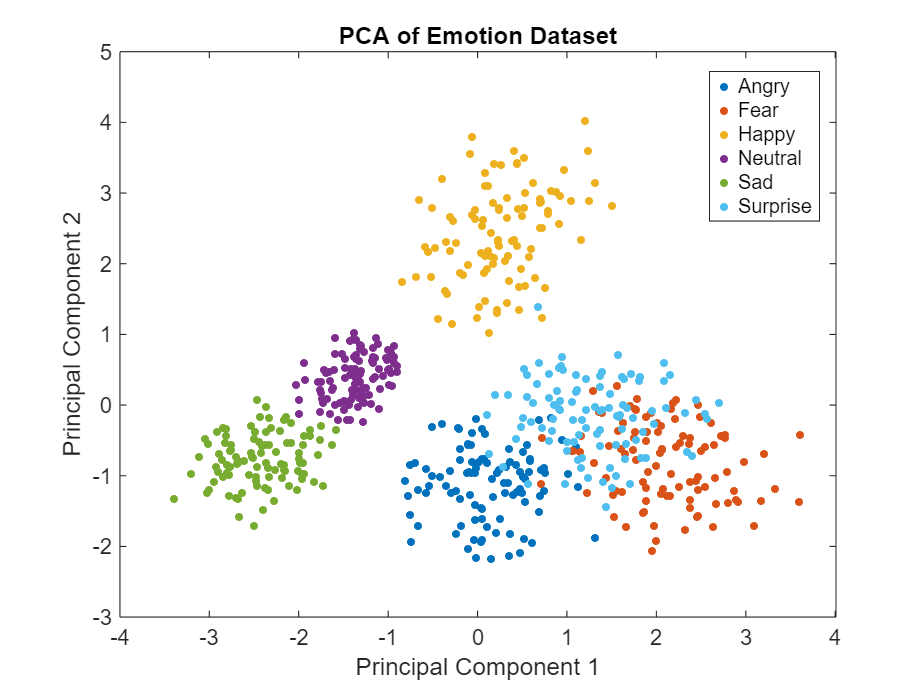
figure;

gscatter(score(:,1), score(:,2), labels);

title('PCA of Emotion Dataset');

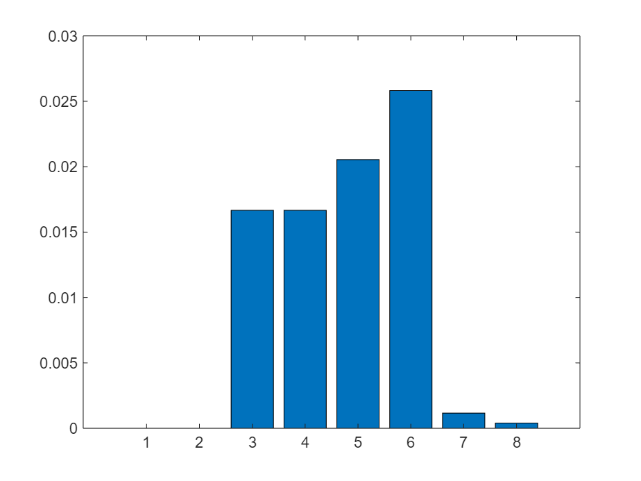
xlabel('Principal Component 1');

ylabel('Principal Component 2');



figure;

bar(tree.predictorImportance);



xticklabels(data.Properties.VariableNames(2:end));

xtickangle(45);

title('Feature Importance (Decision Tree)');

