# GaussianMLE.m

```
% Guassian Distribution Maximum Likelihood Estimation
% Please run 'Init_Read' before this script

% Maxmimum Likelihood fitting
[Means, Covs] = gaussianMLFitting(TrainSet);

% Set the priors
% Since we do not have prior knowledge of the data, the priors are
 uniform
Priors = ones(MAX_CLASS, 1)./MAX_CLASS;

% For each test sample (coming from different classes), test against
% all the classes
Likelihoods = cell(1, 1);
Denominators = cell(1, 1);
Posteriors = cell(1, 1);
for i = 1 : MAX_CLASS % For each test set
    SampleLikelihoods = zeros(size(TestSet{i}, 1), MAX_CLASS);
    for r = 1 : MAX_CLASS % test against each trained set
        DiagCov = diag(diag(Covs{r})); % Diagonalize to avoid rank
 dificient issue
        SampleLikelihoods(:, r) = mvnpdf(TestSet{i}, Means{r},
 DiagCov);
        Likelihoods{i} = SampleLikelihoods;
    end
end


% Calculate Posterior using Bayes' rule.
% This snippet is one of the examples given by the textbook for
i = 1 : MAX_CLASS
    Denominator{i} = 1 ./ (Likelihoods{i} * Priors); Posteriors{i}
    = (Likelihoods{i} * diag(Priors)); Posteriors{i} =
    (Posteriors{i}.' * diag(Denominator{i}));
end

% Check the amount of samples that are correctly labeled.
% For each test sample (coming from different classes), check if the
% maximum posterior is the correct class
CorrectCount = zeros(1, MAX_CLASS);
for i = 1 : MAX_CLASS
    [M,I] = max(Posteriors{i});
    CorrectCount(i) = nnz(I==i);
end

% Print stuff
CorrectPercentages = CorrectCount./TestCount*100;
TotalPercentage = sum(CorrectCount)/sum(TestCount);
disp(sprintf('Correct/Total: %.2f%%', TotalPercentage*100));
```