```
clear
parallel.gpu.enableCUDAForwardCompatibility(true)
canUseGPU
```

```
ans = logical
   1
```

```
gpuDevice
```

```
ans =
  CUDADevice with properties:

                Name: 'NVIDIA RTX PRO 2000 Blackwell Generation Laptop GPU'
               Index: 1 (of 1)
    ComputeCapability: '12.0'
          DriverModel: 'WDDM'
          TotalMemory: 8546549760 (8.55 GB)
      AvailableMemory: 7338979328 (7.34 GB)
      DeviceAvailable: true
       DeviceSelected: true

  Show all properties.
```

```
run("winedata.m")
```

```
reviewText = winemag_data_first150k.description;
documents = tokenizedDocument(reviewText);
documents = lower(documents);
documents = removeStopWords(documents);
bag = bagOfWords(documents)
```

```
bag =
  bagOfWords with properties:

        NumWords: 41427
          Counts: [150930×41427 double]
      Vocabulary: ["tremendous"    "100"    "%"    "varietal"    "wine"    "hails"    "oakville"    "aged"    "three
    NumDocuments: 150930
```

```
X = tfidf(bag);
```

```
size(X)
```

```
ans = 1×2
      150930       41427
```

```
nnz(X)
```

```
ans =
3961510
```

```
elements = 50;
[U, S, V] = svds(X, elements);
Z = U*S;
```

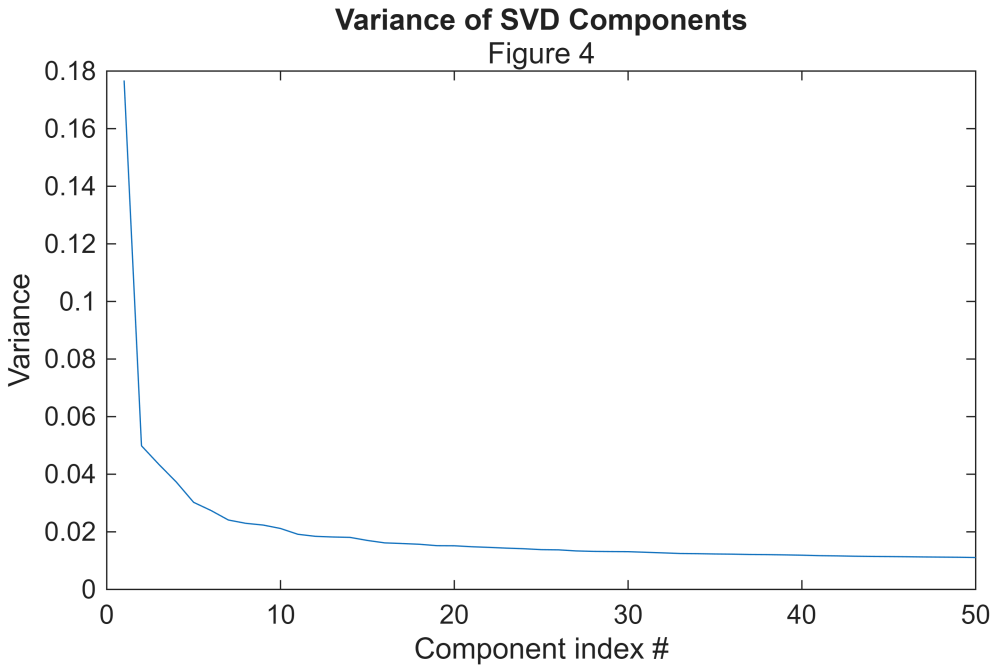
```
vocab = bag.Vocabulary;
pc1 = V(:,1);
pc2 = V(:,2);

[~, idxPos] = sort(pc1, 'descend');
topPosWords_pc1 = vocab(idxPos(1:15))
```

```
topPosWords_pc1 = 1×15 string
"%"          "fruit"      "wine"       "black"      "tannins"    "finish"     "a ···
```

```
[~, idxNeg] = sort(pc1, 'ascend');
topNegWords_pc1 = vocab(idxNeg(1:15))
```

```
topNegWords_pc1 = 1×15 string
"judi"       "flom"       "sorensen"   "gluey-sweet""mawkish"    "strawberry-ba… ···
```

```
figure
singVals = diag(S).^2;
explained = singVals / sum(singVals);
plot(explained(1:50));
xlabel("Component index #")
ylabel("Variance")
subtitle("Figure 4")
title("Variance of SVD Components");
```



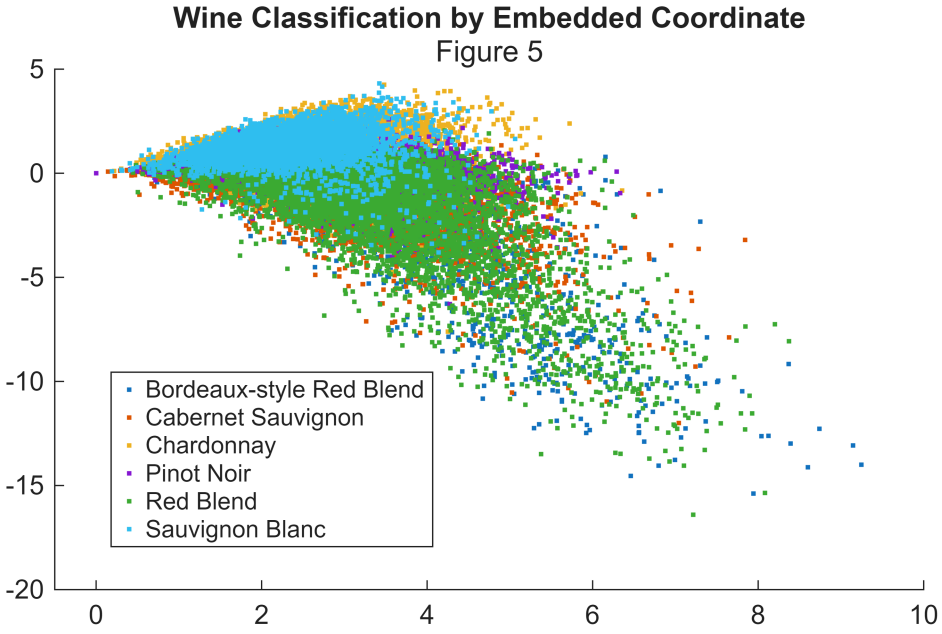

```
labels = categorical(winemag_data_first150k.variety);
[vals,~,idx] = unique(labels);
counts = histcounts(idx, 1:numel(vals)+1);
[~,order] = sort(counts, 'descend');
topN = 6;
```

```
keepVarieties = vals(order(1:topN))
```

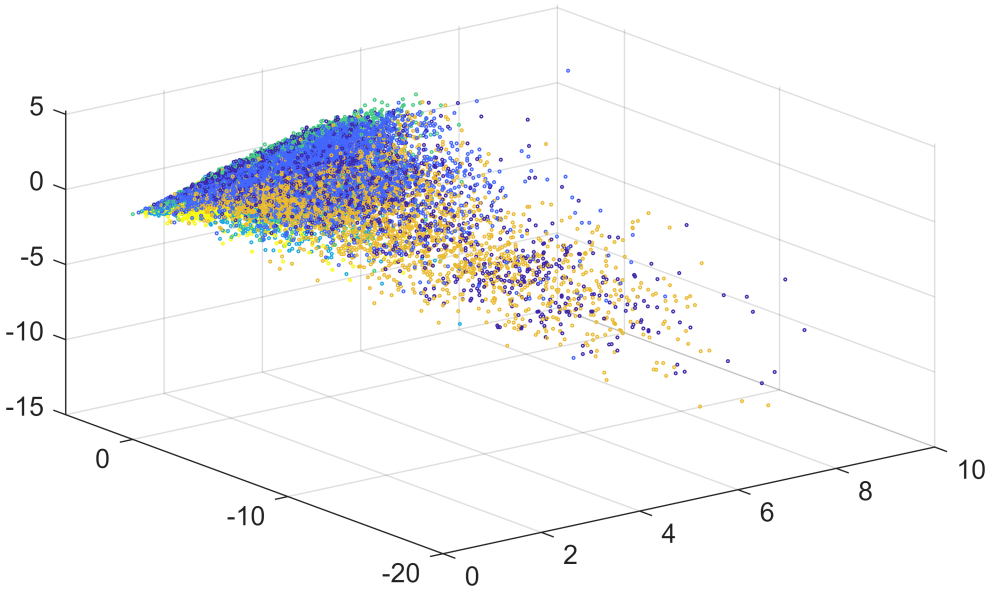
```
keepVarieties = 6×1 categorical
Chardonnay
Pinot Noir
Cabernet Sauvignon
Red Blend
Bordeaux-style Red Blend
Sauvignon Blanc
```

```
mask = ismember(labels, keepVarieties);
Zsmall = Z(mask, :);
labelSmall = removecats(labels(mask));
```

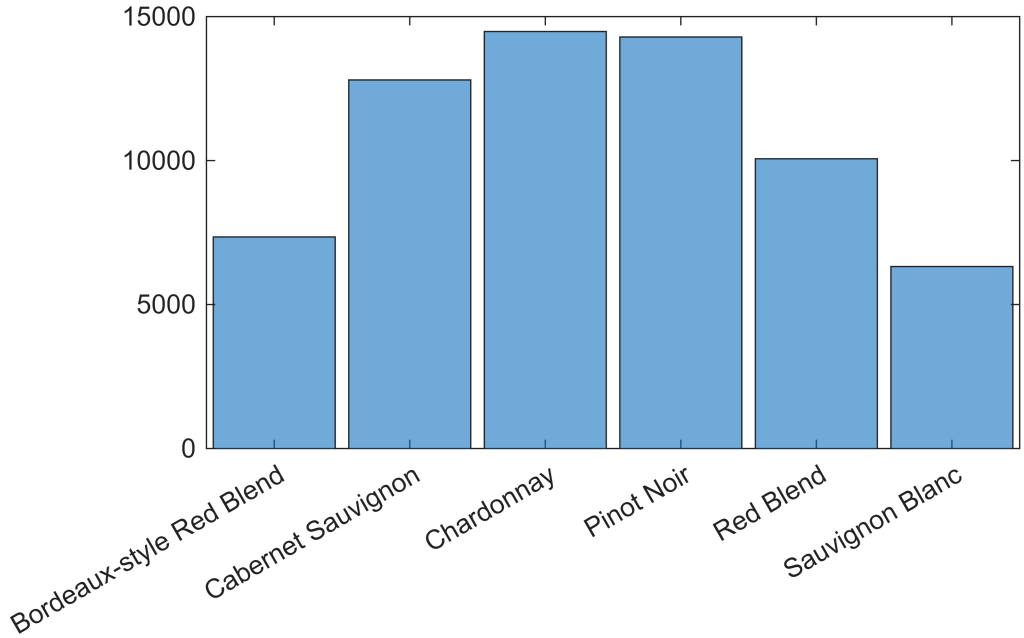
```
figure
gscat = gscatter(Zsmall(:,1), Zsmall(:,2), labelSmall);
title("Wine Classification by Embedded Coordinate")
subtitle("Figure 5")
```




```
figure
scatter3(Zsmall(:,1),Zsmall(:,2),Zsmall(:,3),1,labelSmall)
```




```
features = Zsmall(:,1:elements);
histogram(labelSmall)
```




```
textSmall  = winemag_data_first150k.description(mask);
labelSmall = removecats(labels(mask));

cv = cvpartition(labelSmall,'HoldOut',0.2);

Xtrain = features(training(cv), :)
```

```
Xtrain = 52242×50
    4.2225   -1.8515    0.2714   -1.9516   -0.4496   -0.7143   -0.3211    2.7510 ···
    2.4069    1.1692   -1.0214   -0.7934   -0.1554    0.6963   -1.3242    0.1697
    2.2453    0.1928    0.4572   -0.8629   -0.7797   -0.8457    0.0734    1.0002
    2.4419   -0.1484    0.8487   -0.1871    0.6080    0.0908   -0.5541    0.7944
    2.6550    0.0139    0.5451    0.7903    0.8889    0.3363   -0.9866   -0.1704
    4.7511    0.2594    1.1993    0.2892   -1.0003   -0.6341    0.5998    0.6442
    2.8592   -0.2027    1.1860    0.6882    0.5867    0.9202   -1.0438   -0.2885
    3.0450    1.3425   -0.7239    0.4042   -0.6098    1.1624    1.5772    0.2907
    3.2513    1.8621   -1.5814   -0.3373   -0.2955    0.3859   -1.0671    1.5338
    3.2749    0.2641    0.0793    2.6934   -0.3821   -0.0740    0.6362    0.9937
    3.8123   -2.6409    0.0980   -0.7392    0.0232   -0.3313   -1.9426   -0.3186
    2.4878    0.3323    0.0220    2.2699    0.5335   -0.7768    0.6504    1.2987
    2.5983   -0.2412   -2.9117   -0.9380    0.1906   -1.6095   -0.0585    0.3838
    2.0361    1.1384   -0.6447    0.4587   -0.6038   -0.1354   -0.5114    0.6611
    2.8673   -0.0746    0.7473    2.1212   -0.8525    1.8602    1.7030    0.0287
     ⋮
```

```
Xtest = features(test(cv), :)
```

```
Xtest = 13060×50
    4.8611   -3.5920    0.0487    0.5467   -0.2948   -0.9257   -2.5738   -0.0806 ···
    2.7603    0.6426   -0.1015   -1.0297   -0.0306    0.9530    0.2186    1.1526
    2.0539    0.4082   -0.1673   -1.2488    0.0747    0.2805   -1.5660    0.0058
    2.4344    1.2922   -0.3934   -0.7932   -0.5727    0.5710   -0.3223    1.4144
    3.3969    0.3680    0.0202    0.5527    0.4446    0.2032    0.4484    1.1662
    3.8194    0.0826    0.6321    3.6463   -1.2017    0.8597    1.6504   -0.0535
    3.9546   -4.1490   -0.9971    0.2988    0.6142    0.3115   -0.9887   -3.1194
    1.9796    0.3076    0.5528   -0.4225   -0.4358   -0.2504    0.1387    0.6997
    2.1398    0.6322   -0.4299   -0.2256   -0.2119    0.4050   -0.0397   -1.8518
    1.7180   -0.3587    1.2291   -0.4593    0.7763   -0.5604   -0.8538   -0.2275
    3.7199    0.8906   -0.5906    0.3992    1.8542    0.1926    0.6360    0.6241
    2.9978    2.3772   -1.3041   -1.5821   -0.4161    0.0206    0.6259    0.1805
    3.1361   -2.1239    1.6592   -0.0154   -0.7003   -0.2371    0.8299    0.1359
    5.0820   -5.4081   -3.4593    2.8283   -0.6645    0.3218    0.1560   -1.1801
    1.5695    0.8522   -0.6884   -0.5756    0.3929   -0.4667   -0.8356    0.2838
     ⋮
```

```
%XtrainText = string(textSmall(training(cv)));
Ytrain     = labelSmall(training(cv));

%XtestText  = string(textSmall(test(cv)));
Ytest      = labelSmall(test(cv));
```

```
mdl = fitcecoc(Xtrain, Ytrain);  % multiclass SVM
```

```
Ypred = predict(mdl, Xtest);

accuracy_svm = mean(Ypred == Ytest);
disp(accuracy_svm);
```

```
    0.8032
```

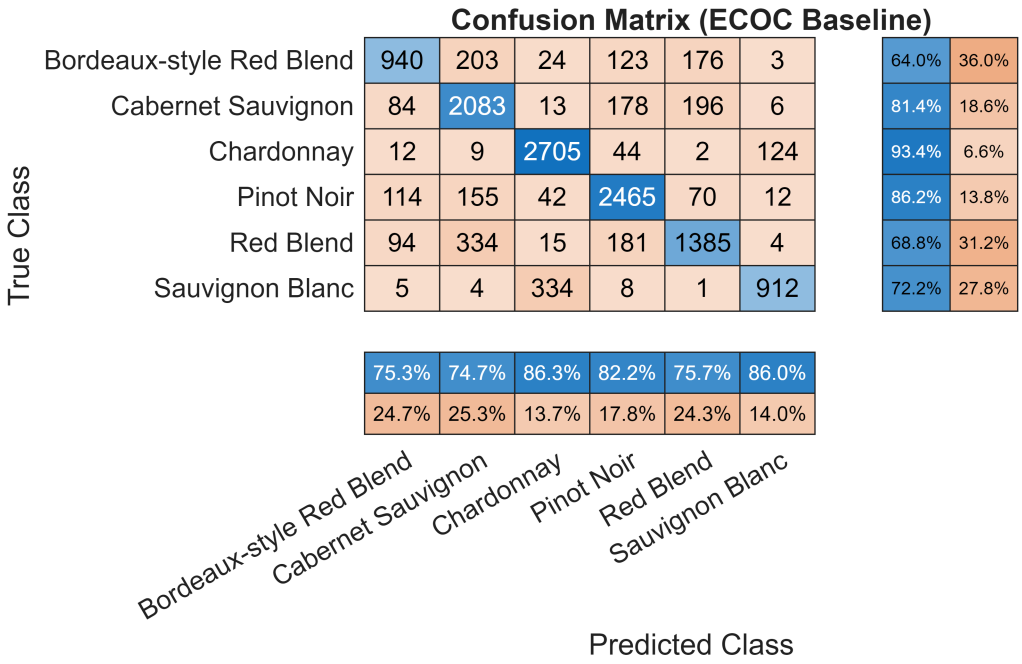
```
countcats(Ypred)
```

```
ans = 6×1
        1249
        2788
        3133
        2999
        1830
        1061
```

```
categories(Ypred)
```

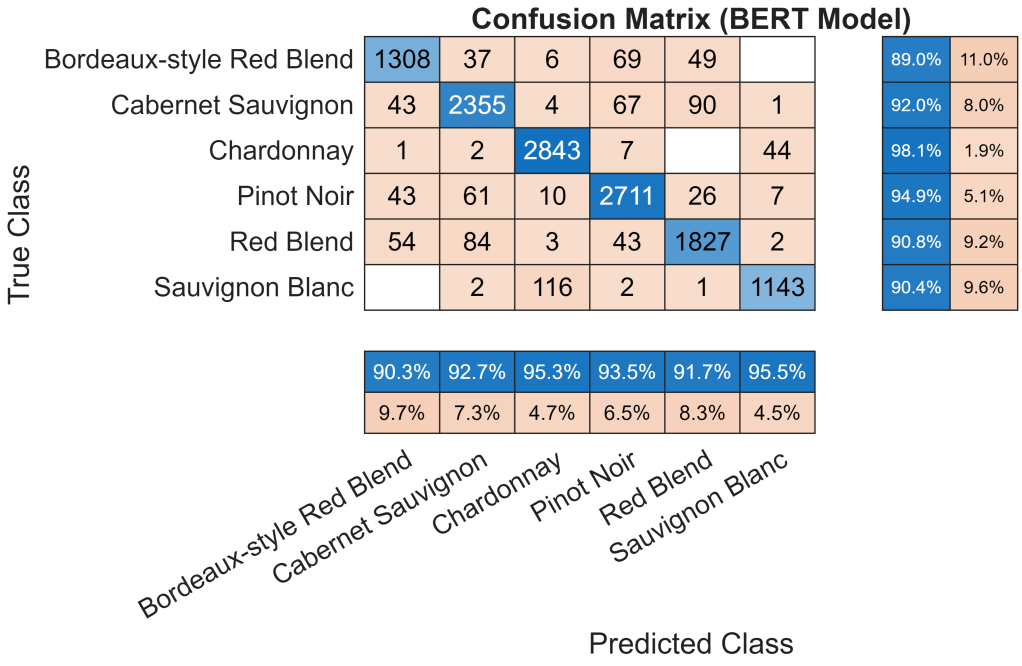
```
ans = 6×1 cell
'Bordeaux-style Red Blend'
'Cabernet Sauvignon'
'Chardonnay'
'Pinot Noir'
'Red Blend'
'Sauvignon Blanc'
```

```
[C_ecoc, order] = confusionmat(Ytest, Ypred);
figure
cm_ecoc = confusionchart(C_ecoc, order);
cm_ecoc.Title = 'Confusion Matrix (ECOC Baseline)';
cm_ecoc.RowSummary = 'row-normalized';
cm_ecoc.ColumnSummary = 'column-normalized';
```

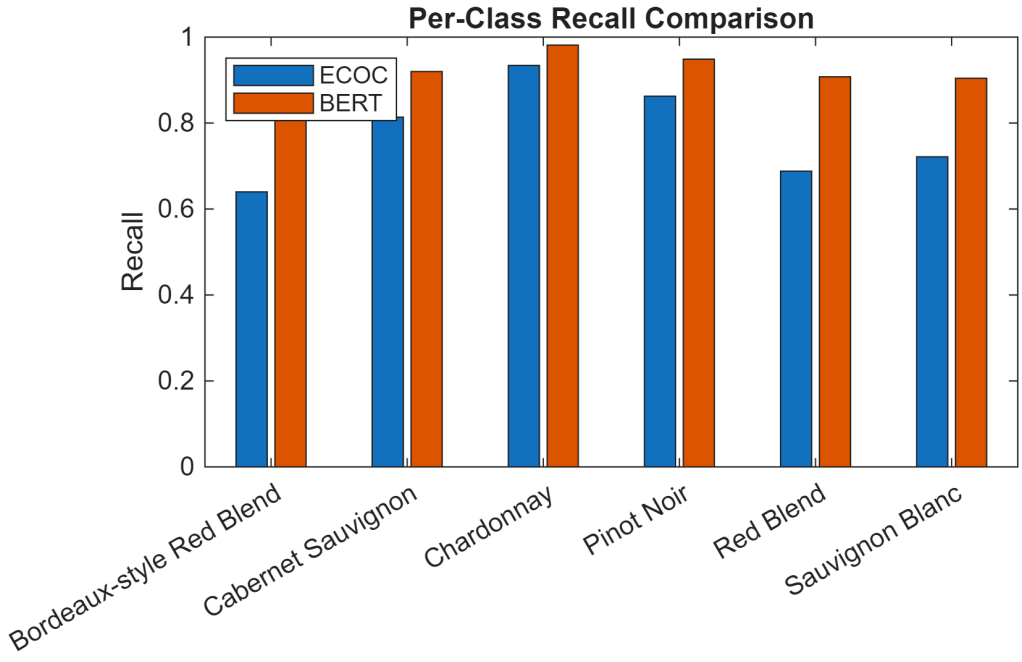



```
run("getBERTConfusionList.m")
run("getBERTConfusionMatrix.m")
figure
cm_bert = confusionchart(confusion_matrix, confusion_matrix_labels);
cm_bert.Title = 'Confusion Matrix (BERT Model)';
cm_bert.RowSummary = 'row-normalized';
```

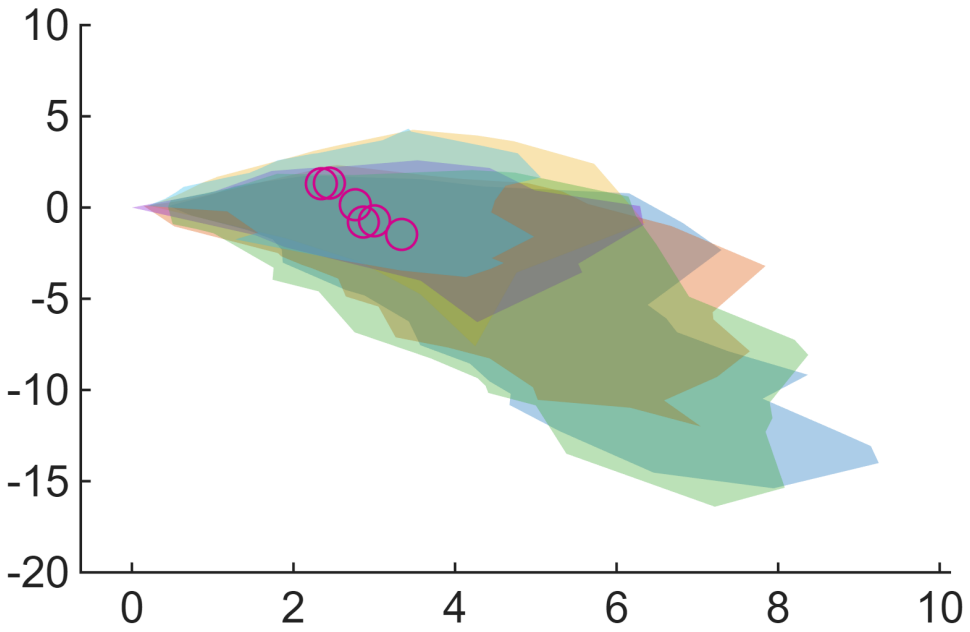
```
cm_bert.ColumnSummary = 'column-normalized';
```



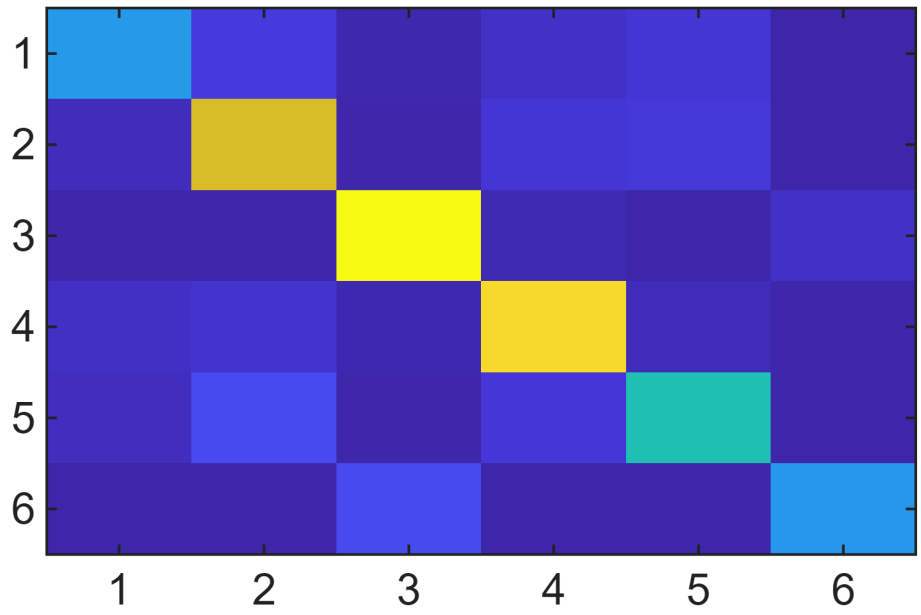

```
recall_ecoc = diag(C_ecoc) ./ sum(C_ecoc,2);
recall_bert = diag(confusion_matrix) ./ sum(confusion_matrix,2);

figure
bar([recall_ecoc recall_bert])
set(gca,'XTickLabel',order)
legend('ECOC','BERT','Location','northwest')
ylabel('Recall')
title('Per-Class Recall Comparison')
```



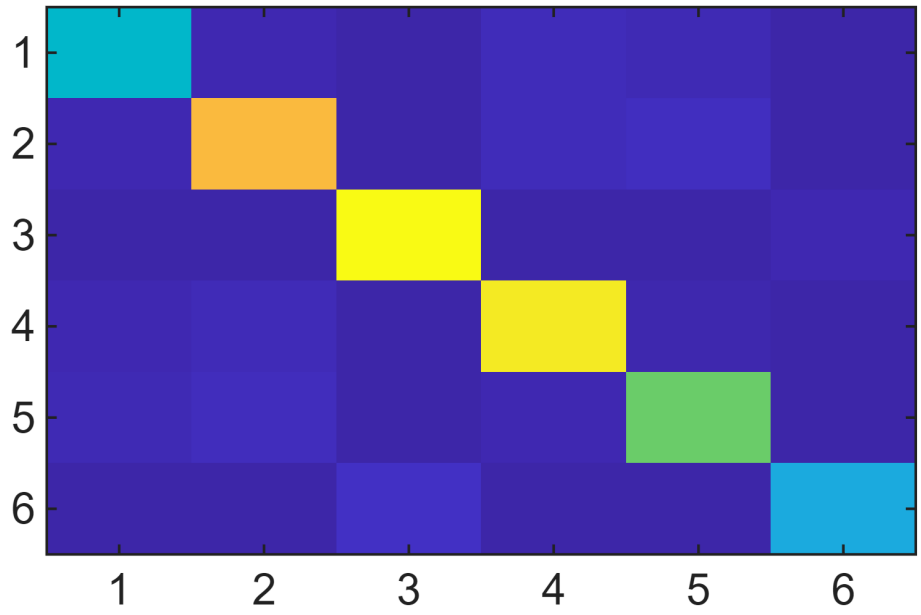
```
figure
hold on
centers = zeros(elements, topN);
for i = 1:topN
    mask = labelSmall == order(i);
    pointsSmall = Zsmall(mask, :);
    centers(:, i) = mean(pointsSmall,1);
    p = pointsSmall(:,1:2);
    b = boundary(p);
    b_points = pointsSmall(b,1:2);
    boundary_plot =
plot(polyshape(b_points),"FaceColor",gscat(i).Color,"EdgeAlpha",0);
end
p = plot(centers(1,:), centers(2,:), "o");

hold off
```




```
figure
imagesc(C_ecoc)
```




```
figure
imagesc(confusion_matrix)
```




```
labels = ["Chardonnay","Pinot Noir","Cabernet Sauvignon", ...
          "Red Blend","Bordeaux-style Red Blend","Sauvignon Blanc"];
counts = [14482 14291 12800 10062 7347 6320];

figure
barh(counts)
set(gca,'YTickLabel',labels)
xlabel('Number of Reviews')
```

```
title('Class Distribution (Top Varieties)')
```