Machine learning (Blended delivery) Assignment 2 (Autumn 2023)

Student number: 22203536

Problem specification:

```
b'edm'      4461
b'rap'      4339
b'pop'      4044
b'latin'    3748
b'rock'     3519
Name: playlist_genre, dtype: int64
```

Based on the value counts for playlist_genre it seems we are dealing with a multi-class classification problem with fairly balanced classes.

Dataset Exploration

I began my analysis by exploring the dataset using python in jupyter notebook. The following are some methods I used when exploring the data.

```
num_rows = df.shape[0]
num_cols = df.shape[1]
datatypes = df.dtypes
missing_values = df.isnull().sum()
print(df.head(10))
print(df.tail(10))
num_duplicates = df.duplicated().sum()
df = df.drop_duplicates()
```

After dataset exploration it was determined there were no missing values and that the values in the dataset were coherent meaningful values based on the feature description. There were 1701 duplicates that were removed as they did not represent meaningful data . The features were all in the correct datatypes save 2 features, key and mode that were then changed to categorical data type as they represented only a few different categories. I then saved the updated data and uploaded it to weka where I made further changes to the data. I started by normalizing the features between 0 and 1 as this is required by some of the algorithms needed for this assignment such as KNN for example.

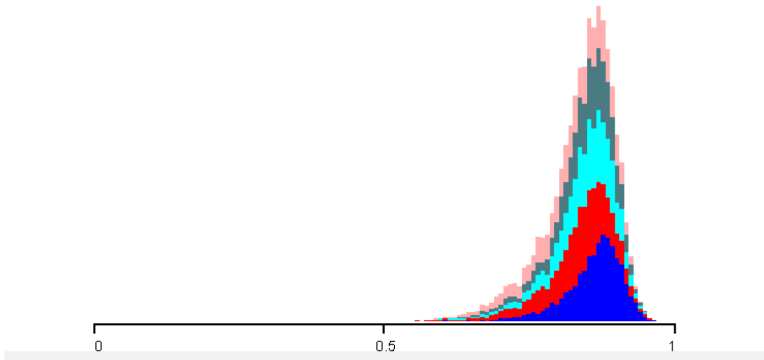

Fig 2. Represents a features scale after normalization

I then used the InterquartileRange and RemoveWithValues methods to remove outliers. However since this actually made performance on the algorithms worse I undid this. I also encoded the nominal features using one hot encoding using the NominalToBinary filter. This had a modest benefit for performance.

Beginning of questions 1.1 to 1.5

1.1

Decision Tree Classifier

- Accuracy: 49.23%. This shows a better performance compared to KNN, likely due to its capability to handle complex relationships between features.
- Kappa Statistic: 0.3634. A higher kappa statistic compared to KNN indicates a better agreement.
- F-Measure: Shows a better balance between precision and recall compared to KNN, especially for 'edm' and ‘rock'.
- ROC and PRC Areas: Improved performance compared to KNN, indicating a better model fit.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        9900               49.2268 %
Incorrectly Classified Instances     10211               50.7732 %
Kappa statistic                          0.3634
Mean absolute error                      0.2289
Root mean squared error                  0.4022
Relative absolute error                 71.6658 %
Root relative squared error            100.6525 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.598    0.144    0.541      0.598    0.568      0.438    0.735     0.465     b'edm'
                 0.644    0.129    0.579      0.644    0.610      0.496    0.766     0.485     b'rap'
                 0.253    0.151    0.297      0.253    0.273      0.109    0.543     0.245     b'pop'
                 0.323    0.116    0.388      0.323    0.352      0.222    0.616     0.289     b'latin'
                 0.627    0.096    0.581      0.627    0.603      0.515    0.783     0.470     b'rock'
Weighted Avg.    0.492    0.129    0.479      0.492    0.484      0.358    0.689     0.393

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2666  442  694  383  276 |    a = b'edm'
  403 2794  419  540  183 |    b = b'rap'
  942  559 1024  733  786 |    c = b'pop'
  593  852  745 1209  349 |    d = b'latin'
  321  177  567  247 2207 |    e = b'rock'
```

Neural Network

- Accuracy: 54.83%. The best among the three, suggesting that the Neural Network is more adept at capturing complex patterns.
- Kappa Statistic: 0.4338. This highest kappa statistic among the three indicates the best agreement between the predicted and actual classifications.
- F-Measure: Consistently higher across classes, indicating a balanced precision-recall trade-off.
- ROC and PRC Areas: Highest among the three models, indicating superior performance in class distinction.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11027               54.8307 %
Incorrectly Classified Instances      9084               45.1693 %
Kappa statistic                          0.4338
Mean absolute error                      0.2264
Root mean squared error                  0.3435
Relative absolute error                 70.8803 %
Root relative squared error             85.9571 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.644    0.122    0.600      0.644    0.621      0.509    0.849     0.673     b'edm'
                 0.678    0.115    0.618      0.678    0.647      0.545    0.870     0.664     b'rap'
                 0.354    0.146    0.380      0.354    0.366      0.214    0.704     0.344     b'pop'
                 0.336    0.091    0.457      0.336    0.387      0.277    0.736     0.402     b'latin'
                 0.716    0.091    0.624      0.716    0.667      0.593    0.906     0.678     b'rock'
Weighted Avg.    0.548    0.114    0.537      0.548    0.540      0.429    0.813     0.555

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2874  378  676  274  259 |    a = b'edm'
  386 2944  376  482  151 |    b = b'rap'
  818  488 1432  542  764 |    c = b'pop'
  464  860  825 1258  341 |    d = b'latin'
  247   95  462  196 2519 |    e = b'rock'
```

kNN (k=1)

K-Nearest Neighbors (k=1):

- Accuracy : 41.43%. This lower accuracy indicates that KNN with k=1 may be overly sensitive to noise in the dataset.
- Kappa Statistic: 0.2657. This value suggests a moderate agreement between the predicted and actual classifications, but it's lower compared to other classifiers.
- F-Measure for Each Class: Varied across classes, with 'rock' performing the best. This could indicate that some genres are more distinct than others.
- ROC and PRC Areas: Moderate performance, but it indicates that there's room for improvement, especially in distinguishing between classes.

```
== Stratified cross-validation ===
== Summary ===

orrectly Classified Instances       8331               41.4251 %
ncorrectly Classified Instances    11780               58.5749 %
appa statistic                         0.2657
ean absolute error                     0.2346
oot mean squared error                 0.4781
elative absolute error                73.4401 %
oot relative squared error           119.6333 %
otal Number of Instances           20111

== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.499    0.184    0.435      0.499    0.465      0.300    0.652     0.347     b'edm'
                0.477    0.112    0.539      0.477    0.506      0.382    0.687     0.390     b'rap'
                0.267    0.205    0.247      0.267    0.257      0.061    0.523     0.215     b'pop'
                0.318    0.147    0.332      0.318    0.325      0.174    0.582     0.241     b'latin'
                0.502    0.087    0.551      0.502    0.525      0.431    0.720     0.384     b'rock'
eighted Avg.    0.414    0.149    0.421      0.414    0.416      0.269    0.632     0.317

== Confusion Matrix ===

   a    b    c    d    e   <-- classified as
2224  444 1004  490  299 |    a = b'edm'
 671 2068  605  767  228 |    b = b'rap'
1096  451 1079  807  611 |    c = b'pop'
 669  689  895 1193  302 |    d = b'latin'
 449  184  782  337 1767 |    e = b'rock'
```

Conclusion

In conclusion, the Neural Network demonstrates the highest efficacy in classifying the genres in the given dataset, with superior performance across most metrics. This is followed by the Decision Tree and then KNN with k=1. The higher performance of the Neural Network can be attributed to its ability to model complex and non-linear relationships in the data. I feel it's also important to consider the model complexity and the computational resources required.

Why did I pick the metrics I picked ?

First I noted it was a multiclass classification problem with relatively balanced classes (shown in the problem specification on page 1 of this report). With that in mind.

Accuracy Fundamental to understanding the overall effectiveness of the model in classifying instances correctly.

Kappa Statistic: It provides insight into the agreement between the predicted and actual classifications, adjusting for the agreement that would be expected by chance.

F-Measure: As a harmonic mean of precision and recall, it is crucial in a balanced multiclass setting to understand how well the classifier is performing in terms of both false positives and false negatives.

ROC and PRC Areas: These provide a comprehensive view of the model's performance across different thresholds, which is particularly important in multiclass problems to evaluate the model's ability to distinguish between classes.

1.1 (second half of 1.1)

Majority vote

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10491               52.1655 %
Incorrectly Classified Instances      9620               47.8345 %
Kappa statistic                          0.4005
Mean absolute error                      0.1913
Root mean squared error                  0.4374
Relative absolute error                 59.9082 %
Root relative squared error            109.4607 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.624    0.133    0.572      0.624    0.597      0.476    0.746     0.440     b'edm'
                 0.650    0.111    0.618      0.650    0.634      0.530    0.770     0.477     b'rap'
                 0.282    0.153    0.316      0.282    0.298      0.134    0.564     0.234     b'pop'
                 0.356    0.114    0.416      0.356    0.384      0.257    0.621     0.268     b'latin'
                 0.685    0.088    0.623      0.685    0.653      0.576    0.799     0.482     b'rock'
Weighted Avg.    0.522    0.121    0.510      0.522    0.515      0.396    0.700     0.382

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2785  365  724  366  221 |    a = b'edm'
  416 2822  414  516  171 |    b = b'rap'
  908  481 1139  765  751 |    c = b'pop'
  518  787  793 1334  316 |    d = b'latin'
  242  112  530  224 2411 |    e = b'rock'
```

Average of probs

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10111               50.276  %
Incorrectly Classified Instances     10000               49.724  %
Kappa statistic                          0.377
Mean absolute error                      0.2299
Root mean squared error                  0.3611
Relative absolute error                 71.9954 %
Root relative squared error             90.3614 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.600    0.139    0.552      0.600    0.575      0.448    0.824     0.630     b'edm'
                 0.620    0.110    0.609      0.620    0.614      0.507    0.863     0.653     b'rap'
                 0.260    0.159    0.292      0.260    0.275      0.106    0.644     0.275     b'pop'
                 0.359    0.128    0.391      0.359    0.374      0.239    0.716     0.379     b'latin'
                 0.666    0.087    0.619      0.666    0.642      0.563    0.904     0.668     b'rock'
Weighted Avg.    0.503    0.126    0.494      0.503    0.498      0.373    0.790     0.523

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2676  379  774  410  222 |    a = b'edm'
  440 2691  418  606  184 |    b = b'rap'
  941  481 1053  833  736 |    c = b'pop'
  530  753  820 1347  298 |    d = b'latin'
  261  118  545  251 2344 |    e = b'rock'
```

Maximum probs

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8394               41.7384 %
Incorrectly Classified Instances     11717               58.2616 %
Kappa statistic                          0.2702
Mean absolute error                      0.2428
Root mean squared error                  0.3604
Relative absolute error                 76.0291 %
Root relative squared error             90.1783 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.484    0.159    0.464      0.484    0.474      0.320    0.813     0.607     b'edm'
                 0.493    0.117    0.536      0.493    0.514      0.388    0.855     0.636     b'rap'
                 0.260    0.202    0.244      0.260    0.252      0.056    0.637     0.266     b'pop'
                 0.329    0.158    0.323      0.329    0.326      0.169    0.709     0.371     b'latin'
                 0.514    0.092    0.542      0.514    0.527      0.431    0.896     0.649     b'rock'
Weighted Avg.    0.417    0.147    0.423      0.417    0.420      0.273    0.782     0.508

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2160  461  980  548  312 |    a = b'edm'
  558 2141  586  818  236 |    b = b'rap'
  944  491 1051  892  666 |    c = b'pop'
  579  711  910 1234  314 |    d = b'latin'
  413  189  775  334 1808 |    e = b'rock'
```

Analysis of the ensembles

- Majority Vote: Achieved 52.17% accuracy with a Kappa statistic of 0.4005. Notably better than KNN k=1, but slightly better than the Decision Tree and significantly underperformed compared to the Neural Network.
- Average of Probabilities: Recorded 50.28% accuracy with a Kappa statistic of 0.377. This was lower than the Majority Vote ensemble and also underperformed compared to the Neural Network, but still better than KNN k=1.
- Maximum Probability: Showed the lowest performance with 41.74% accuracy and a Kappa statistic of 0.2702, underperforming all individual classifiers.
- The Majority Vote ensemble provided a balanced output, slightly enhancing the Decision Tree's performance but not reaching the Neural Network's effectiveness.
- The Average of Probabilities method showed a decrease in performance compared to the Majority Vote, suggesting that averaging probabilities may dilute the effect of stronger classifiers.
- The Maximum Probability ensemble performed the worst, indicating that relying solely on the classifier with the highest confidence for each instance may not always yield the best results, especially if it overly trusts a weaker classifier.

Why were there differences in performance of the combination rules?

1. **Majority Voting**:
    - Majority Vote tends to be a robust method as it requires a consensus among classifiers. It works well when classifiers are diverse and make independent errors. However, its performance is limited by the accuracy of the majority, which explains why it couldn't surpass the Neural Network's performance.
2. **Average of Probabilities**:

- This method considers the probability estimates from each classifier and averages them. It is effective when classifiers are well-calibrated. However, in this case, it seems that averaging diluted the impact of the stronger classifier (the Neural Network), leading to a reduction in overall accuracy.

3. **Maximum Probability**:
    - This rule selects the class with the maximum probability as predicted by any of the classifiers. It heavily relies on the confidence of individual classifiers and can be misled if a weak classifier is overly confident. This could explain its lowest performance, as it may have been overly influenced by the weaker KNN k=1 classifier.

Part 1.2

Decision Tree

2 iterations

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        9243               45.9599 %
Incorrectly Classified Instances     10868               54.0401 %
Kappa statistic                          0.3222
Mean absolute error                      0.2254
Root mean squared error                  0.3893
Relative absolute error                 70.5879 %
Root relative squared error             97.4183 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.568    0.152    0.517      0.568    0.541      0.403    0.771     0.537     b'edm'
                 0.580    0.131    0.549      0.580    0.564      0.440    0.801     0.544     b'rap'
                 0.264    0.178    0.272      0.264    0.268      0.087    0.576     0.250     b'pop'
                 0.312    0.131    0.354      0.312    0.332      0.191    0.652     0.316     b'latin'
                 0.554    0.086    0.578      0.554    0.566      0.476    0.834     0.547     b'rock'
Weighted Avg.    0.460    0.137    0.455      0.460    0.456      0.321    0.727     0.441

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2536  457  797  436  235 |    a = b'edm'
  491 2517  502  647  182 |    b = b'rap'
  986  571 1069  720  698 |    c = b'pop'
  558  856  854 1170  310 |    d = b'latin'
  338  185  712  333 1951 |    e = b'rock'
```

18 iterations (optimal)

```
Correctly Classified Instances       10880               54.0997 %
Incorrectly Classified Instances      9231               45.9003 %
Kappa statistic                          0.4249
Mean absolute error                      0.2255
Root mean squared error                  0.3472
Relative absolute error                 70.6189 %
Root relative squared error             86.8796 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.633    0.113    0.615      0.633    0.624      0.514    0.837     0.676     b'edm'
                 0.696    0.114    0.627      0.696    0.659      0.561    0.876     0.671     b'rap'
                 0.286    0.146    0.329      0.286    0.306      0.147    0.653     0.285     b'pop'
                 0.371    0.110    0.436      0.371    0.401      0.278    0.730     0.423     b'latin'
                 0.707    0.091    0.623      0.707    0.662      0.587    0.913     0.698     b'rock'
Weighted Avg.    0.541    0.116    0.528      0.541    0.533      0.419    0.802     0.553

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2824  363  715  346  213 |    a = b'edm'
  296 3020  343  527  153 |    b = b'rap'
  823  509 1155  733  824 |    c = b'pop'
  449  821  769 1392  317 |    d = b'latin'
  203  107  525  195 2489 |    e = b'rock'
```

For the decision tree classifier from 2 to 18 iterations (i.e 2,4,6,8…..) the performance was increasing i.e increasing accuracy from 45.95% to 54.1% and increasing kappa statistic from 0.32 to 0.42 and also increasing precision/ recall / F measure/ ROC /PRC . However the performance leveled off at 18 iterations and did not increase for 20 iterations.

KNN k=1 (18 iterations)

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8425               41.8925 %
Incorrectly Classified Instances     11686               58.1075 %
Kappa statistic                          0.2721
Mean absolute error                      0.2372
Root mean squared error                  0.4133
Relative absolute error                 74.276  %
Root relative squared error            103.4321 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.488    0.166    0.456      0.488    0.471      0.314    0.735     0.496     b'edm'
                 0.483    0.108    0.553      0.483    0.515      0.395    0.775     0.550     b'rap'
                 0.267    0.206    0.246      0.267    0.256      0.059    0.581     0.251     b'pop'
                 0.336    0.159    0.326      0.336    0.331      0.174    0.645     0.322     b'latin'
                 0.515    0.088    0.553      0.515    0.533      0.439    0.811     0.538     b'rock'
Weighted Avg.    0.419    0.147    0.427      0.419    0.422      0.276    0.709     0.433

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2178  419 1019  552  293 |    a = b'edm'
  599 2095  628  792  225 |    b = b'rap'
  996  433 1080  903  632 |    c = b'pop'
  585  666  921 1258  318 |    d = b'latin'
  422  178  748  357 1814 |    e = b'rock'
```

KNN K=1 , 20 iterations

```
:orrectly Classified Instances        8431               41.9223 %
:ncorrectly Classified Instances     11680               58.0777 %
:appa statistic                          0.2725
lean absolute error                      0.2371
:oot mean squared error                  0.4128
:elative absolute error                 74.2468 %
:oot relative squared error            103.2922 %
'otal Number of Instances            20111

== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.488    0.166    0.456      0.488    0.472      0.315    0.737     0.498     b'edm'
                0.482    0.107    0.552      0.482    0.515      0.394    0.777     0.551     b'rap'
                0.268    0.207    0.246      0.268    0.256      0.059    0.583     0.251     b'pop'
                0.338    0.159    0.328      0.338    0.332      0.177    0.647     0.323     b'latin'
                0.515    0.088    0.553      0.515    0.534      0.439    0.812     0.539     b'rock'
leighted Avg.   0.419    0.147    0.428      0.419    0.423      0.277    0.711     0.435

== Confusion Matrix ===

   a    b    c    d    e   <-- classified as
2176  413 1024  553  295 |    a = b'edm'
 604 2091  629  792  223 |    b = b'rap'
 988  438 1085  899  634 |    c = b'pop'
 583  668  919 1265  313 |    d = b'latin'
 416  175  761  353 1814 |    e = b'rock'
```

KNN k=1 continued to show increased performance for higher numbers of iterations (i.e from 2,4,6,8…..etc) showing higher accuracy / F1 Measure/ precision / recall / kappa/ ROC/PRC statistic however this leveled off at 20 iterations and only increased by a tiny amount from 18 iterations suggesting a peak at 20 iterations. Accuracy at 20 iterations was only 0.02% higher than at 18 iterations and F1/Precision/Recall were similarly only 0.01% or less higher at 20 than 18 iterations.

Neural net (optimal 18 iterations)

```
Time taken to build model: 719.38 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11293               56.1533 %
Incorrectly Classified Instances      8818               43.8467 %
Kappa statistic                          0.4507
Mean absolute error                      0.231
Root mean squared error                  0.3383
Relative absolute error                 72.3299 %
Root relative squared error             84.6444 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.609    0.101    0.632      0.609    0.620      0.515    0.857     0.679     b'edm'
                 0.700    0.113    0.631      0.700    0.664      0.567    0.879     0.685     b'rap'
                 0.406    0.155    0.397      0.406    0.402      0.249    0.726     0.366     b'pop'
                 0.375    0.093    0.480      0.375    0.421      0.311    0.760     0.456     b'latin'
                 0.706    0.086    0.634      0.706    0.668      0.595    0.917     0.707     b'rock'
Weighted Avg.    0.562    0.111    0.556      0.562    0.557      0.448    0.828     0.581

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2718  416  820  270  237 |    a = b'edm'
  312 3039  334  522  132 |    b = b'rap'
  679  470 1643  534  718 |    c = b'pop'
  372  808  813 1407  348 |    d = b'latin'
  221   81  530  201 2486 |    e = b'rock'
```

For the neural network, the performance continued to increase with increasing iterations (2,4,8,10….) i.e (accuracy/f1/precision/recall/kappa/ROC/PRC) for example accuracy is 2% higher at 18 iterations than the model made earlier. The performance leveled off at 18 iterations. At 20 iterations the performance was very slightly worse i.e less than 0.1% worse on accuracy/f1/precision/recall and 0.006 lower kappa statistic.

1.2 (second half of 1.2)

Decision tree 18 iterations
70% Bag size

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11154               55.4622 %
Incorrectly Classified Instances      8957               44.5378 %
Kappa statistic                          0.442
Mean absolute error                      0.2279
Root mean squared error                  0.3408
Relative absolute error                 71.3532 %
Root relative squared error             85.285  %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.641    0.108    0.629      0.641    0.635      0.530    0.849     0.695     b'edm'
                 0.713    0.114    0.632      0.713    0.670      0.574    0.883     0.682     b'rap'
                 0.305    0.143    0.349      0.305    0.326      0.171    0.676     0.302     b'pop'
                 0.371    0.102    0.455      0.371    0.409      0.292    0.741     0.441     b'latin'
                 0.731    0.090    0.632      0.731    0.678      0.606    0.919     0.710     b'rock'
Weighted Avg.    0.555    0.112    0.541      0.555    0.546      0.436    0.814     0.568

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2860  364  699  330  208 |    a = b'edm'
  287 3093  328  467  164 |    b = b'rap'
  790  513 1235  693  813 |    c = b'pop'
  417  831  792 1392  316 |    d = b'latin'
  190   95  484  176 2574 |    e = b'rock'
```

For the decision tree the performance was higher at 80% bag size compared to 100% with higher accuracy/ precision/ recall/ROC/PRC/ F measure and kappa statistic. I then tried for 60% and noticed performance was slightly worse. At 70% the performance was very marginally higher than at 60% and at 75% it was worse than 70% suggesting an optimum between 70% and 75%.

KNN k=1 20 iterations

15% bag size

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        9521              47.3423 %
Incorrectly Classified Instances     10590              52.6577 %
Kappa statistic                          0.3399
Mean absolute error                      0.2493
Root mean squared error                  0.366
Relative absolute error                 78.0517 %
Root relative squared error             91.5773 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.577    0.159    0.509      0.577    0.541      0.400    0.784     0.572     b'edm'
                 0.526    0.092    0.611      0.526    0.565      0.459    0.823     0.602     b'rap'
                 0.345    0.194    0.310      0.345    0.326      0.145    0.636     0.284     b'pop'
                 0.373    0.133    0.391      0.373    0.382      0.244    0.700     0.370     b'latin'
                 0.531    0.081    0.580      0.531    0.554      0.466    0.859     0.596     b'rock'
Weighted Avg.    0.473    0.133    0.481      0.473    0.476      0.344    0.760     0.487

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2576  315  894  416  260 |    a = b'edm'
  551 2284  569  743  192 |    b = b'rap'
  945  400 1394  703  602 |    c = b'pop'
  521  629  901 1399  298 |    d = b'latin'
  472  113  745  321 1868 |    e = b'rock'
```

For KNN on the other hand the performance (accuracy/precision/recall/f1/kappa/ROC/PRC) continued to increase with smaller bag sizes until 15%. But then the performance went down somewhat for 10% indicating the optimal bag size is between 10 and 15%.

Neural network 18 iterations (90% bag size)

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11250              55.9395 %
Incorrectly Classified Instances      8861              44.0605 %
Kappa statistic                          0.4481
Mean absolute error                      0.2313
Root mean squared error                  0.3389
Relative absolute error                 72.4151 %
Root relative squared error             84.8008 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.608    0.102    0.629      0.608    0.618      0.512    0.854     0.674     b'edm'
                 0.695    0.113    0.627      0.695    0.659      0.561    0.876     0.682     b'rap'
                 0.393    0.150    0.397      0.393    0.395      0.244    0.724     0.365     b'pop'
                 0.376    0.097    0.471      0.376    0.418      0.305    0.759     0.450     b'latin'
                 0.718    0.089    0.632      0.718    0.672      0.599    0.919     0.709     b'rock'
Weighted Avg.    0.559    0.111    0.553      0.559    0.554      0.445    0.826     0.578

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2712  426  791  301  231 |    a = b'edm'
  331 3015  340  515  138 |    b = b'rap'
  678  461 1590  573  742 |    c = b'pop'
  370  826  786 1408  358 |    d = b'latin'
  224   77  499  194 2525 |    e = b'rock'
```

For the neural network performance was worse at 90% and even further worse at 80% and 70%....etc . Implying that the optimal bag size is between 90% and 100%

1.3

First before varying the subspace we need to find the optimal number of iterations for each of the algorithms again.

Decision tree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11017               54.781  %
Incorrectly Classified Instances      9094               45.219  %
Kappa statistic                          0.433
Mean absolute error                      0.2564
Root mean squared error                  0.3505
Relative absolute error                 80.2769 %
Root relative squared error             87.7147 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.663    0.121    0.609      0.663    0.635      0.526    0.839     0.663     b'edm'
                 0.727    0.127    0.611      0.727    0.664      0.565    0.877     0.649     b'rap'
                 0.273    0.129    0.348      0.273    0.306      0.159    0.647     0.284     b'pop'
                 0.356    0.096    0.460      0.356    0.401      0.288    0.733     0.377     b'latin'
                 0.701    0.094    0.614      0.701    0.655      0.577    0.910     0.671     b'rock'
Weighted Avg.    0.548    0.114    0.530      0.548    0.535      0.425    0.801     0.532

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2956  395  609  305  196 |    a = b'edm'
  300 3156  285  412  186 |    b = b'rap'
  847  571 1105  688  833 |    c = b'pop'
  477  897  704 1333  337 |    d = b'latin'
  272  147  473  160 2467 |    e = b'rock'
```

After testing iterations in increasing order (2,4,6,8 etc) the accuracy and other benchmarks (f1 /precision/ recall / kappa /ROC/PRC) continued to increase till 20 and then leveled off at 20.

KNN k=1

After testing iterations from 2 to 20 KNN also continued to increase but leveled off at 20 iterations for accuracy / f1/ precision /recall/kappa/ROC/PRC.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10037               49.908  %
Incorrectly Classified Instances     10074               50.092  %
Kappa statistic                          0.3715
Mean absolute error                      0.259
Root mean squared error                  0.3713
Relative absolute error                 81.0949 %
Root relative squared error             92.9177 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.621    0.152    0.539      0.621    0.577      0.447    0.786     0.530     b'edm'
                 0.639    0.121    0.593      0.639    0.615      0.505    0.838     0.586     b'rap'
                 0.274    0.156    0.307      0.274    0.289      0.123    0.586     0.251     b'pop'
                 0.343    0.118    0.401      0.343    0.370      0.240    0.681     0.324     b'latin'
                 0.596    0.083    0.604      0.596    0.600      0.516    0.874     0.602     b'rock'
Weighted Avg.    0.499    0.127    0.490      0.499    0.493      0.368    0.753     0.460

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2772  409  707  357  216 |    a = b'edm'
  414 2773  407  571  174 |    b = b'rap'
 1003  510 1108  725  698 |    c = b'pop'
  569  831  777 1286  285 |    d = b'latin'
  386  152  613  270 2098 |    e = b'rock'
```

Neural network

For the neural network performance (accuracy/f1/precision/recall/kappa/ROC/PRC) also continued to increase steadily with the number of iterations for the given subspace and continued to increase until 20 iterations where it leveled off.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11140               55.3926 %
Incorrectly Classified Instances      8971               44.6074 %
Kappa statistic                          0.4402
Mean absolute error                      0.266
Root mean squared error                  0.3507
Relative absolute error                 83.2729 %
Root relative squared error             87.7657 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.684    0.126    0.608      0.684    0.644      0.536    0.857     0.705     b'edm'
                 0.741    0.143    0.588      0.741    0.656      0.553    0.877     0.662     b'rap'
                 0.305    0.122    0.387      0.305    0.341      0.201    0.709     0.350     b'pop'
                 0.281    0.067    0.491      0.281    0.358      0.270    0.753     0.414     b'latin'
                 0.734    0.102    0.603      0.734    0.662      0.586    0.915     0.697     b'rock'
Weighted Avg.    0.554    0.114    0.537      0.554    0.535      0.432    0.822     0.569

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 3052  432  538  198  241 |    a = b'edm'
  329 3217  280  321  192 |    b = b'rap'
  847  634 1234  443  886 |    c = b'pop'
  484 1057  773 1054  380 |    d = b'latin'
  310  129  366  131 2583 |    e = b'rock'
```

1.3 second half : Varying the subspace size.
KNN

For KNN k=1 when the number of features was lowered from 100% in iterations of 10% of the total the performance benchmarks continued to increase until they reached a peak at a subspace of about 40%. Further decreasing the subspace size caused the performance to get worse i.e lower accuracy/f1/precision/recall/kappa/ROC/PRC.

KNN k=1 20 iterations subspace 40%

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10037               49.908  %
Incorrectly Classified Instances     10074               50.092  %
Kappa statistic                          0.3715
Mean absolute error                      0.259
Root mean squared error                  0.3713
Relative absolute error                 81.0949 %
Root relative squared error             92.9177 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.621    0.152    0.539      0.621    0.577      0.447    0.786     0.530     b'edm'
                 0.639    0.121    0.593      0.639    0.615      0.505    0.838     0.586     b'rap'
                 0.274    0.156    0.307      0.274    0.289      0.123    0.586     0.251     b'pop'
                 0.343    0.118    0.401      0.343    0.370      0.240    0.681     0.324     b'latin'
                 0.596    0.083    0.604      0.596    0.600      0.516    0.874     0.602     b'rock'
Weighted Avg.    0.499    0.127    0.490      0.499    0.493      0.368    0.753     0.460

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2772  409  707  357  216 |    a = b'edm'
  414 2773  407  571  174 |    b = b'rap'
 1003  510 1108  725  698 |    c = b'pop'
  569  831  777 1286  285 |    d = b'latin'
  386  152  613  270 2098 |    e = b'rock'
```

Decision tree

For the decision tree the performance got worse at lower numbers of features and increased for higher numbers of features until it peaked at a subspace of around 70%. Further increasing it caused the performance benchmarks to worsen.

Decision tree 20 iterations subspace 70%

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11100               55.1937 %
Incorrectly Classified Instances      9011               44.8063 %
Kappa statistic                          0.4385
Mean absolute error                      0.2453
Root mean squared error                  0.3462
Relative absolute error                 76.8027 %
Root relative squared error             86.6369 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.656    0.117    0.615      0.656    0.635      0.527    0.840     0.666     b'edm'
                 0.716    0.118    0.625      0.716    0.667      0.570    0.881     0.664     b'rap'
                 0.273    0.129    0.348      0.273    0.306      0.158    0.657     0.292     b'pop'
                 0.379    0.103    0.457      0.379    0.414      0.297    0.737     0.393     b'latin'
                 0.722    0.093    0.621      0.722    0.668      0.593    0.915     0.684     b'rock'
Weighted Avg.    0.552    0.113    0.535      0.552    0.540      0.431    0.806     0.543

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2925  367  623  333  213 |    a = b'edm'
  285 3105  303  464  182 |    b = b'rap'
  850  549 1106  710  829 |    c = b'pop'
  467  827  705 1422  327 |    d = b'latin'
  228  119  445  185 2542 |    e = b'rock'
```

Neural net

The neural net performance on benchmarks (accuracy / f1 / precision / recall / kappa/ ROC/PRC) worsened when subspace was lowered below 50% i.e 40% , 30% … etc.

When it was increased from 50% the performance it continued to improve all the way till 100% (full set of features).

Neural net , 20 iterations, subspace of 100% (full set)

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11465               57.0086 %
Incorrectly Classified Instances      8646               42.9914 %
Kappa statistic                          0.4612
Mean absolute error                      0.2352
Root mean squared error                  0.3366
Relative absolute error                 73.6467 %
Root relative squared error             84.2266 %
Total Number of Instances            20111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.644    0.101    0.644      0.644    0.644      0.543    0.865     0.700     b'edm'
                 0.717    0.115    0.631      0.717    0.671      0.575    0.885     0.698     b'rap'
                 0.401    0.149    0.404      0.401    0.403      0.253    0.730     0.375     b'pop'
                 0.345    0.084    0.484      0.345    0.403      0.299    0.764     0.452     b'latin'
                 0.729    0.088    0.636      0.729    0.679      0.608    0.921     0.714     b'rock'
Weighted Avg.    0.570    0.109    0.562      0.570    0.563      0.457    0.833     0.590

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2873  379  714  266  229 |    a = b'edm'
  305 3110  352  426  146 |    b = b'rap'
  678  483 1623  521  739 |    c = b'pop'
  396  876  828 1294  354 |    d = b'latin'
  206   81  499  168 2565 |    e = b'rock'
```

1.4

From the lectures….

Bagging is more beneficial for decision trees and possibly for KNN (k=1), as it helps reduce variance and overfitting.

Random SubSpacing could be more beneficial for decision trees, but less so for neural networks and KNN (k=1), due to their dependence on the full feature set for prediction accuracy.

Finding the best strategy for each classifier

I used python to perform a grid search in order to find the optimum mix of subspace size, bag size and number of iterations.

Here is the code I used to find the optimum for Decision tree

```
X_train = df.iloc[:, :-1].values
y_train = df.iloc[:, -1].values
tree = DecisionTreeClassifier()
bagging_tree = BaggingClassifier(base_estimator=tree)
param_grid_tree = {
    'max_samples': [0.2, 0.4, 0.6, 0.8, 1.0],
    'max_features': [0.2, 0.4, 0.6, 0.8, 1.0],
    'n_estimators': [2,4,6,8,10,12,14,16,18,20]}
grid_search_tree = GridSearchCV(bagging_tree, param_grid_tree, cv=5, n_jobs=-1, verbose=2)
grid_search_tree.fit(X_train, y_train)
```

Output

Best parameters for Decision Tree: {'max_features': 1.0, 'max_samples': 0.4, 'n_estimators': 20}
Best score for Decision Tree: 0.5570577459073283

I then used the optimum to generate benchmarks for decision tree

Classification Report:

	precision	recall	f1-score	support
0	0.62	0.65	0.64	4461
1	0.45	0.41	0.43	3748
2	0.35	0.32	0.34	4044
3	0.65	0.69	0.67	4339
4	0.66	0.70	0.68	3519
accuracy			0.56	20111
macro avg	0.55	0.55	0.55	20111
weighted avg	0.55	0.56	0.55	20111

Cohen's Kappa: 0.4445369065318724

Discussion
The findings align with the theoretical expectations, demonstrating the effectiveness of bagging in improving the Decision Tree's performance. The reliance on the full feature set (100% max features) also aligns with the expectation that Decision Trees can handle and benefit from more complex feature spaces.

KNN K=1

Similar code was used for KNN k=1

```
X_train = df.iloc[:, :-1].values
y_train = df.iloc[:, -1].values
knn = KNeighborsClassifier(n_neighbors=1)
bagging_knn = BaggingClassifier(base_estimator=knn)
param_grid_knn = {
    'max_samples': [0.2, 0.4, 0.6, 0.8, 1.0],
    'max_features': [0.2, 0.4, 0.6, 0.8, 1.0],
  'n_estimators': [2,4,5,8,10,12,14,16,18,20]}
grid_search_knn = GridSearchCV(bagging_knn, param_grid_knn, cv=5, n_jobs=-1, verbose=2)
grid_search_knn.fit(X_train, y_train)
```

Output
Best parameters for kNN: {'max_features': 0.6, 'max_samples': 0.4, 'n_estimators': 20}
Best score for kNN: 0.5006214638775821

Classification Report KNN K=1

	precision	recall	f1-score	support
0	0.62	0.65	0.64	4461
1	0.45	0.40	0.42	3748
2	0.36	0.33	0.35	4044
3	0.66	0.70	0.68	4339
4	0.66	0.70	0.68	3519
accuracy			0.56	20111
macro avg	0.55	0.56	0.55	20111
weighted avg	0.55	0.56	0.55	20111

Cohen's Kappa: 0.4489736748565837

Discussion

This result is somewhat in line with the theoretical expectations. While bagging was expected to be beneficial, the improvement in performance was not as significant as it was for the Decision Tree. The reduced feature set (60% max features) indicates some departure from the expectation that kNN relies heavily on the full feature set. This could be due to the specific nature of the dataset or the characteristics of kNN when k=1.

Neural network

```
X_train = df.iloc[:, :-1].values
y_train = df.iloc[:, -1].values
mlp = MLPClassifier(max_iter=1000)
bagging_mlp = BaggingClassifier(base_estimator=mlp)
param_grid_mlp = {
    'max_samples': [0.2, 0.4, 0.6, 0.8, 1.0],
    'max_features': [0.2, 0.4, 0.6, 0.8, 1.0],
    'n_estimators': [20]
}
grid_search_mlp = GridSearchCV(bagging_mlp, param_grid_mlp, cv=5, n_jobs=-1, verbose=2)
grid_search_mlp.fit(X_train, y_train)
```

Best parameters for MLP: {'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 20}
Best score for MLP: 0.5686300497520249
Discussion neural network
These findings align well with the theoretical expectations. The reliance on the full feature set (100% max features) and the full sample set (100% max samples) fits in with the understanding that neural networks benefit from a more extensive data and feature space to model complex relationships effectively.

1.5

Linear regression

```
Linear Regression Model

energy =

      1.8559 * loudness +
      0.1221 * liveness +
      0.1162 * tempo +
     -0.922

Time taken to build model: 0.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.6945
Mean absolute error                      0.0998
Root mean squared error                  0.1257
Relative absolute error                 70.9086 %
Root relative squared error             71.9433 %
Total Number of Instances            20111
```

energy =

1.8559 * loudness +
0.1221 * liveness +
0.1162 * tempo +
-0.922

The linear regression model shows 'loudness' as the most influential predictor for 'energy', with a correlation coefficient of 0.6945 indicating a moderate positive relationship between the predictors and the target. The model has an MAE of 0.0998 and an RMSE of 0.1257, which suggests that predictions are relatively close to the actual values. However, the high relative errors (over 70%) imply that the model's performance is not vastly superior to a basic predictor based on the mean.

SGD

```
energy =

        1.8503 (normalized) loudness
 +      0.1169 (normalized) liveness
 +      0.1094 (normalized) tempo
 -      0.9283

Time taken to build model: 0.65 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.6906
Mean absolute error                      0.1008
Root mean squared error                  0.1265
Relative absolute error                 71.6253 %
Root relative squared error             72.4058 %
Total Number of Instances            20111
```

energy =

1.8503 (normalized) loudness
+ 0.1169 (normalized) liveness
+ 0.1094 (normalized) tempo
- 0.9283

The SGD regression model indicates 'loudness' is the strongest predictor of 'energy'. The model also has moderate prediction accuracy (correlation coefficient of 0.6906). The MAE and RMSE values are low, showing close predictions to actual 'energy' levels, but relative errors over 70% suggest the model's predictions are not substantially better than a simple average based model.

Comparison of Linear regression and SGD

Comparing the Linear Regression and SGD models, both identify 'loudness' as the primary influencing factor for predicting 'energy'. The Linear Regression model has a slightly higher correlation coefficient (0.6945 vs. 0.6906), suggesting it may fit the data marginally better than the SGD model. The MAE and RMSE are very close for both models, indicating similar predictive accuracy. However, both models exhibit high relative errors exceeding 70%, indicating that neither model is a significant improvement over a simple model that would predict the average 'energy' for all tracks. Overall, the performance of both models is comparable with no substantial differences in predictive accuracy.

END OF ASSIGNMENT 2