COMP47460

Assignment 2

Judith Smolenski, 22204071

1.1 - Classifier Performance

Overall the outputs of the three basic classifiers are not too promising in predicting the output class of the song in question. The correctly classified instances are all only about 50% for each of the models with k-NN performing the worst at 48.7025%. In terms of accuracy measures and their values, as well as when considering the confusion matrix of each, the Neural Network performed quite poorly with the TP rate being only around 50% for each of the classes.While accuracy is worth considering overall it can be limited in understanding the performance of each model especially when it comes to misclassification. The correct identification of each genre is important when classifying music genres as listeners will have a strong opinion when something is grouped into the wrong genre. Therefore precision, recall and f-measure are all useful to consider. The Kappa statistic and its taking into account of correct classification achieved by chance is useful in determining an overall performance. The confusion matrix shows us where the model is struggling with classification, and the ROC Are is important to see how good the models are at distinguishing between genres.

The average values in the table below show that none of the models are really ideal as they currently are, and when looking at the specific errors made in the classification of the different music genres misclassification is common between edm and pop on average. Each model also has it's own issues too with k-NN and decision tree models both having trouble with edm and rock, while the neural network misclassified latin and pop more.

	Correct	F-Measure	Kappa	ROC Area
Decision Tree	51.7009%	0.516	0.3954	0.714
Neural Network	52.5218%	0.519	0.4065	0.809
k-NN	48.7025%	0.489	0.3575	0.692

After having applied the Vote ensemble method these methods achieved the most varying results:

	Correct	F-Measure	Kappa	ROC Area
Product (Mid)	50.1605%	0.505	0.3763	0.714
Majority (Best)	55.7766%	0.551	0.4453	0.722
Minimum (Worst)	48.1799%	0.486	0.3506	0.704

Majority voting worked best for this problem, outperforming the others, although the Average of Probabilities was very close behind. It is a simple approach that focuses on the most common prediction. With edm having a majority presence in the dataset it makes sense that majority voting would be a decent enough method to apply here.
The Minimum Probability rule selects the minimum probability assigned by the classifiers. This suggests that conservative prediction is not the best approach.
And the Product rule multiplies the probabilities assigned to classifiers with more weight being assigned to classifiers with higher probability. Given the nature of this rule it makes sense that it would rank in the middle of the best and worst shown above.

1.2 - Ensemble with Bagging

When applying ensemble with bagging all three classifiers from task 1 performed best with the largest ensemble size of 20. Even though the overall performance difference is not large, the models improve on a larger ensemble size. Using the best performing ensemble size of each (in **bold**) changing the bag size percent value had an impact on the accuracy of the models as well. We can see that k-NN benefitted more from a smaller bag size while the Decision Tree and Neural Networks both improve with larger bag size values. Improvements are consistent across the different metrics observed with all values improving together for each best set.

k-NN model results				
Ensemble Size	**Correct**	**F-Measure**	**Kappa**	**ROC**
2	46.7999%	0.471	0.3334	0.709
10	49.2252%	0.494	0.3642	0.750
20	**49.3948%**	**0.496**	**0.3664**	**0.759**
Bag Size Percent				
10	**51.7652%**	**0.518**	**0.3955**	**0.794**
20	51.1921%	0.512	0.3886	0.789
50	50.2063%	0.502	0.3763	0.772
90	48.762%	0.488	0.3584	0.757
100	49.3948%	0.496	0.3664	0.759

Decision Tree model results				
Ensemble Size	**Correct**	**F-Measure**	**Kappa**	**ROC**
2	49.3582%	0.495	0.3658	0.758
10	58.3853%	0.579	0.4791	0.830
12	58.7566%	0.582	0.4837	0.834
18	59.4489%	0.589	0.4925	0.840
20	**59.5865%**	**0.591**	**0.4942**	**0.842**
Bag Size Percent				
10	56.9922%	0.563	0.4618	0.828
90	58.5511%	0.579	0.4815	0.838
100	**59.5865%**	**0.591**	**0.4942**	**0.842**

Neural Network model results				
Ensemble Size	**Correct**	**F-Measure**	**Kappa**	**ROC**
2	53.8973%	0.536	0.423	0.814
10	56.0064%	0.554	0.45	0.827
16	56.5796%	0.561	0.457	0.831
20	**56.6254%**	**0.560**	**0.4577**	**0.832**
Bag Size Percent				
10	55.5938%	0.550	0.4447	0.829
90	56.5337%	0.561	0.4565	0.831
100	**56.6254%**	**0.560**	**0.4577**	**0.832**

1.3 - Ensemble with Subspacing

Altering the subspace sizes for the most effective ensemble size shows the effect that these values have on the predictive ability of each of the models. While the k-NN model is best at a subspace size of 0.5, both the Decision Tree and Neural Network models benefit from larger values of 0.75 and 0.95. Again the metrics improve consistently across the different values observed with no unexpected outliers among them.

k-NN model results				
Ensemble Size	**Correct**	**F-Measure**	**Kappa**	**ROC**
2	42.0823%	0.418	0.2724	0.699
10	52.439%	0.521	0.4039	0.779
20	**54.901%**	**0.544**	**0.4349**	**0.793**
Subspace Size				
0.25	49.5003%	0.488	0.3667	0.755
0.5	**54.901%**	**0.544**	**0.4349**	**0.793**
0.75	53.929%	0.538	0.4228	0.792
Decision Tree model results				
Ensemble Size	**Correct**	**F-Measure**	**Kappa**	**ROC**
2	47.2905%	0.470	0.3401	0.741
10	57.207%	0.564	0.4641	0.818
20	**58.9171%**	**0.579**	**0.4856**	**0.831**
Subspace Size				
0.25	55.254%	0.539	0.439	0.809
0.5	58.9171%	0.579	0.4856	0.831
0.75	**59.6323%**	**0.589**	**0.4947**	**0.841**
Neural Network model results				
Ensemble Size	**Correct**	**F-Measure**	**Kappa**	**ROC**
10	54.7455%	0.535	0.4336	0.825
20	**54.7685%**	**0.532**	**0.4336**	**0.828**
Subspace Size				
0.25	52.9204%	0.506	0.4089	0.809
0.5	54.7685%	0.532	0.4336	0.828
0.75	57.528%	0.563	0.4685	0.838
0.95	**57.7068%**	**0.568**	**0.4707**	**0.838**

1.4 - Benefit and Strategy

Different classification methods can benefit from different ensemble method applications. The effectiveness of these methods depends on a number of factors but generalised rules can be applied. In the lectures we learned that Neural Networks and Decision Trees benefit and perform well in Bagging ensembles, while k-NN classification works well with the Random Subspace method.
The results achieved with the models above show the following best results:

	Decision Tree	**k-NN**	**Neural Network**
Bagging	59.5865%	51.7652%	56.6254%
Random Subspacing	59.6323%	54.901%	57.7068%

We can see here that the differences in performance for the Decision Tree and Neural Network models is not significant or as expected. The Decision Tree model performed extremely marginally better in the random subspacing model, as did the Neural Network. The k-NN model followed prediction more accurately, with the Random Subspacing method outperforming the Bagging one by about 3%.
These results are not in line with the expected optimal ensemble methods as outlined in the lecture slides and there may be a number of reasons for this. The complexity of the starting point model may have an impact in that if the model is already complex, improvement will be small. While noise in the data and an unsuitable large dimensionality can affect performance as well that is not likely to have caused an issue here as there are relatively few features being worked with and the data is clean. The size of the dataset on the other hand may have caused some issues as Bagging typically benefits from larger datasets rather than smaller ones, and while we were working with a substantial set of data, scaling it up might help improve accuracy of the models here. The base models themselves may also be to blame. If the models are highly correlated for example, ensemble methods are not likely to provide much improvement.

1.5 - Correlation and Regression

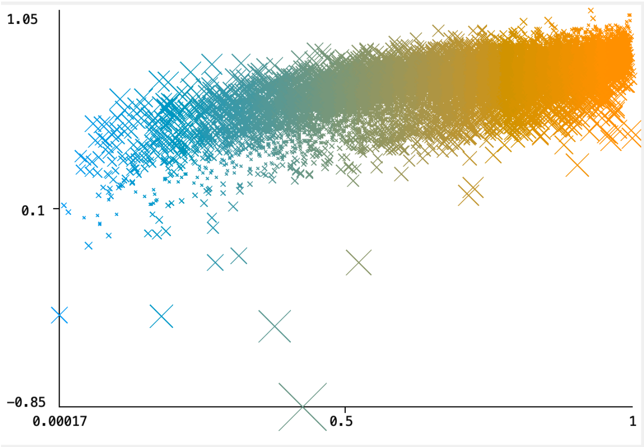
These are the results of building a Linear Regression and SDG for energy using loudness, liveness and tempo:

```
Linear Regression Model

energy =

      0.0387 * loudness +
      0.1189 * liveness +
      0.0005 * tempo +
      0.8834

Time taken to build model: 0.16 seconds

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

Correlation coefficient                  0.6913
Mean absolute error                      0.0996
Root mean squared error                  0.1254
Relative absolute error                 71.2992 %
Root relative squared error             72.2515 %
Total Number of Instances            21812
```

```
Loss function: Squared loss (linear regression)

energy =

         1.8561 (normalized) loudness
 +       0.113  (normalized) liveness
 +       0.1111 (normalized) tempo
 -       0.9118

Time taken to build model: 0.3 seconds

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

Correlation coefficient                  0.6898
Mean absolute error                      0.1001
Root mean squared error                  0.1257
Relative absolute error                 71.6045 %
Root relative squared error             72.4231 %
Total Number of Instances            21812
```

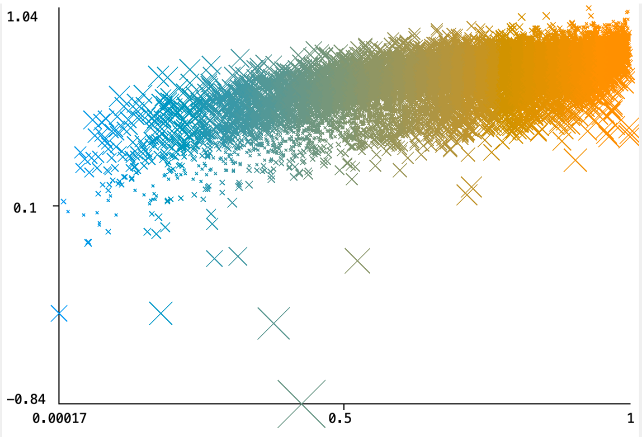
Both of these models have a high correlation coefficient of around 0.69 which indicates a positive linear relationship between the target feature “energy” and the others. The MAE and RMSE are low, suggesting relatively accurate predictions, however the relative absolute and root relative squared errors are quite high in the 70%s which is not great in terms of the reliability of the models.
We can also easily see in the visualisations for each how similarly they performed:

Linear Regression



x: energy, y: predictedenergy

SGD



X: energy, y: predictedenergy