COMP30027 Assignment1 Report
Student IDs: 1214260, 1255109

Task1

Q1.

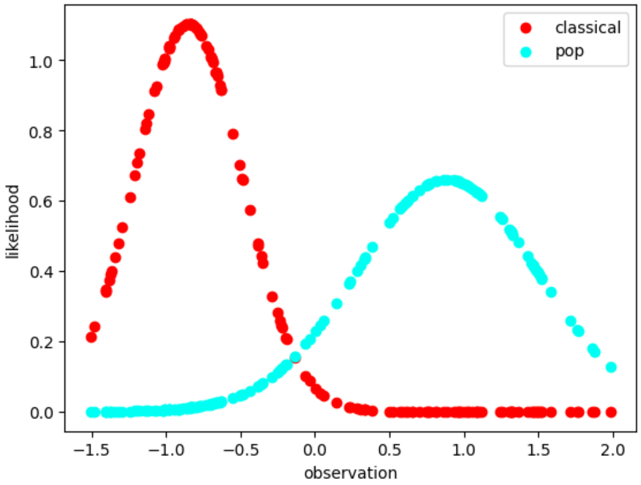
Accuracy	Macro Precision	Macro Recall	Macro F1 Score
0.9767	0.9767	0.9783	0.9762

On performing analysis on the pop vs classical test set by training on the pop vs classical train set we obtain the following performance metrics. Using built-in functions from the scikit-learn library, we evaluated our model.
We used macro precision, macro recall as well as macro F1 score in order to give all classes equal importance in the imbalanced dataset. The accuracy of our model is around 97.67%, the precision is around 97.67%, the recall is around 97.83% and the F1 score is 0.9762.

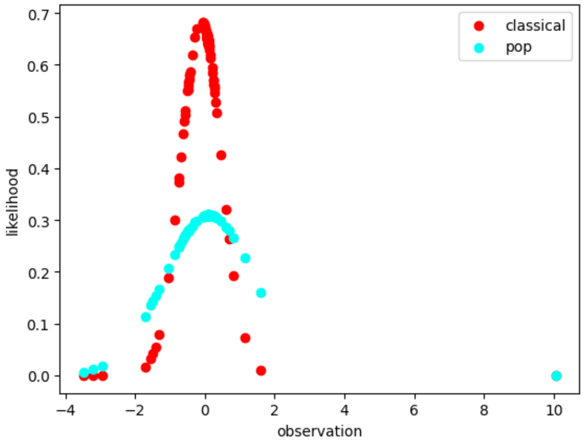
Q2.

Among the three features (spectral_centroid_mean, harmony_mean, and tempo), spectral_centroid_mean appears to be the most suitable feature for differentiating between pop and classical music based on the likelihood plot. This is because features that overlap between classes will not be as useful in distinguishing them.
The likelihood plot reveals that spectral_centroid_mean has two distinct distributions for classical and pop classes, with well-separated probability density functions. In contrast, tempo has similar distributions on the same ranges, and harmony_mean overlaps in their observation ranges. Furthermore, spectral_centroid_mean has no significant outliers, unlike harmony_mean, which has an outlier at scale 10 on the observation axis.
Regarding the size of the data, spectral_centroid_mean is more attractive than the other two features, as it has the most continuous data points on the plot. Harmony_mean has fewer data points, and tempo shows even fewer.
These three reasons can be demonstrated with a simple example. Spectral_centroid_mean is useful on the whole observation scale, whereas

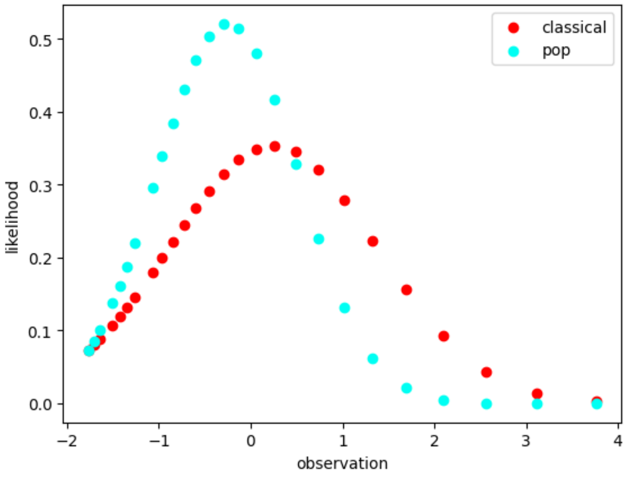
harmony_mean is useless after observation scale 2, and tempo is useless in the beginning and at the end of the scale and not as useful as spectral_centroid_mean. In summary, spectral_centroid_mean has distinct distributions for each class with no significant outliers and significant amount of continuous data points, making it the most appropriate feature to classify pop vs. classical music based on the likelihood plots.



< spectral_centroid_mean.png >



< harmony_mean.png >



< tempo.png >

Task2

Q3.

On performing analysis on the gztan_test set by training on the gztan_train set we obtain the following performance metrics. Overall, the model accuracy is 49.5%.

Gztan test

Accuracy	Macro Precision	Macro Recall	Macro F1 Score
0.495	0.4777	0.5065	0.5018

Decision Tree Accuracy: 0.9988
Our model is structured from the workshop for COMP30027.
Decision tree model is a complex version of the 1R model, with multiple branches instead of a single branch like the 1R baseline. As the decision tree is more complex, it helps in classification of larger complicated datasets thus resulting in higher accuracy.

0R Accuracy: 0.1075
We obtained a model performance of 10.75% (predicting label: Reggae)

The 0R baseline model is a simple model that predicts the most frequent class in the training data. This can equate the accuracy with the greatest prior probability per label. The accuracy therefore suggests that when predicting the most probable label, our model would predict that particular label for 10.75% of the data.

1R Accuracy: 0.1962
Best Attribute: perceptr_var
The likelihood and prior probabilities for each class can be used to implement the 1R model. The accuracy of perceptr_var is the highest and therefore, perceptr_var is the best feature for classification.

The full model has an obviously better performance than the 0R and 1R models, which indicates that the additional features included in the full model provide valuable information to predict the label and, hence affecting the accuracy. It seems that some important relationships between the features that can't be captured by 0R and 1R models can be detected by the full model. The high accuracy of the decision tree also supports the better performance of the full model.

Q4.

Test-size	**Average Accuracy**	**Average Macro Precision**	**Average Macro Recall**	**Average Macro F1 Score**
10%	0.3667	0.2661	0.3253	0.2517
20%	0.5375	0.5094	0.5031	0.4714
30%	0.5619	0.5645	0.556	0.5338
40%	0.5292	0.5302	0.5046	0.4832
50%	0.516	0.5316	0.525	0.4943

60%	0.4875	0.5088	0.491	0.4695
70%	0.5286	0.5047	0.5131	0.4761
80%	0.5375	0.5149	0.5292	0.492
90%	0.3444	0.2472	0.2841	0.2494

(Computation time: 1 minute)
Model performance improves as training set size increases, however beyond a point as train set increases, the model might be overfitting, resulting in lower accuracy. From the table above we can observe that performance (accuracy) increases with increase in training set and then decreases as model overfits. Additionally, we are using 5 fold cross validation for our model evaluation. Another way to improve our model evaluation can be by setting up the experiment for multiple k values under k-fold cross validation to observe what strategy yields higher accuracy. Under k-fold cross validation, we would expect the model to perform significantly better in dealing with overfitting data, thereby explaining only a minor change in accuracy for larger training dataset. As the experiment conducted randomly, for each run our evaluated values would vary. Furthermore, evaluating on the basis of 0R, 1R and decision tree baseline models could also help in identifying the best fitting model for our gztan dataset. From this experiment we can conclude that the gaussian distribution would not yield higher accuracy and we might have to find a better fitting distribution for improved predictions.

Q6.

Delete-size	**Accuracy**	**Macro Precision**	**Macro Recall**	**Macro F1 Score**
0%	0.495	0.4777	0.5065	0.5018
10%	0.485	0.4672	0.4914	0.4926
20%	0.51	0.4934	0.5067	0.5067
30%	0.505	0.4835	0.5152	0.5049

40%	0.495	0.4826	0.4923	0.5063
50%	0.53	0.5108	0.5357	0.5471
60%	0.515	0.4955	0.522	0.5041
70%	0.555	0.5592	0.5683	0.5691
80%	0.515	0.4897	0.5168	0.489
90%	0.11	0.0198	0.1	0.011

(Computation time: 2 minutes 30 seconds)
A function was created to delete data randomly from the training set. While theoretically as missing data increases the accuracy of the model should decrease. However, from our experiment we can observe that as data deletion increases from 0% to 60% in equal intervals of 10% the accuracy of our model improves and as data deletion increases from 60% to 90% the accuracy decreases. As the model accuracy for the full dataset is low, we can deduce that the training set is overfitting the model. Which would subsequently explain the better performance of the model as more data is deleted. Additionally, it is important to consider that we are using the mean and standard deviations from our training set to predict the labels of our test set. As data deletion increases and the process is random, some features might have largely different means and/or standard deviations which would account towards lower accuracy. Especially, considering data deletion at 90%, there are some features with all data missing thus resulting in even lower accuracy. It is important to note that as this experiment is conducted randomly, the accuracy would alter slightly each time. From this we can appreciate that we were overfitting the model initially, highest accuracy was observed at 60% data deletion and as we delete more data, we are underfitting the model.