A Hybrid Approach for Music Classification and Recommendation through Genre Feedback

---

Mashaole Masekwameng

Supervisor:
Dr. Ritesh Ajoodha

A research report submitted in partial fulfilment of the requirements for the degree of BScHons Big Data Analytics

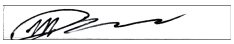
in the

School of Computer Science and Applied Mathematics
University of the Witwatersrand, Johannesburg

28 November 2021

Declaration

I, Mashaole Masekwameng, declare that this research report is my own, unaided work. It is being submitted for the degree of BScHons Big Data Analytics at the University of the Witwatersrand, Johannesburg. It has not been submitted for any degree or examination at any other university.



Mashaole Masekwameng

28 November 2021

Acknowledgements

I would like to thank my supervisor Dr. Ritesh Ajoodha for the extensive guidance and resources which allowed for the completion of this research report.

A Hybrid Approach for Music Classification and Recommendation through Genre Feedback

Mashaole Masekwameng
School of Computer Science and Applied Mathematics
University of the Witwatersrand, Johannesburg
Johannesburg, South Africa
686390@students.wits.ac.za

Dr. Ritesh Ajoodha
School of Computer Science and Applied Mathematics
University of the Witwatersrand, Johannesburg
Johannesburg, South Africa
Ritesh.Ajoodha@wits.ac.za

***Abstract*—Automatic music classification relies primarily on traditional aspects of music encapsulated by content-based features. This means cultural aspects of music classification are not incorporated into the classification process. In order to improve classification, a hybrid method of using feedback from users is proposed to incorporate these cultural aspects. This paper proposes a pipeline of classifying music by genre through the use of content-based features followed by the implementation of Q-learning. The GTZAN dataset is used to model four content-based classifiers, namely: Logistic Regression, Support Vector Machines, Multilayer Perceptron and Random Forests. The highest performing model was the Support Vector Machine (SVM) with and accuracy score of 75.50% based on a 10-fold confusion matrix, in line with other experiments in this field of research [3]. Q-learning is used to further refine the predictions of the SVM by using user input to develop a Q-table with Q-values for each piece of music, associating it with a certain genre based on user feedback. This resulted in an accuracy score of 75.50%, similar to that of the SVM classification, indicating that Q-learning accuracy cannot overcome the initial classification introduced by the SVM predicted values.**

***Index Terms*—music genre classification, q-learning, recommendation, content-based features, support vector machines**

I. Introduction

Music classification is an area of research that has had many approaches and attempts at finding suitable solutions to automatic classification without human intervention. We live in a world where music streaming services use large databases to store millions of pieces of music for users to listen to [3]. Thus, for users to find the right assortment of music based on their tastes, it's common for music to be recommended by genre. Furthermore, hand-crafted labelling techniques [2] have now become impractical with the shear amount of pieces of music hosted by these services.

Automatic music genre classification overcomes these issues by using content-based features to classify pieces of music in a more objectively by making use of features inherent to the song itself.This form of labelling removes issues related to the subjectivity of genre classification when it is done by humans , making the search of a ground truth of a piece of music's genre less elusive [3]. This classification is important for the recommendation of music based on the genres a user has a preference with.

The work by Ajoodha et al. [3] and Nkambule and Ajoodha [2] indicate that the application of automatic labelling of pieces of music results in highly accurate predictions (81% and 80.8% respectively). De Mulder et al. [4] also showed that reinforcement learning is also effective at mood classification of music by using the feedback from user after a recommendation is made. The recommendation system in Wang et al. also allows users to give indications on whether or not the recommended music is related to the genre requested [6], further fine tuning auto-classification.

The aim of this paper seeks to explore how using well known and effective automatic music classification methods can be further improved by using Q-learning, another reinforcement learning technique. This will allow for the recommendation music to a user and have them provide feedback on whether or not the piece of music fits in with the genre they requested.

Four music genre classification models are implemented using the GTZAN dataset, namely: Naïve Bayes, Support vector machines, Multi-layer perceptron, Linear logistic regression, K-Nearest neighbours and Random forests. The most accurate of these classifiers is used to create initial values for the Q-table used by the Q-learning algorithm in order to bias the learning towards what was discovered by the classifier. Further updates to the Q-table will be made through the feedback provided by the user on the accuracy of the recommendation made according to the user requested genre.

Out of the four classifiers used, the Support Vector Machine (SVM) obtained the highest accuracy score of 75.5% when using a 10-fold confusion matrix as the evaluation metric. The predictions made by the SVM are used to bias the Q-table of the Q-learning algorithm and after 1000000 iterations the 75.5% accuracy score was maintained.

While no increase in the accuracy was obtained, the experiments contained in this paper indicate that the use of a hybrid method for recommendation using Q-learning can, with some success, recommend relevant pieces of music. The contributions of this research are as follows:

- Produced models with high classification accuracy using only the mean and standard deviations for feature representation.

- Produced a reliable recommendation engine based on Q-learning reinforcement learning

In the next section, the work explored in the compilation of this research is expanded upon including how the work is interrelated and how different approaches were taken to achieve classification and recommendation.

II. Related Work

Music recommendation is an area of study that has received immense attention due to the prevalence of online music stores and streaming services that allow users to play songs stored on the cloud. Genre classification allows users to easily find music that suits their taste, while recommendation systems come with the convenience factor of recommending music when the user is not quite sure what they would like to listen to. This opens the door for users to explore new pieces of music based on their listening preferences [6].

A. Data

Currently in content-based approaches to music classification, the magnitude spectrum of a piece of music is used to find commonalities between genres of music based on the features present in the magnitude spectrum [3] [2] [4]. In Ajoodha et al. [3] and Nkambule and Ajoodha [2], content-based features are extracted from pieces of music in the GTZAN dataset in order to use them as predictors for music genre, similar to what is done in this paper.

In Wang et al. [6], 1000 30 second snippets are also used to to develop the models in the paper, in contrast however the music is obtained from YouTube videos evenly spread across the 10 genres. Lippens et al. [4] uses an altered MAMI dataset, which includes 160 full length pieces of music that come from 6 genres instead of the original 11. The distribution of music in the 6 genres is as follows: 24 classical, 18 dance, 69 pop, 8 rap, 25 rock and 16 other tracks.

Two papers use user feedback to refine these labels by using reinforcement techniques to classify based on genre in Wang et al. [6] and by mood in Stockholm and Pasquier [5].

B. Features

Music classification typically makes use of four categories of content-based features in order for the models to differentiate between the genres, namely magnitude-based features, tempo-based features, pitch-based features and chordal progressions [3]. Adjoodha et al. [3] and Nkambule and Ajoodha [2] make use of all the categories named above to classify music into it genres, while Wang et al. [6] and Lippens et al. [4] extracted only magnitude-based features to conduct modelling.

In Ajoodha et al. [3] along with [4], the features extracted are represented as MFCC aggregations, area moments and feature histograms. Wang et al. [6] and Nkambule and Ajoodha [2] make use of more aggregated feature representation, which includes the mean and standard deviation of the time-series data used.

In order to optimise the number of features used to train the models used, Ajoodha et al. [3] and Nkambule and Ajoodha [2] make use of information gain ranking to reduce the dimensionality of the training data. Wang et al. [6] instead uses principal component analysis to reduce feature dimensionality [6].

C. Models

In implementing automatic content-based genre classification, Ajoodha et al. [3] and Nkambule and Ajoodha [2] make use of six classifiers, namely: Naïve Bayes, Support vector machines, Multi-layer perceptron, Linear logistic regression, K-Nearest neighbours and Random forests. Support vector machines were left as future work in the classification work done by Lippens et al. [4], which instead chose to use Gaussian classifier, Gaussian mixture model, Iterative expectation maximization and K-Nearest neighbours.

The Q-learning implementation in Stockholm and Pasquier [5] is based on the Q-value update rule

$$Q_{(j)}(x,y) = Q_{(j)}(x,y) + \alpha R \tag{1}$$

where α is the learning rate and R is the reward obtained from user feedback. Gibb's measure is used to implement a Softmax selection policy in the paper with the following equation

$$\frac{e^{Q_{(j)}(x,y)/T}}{\sum_{x,y} e^{Q_{(j)}(x,y)/T}} \tag{2}$$

Wang et al. [6] opted to use multi-arm bandit approach to implements its reinforcement learning based recommender system. Both user feedback and the novelty of the piece of music is given by the following equation:

$$U = U_c U_n = \bar{\theta}' \mathbf{x}(1 - e^{t/s}) \tag{3}$$

D. Evaluation

The predictively accuracy of the models used in Ajoodha et al. [3] and Nkambule and Ajoodha [2] was evaluated using confusion matrices. The probabilistic classifiers that obtained the most success in the two papers were Support Vector Machines with classification accuracy of 80.8% [2] and Linear Logistic Regression with a classification accuracy of 81% [3]. Other well known results are referenced in table 1, obtained from [3].

Benetos and Kotropoulos (2008)	75.00%
Bergstra et al. (2006)	82.5%
Holzapfel and Stylianou (2008)	74.0%
Li et al. (2003)	79.7%
Lidy et al. (2007)	76.8%
Panagakis et al. (2008)	78.2%
Sturm (2013)	83.0%
Tzanetakis and Cook (2002)	61.0%
Ajoodha et al (2015)	81%
Nkambule and Ajoodha	80.8 %

TABLE I
OTHER WELL KNOWN MUSIC GENRE CLASSIFICATION RESULTS USING THE GTZAN DATASET

Along with confusion matrices, Lippens et al. [4] made use of a lower and upper bound classification consisting of

a random classification and human classification respectively. The lower bound model achieved an accuracy of 58% while the upper bound classification achieved an accuracy of 90%.

Reinforcement learning techniques use the feedback from users to improve the recommendation of music based on the requested genre. The two techniques explored are Multi-arm bandit with user genre preferences [6] and in the case of Stockholm and Pasquier [5], Q-learning based on the mood of a piece of music. In this report, emphasis is placed on using Q-learning to get a q-table that gives an indication of how high a user rates a piece of music into one of the 10 genres present in the GTZAN dataset.

III. Feature Analysis

Since content-based features are used to classify the pieces of music in th GTZAN dataset into their respective genres, time-series features are extracted from the piece of music itself in order to apply modelling [3] [6]. Features related to pieces of music broadly fall under four categories [3] [2], namely:

- Magnitude-based features: timbral features that represent loudness of the piece of music by analysing its magnitude spectrum to find signal changes, noisiness and other spectral features.
- Tempo-based features: represents rhythm present in the piece of music by calculating the spectral energy (root-mean-square) of its signal
- Pitch-based features: frequency of sounds present in the piece of music related to the different scales of pitch used in specific genres
- Chordal progression features: use of the 12 component design matrix (chroma) where each component represents the intensity of a semi-tone in order to distinguish chords present in the piece of music

A. Data

The dataset used in this report is the GTZAN dataset that comprises of 1000 pieces of music, with a length of 30 secs each. Each piece of music has a label based on one of the 10 genres of music it belongs to. This dataset has been widely used in building music classification models that have high levels of accuracy [2] [3].

B. Features

The specific GTZAN dataset used contains 57 features that each fall under one of the four categories of music related features. Each feature contains two dimensions, the mean of the feature across the 30 seconds of playtime, and the standard deviation of the feature across playtime. These aggregations are used to predict the genre each piece of music belongs to. Table 2 indicates the measure used to represent the feature and the category it falls under.

All the features were preprocessed using feature scaling and feature normalization before any modelling was done. Feature scaling is used to give uniformity to the range of values in the features by forcing them to have zero mean and unit variance. The equation used in feature scaling is

$$X' = \frac{X - \mu}{\sigma} \tag{4}$$

where μ is the mean and σ is the variance.

Feature normalization is used to scale and shift the values of the feature in order to ensure that they fall within the range of 0 and 1. The equation used for feature scaling is

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5}$$

Category	Feature	Representation	Dimensions
Magnitude-based			
	Spectal-centroid	mean+variance	2
	Spectal-bandwidth	mean+variance	2
	Spectal-rolloff	mean+variance	2
	MFCC 1-20	mean+variance	2
Tempo-based			
	Tempo	single value	1
	Energy	mean+variance	2
Pitch-based			
	Zero crossing rate	mean+variance	2
Chordal progression			
	Zero crossing rate	mean+variance	2
	Harmony	mean+variance	2

TABLE II
FEATURES AND THEIR REPRESENTATION IN THE GTZAN DATASET

C. Feature Selection

To ensure the models implemented are highly performant as well as generalizable, feature selection is applied in order to remove features that are most informative and leave out those that are either redundant or do not contribute to the accuracy of the model. Embedded feature selection ranks said features using their impurity-based importances. This is done by using a forest of trees classifier to calculate these feature importances. Figure 1 shows the sorted values obtained for the feature importances.

In Figure 1, the chroma_sftf_mean feature has the highest importance however, a lot of the features carry similar values of importance. This leads to the decision to include all the features listed in section C, which will be shown later to be beneficial to the accuracy of the model without hampering performance.

IV. Automatic Music Genre Classification

In this section, the features listed above are used to train four classifiers that will classify each piece of music of the GTZAN dataset into one of ten genres. Predictions made by the classifier with the highest accuracy will be used as a starting point for the Q-learning reinforcement algorithm to further improve its accuracy based on a q-table learned from user feedback. The four classifiers were chosen from [3] and [2] because they were the most performant.

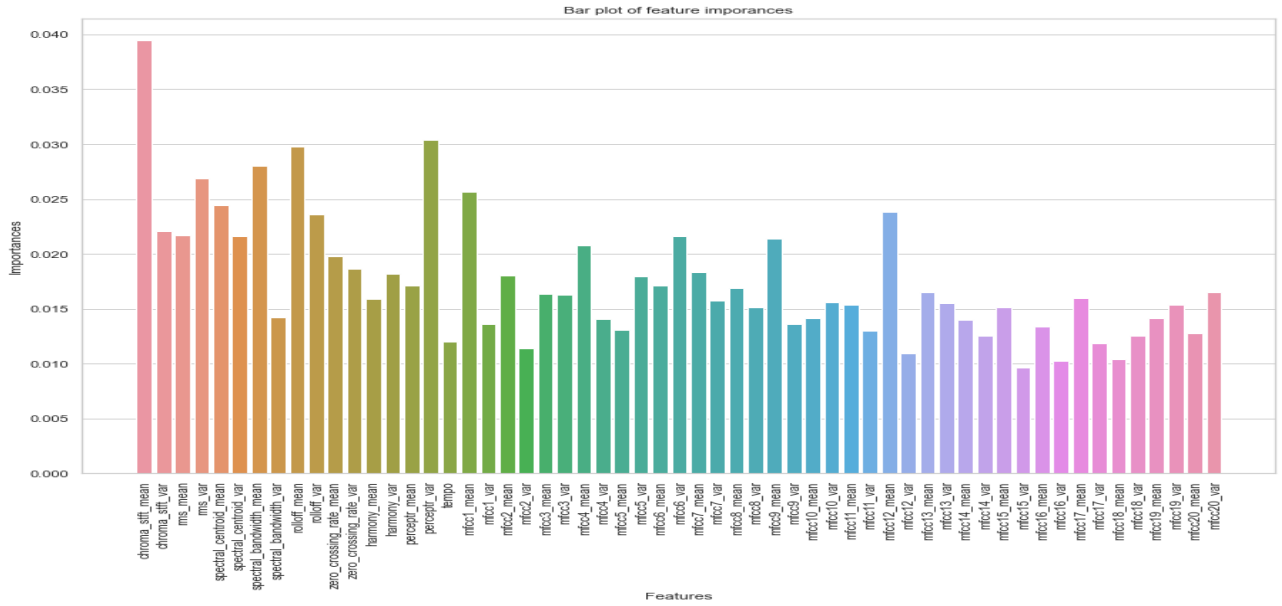


Fig. 1. Figure representing feature importance values using tree-based feature selection

These off-the-shelf models include: Logistic Regression, Support Vector Machines, Multilayer Perceptron and Random Forests. The results of this experiment are contained in table 3, where the accuracy for each model is based on the use of a 10-fold confusion matrix to evaluate the accuracy of the model predictions. The values in table 3 are averages obtained for the prediction accuracy across the 10 genres contained in the GTZAN dataset.

From table 3 the Support Vector is a marginal winner in terms of accuracy with a value of 75.5%, narrowly beating out the Multilayer Perceptron with an accuracy value of 74%. However, when one looks at the time taken for the Multilayer Perceptron to achieve that second place result, it is clear that the Support Vector Machine wins on that metric handily with a Time to Fit that is 40× faster.

The Logistic Regression and Random Forest classifiers are also close to one another in terms of accuracy, with 65.5% and 66.5% respectively. However, we also observe a gap in the Time to Fit model metric as the Logistic Regression model lags behind the Random Forests model by 3× the time taken to fit the model.

To illustrate how these accuracy values are calculated, the 10-fold confusion matrix of the Support Vector Machine model is show in figure 2. Each true label for a piece of music is compared to that of the predicted label across the entire data set. The accuracies for each genre label is aggregated into an average associated with each model.

As mentioned above, feature selection was not used in this report due to loss of accuracy performance when applied to the features used here. In table 4, the same models are run after tree-based feature selection is implemented.

There is a clear benefit in the Time to Fit Model metric across the board for the models, however this is at the detriment of the accuracy as all models dropped in that metric. This means feature selection should only be done when features have high dimensionality that makes them impractical to fit without reducing the number of features used.

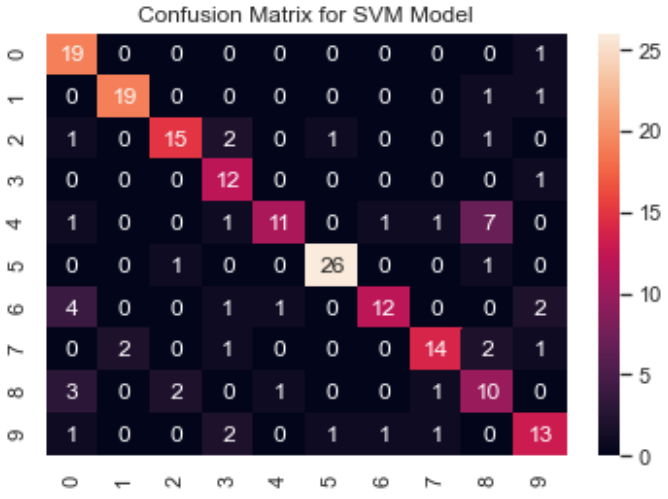


Fig. 2. Figure representing SVM confusion matrix where the bottom axis represents the true genre labels for the pieces of music, while the left hand axis represents the genre classifications made by the SVM model.

V. Music Recommendation Engine

From the above classifiers, the obvious choice to use for the music recommender is the Support Vector Machine. Using the predicted genre classifications, a Q-learning algorithm will be used to further refine these recommendations for increased accuracy of the recommender.

Classifier	Model	Accuracy	Time to Fit Model (seconds)
Logistic Regression	$log(\frac{p}{1-p})$	65.50%	1.2135
Support Vector Machine	$\frac{1}{n}\Sigma_{i=1}^{n} max(0, 1 - y_i(\bar{w}^T \bar{x}_i - b)) + \lambda \|\|\bar{w}\|\|^2$	75.50%	0.1475
Multilayer Perceptron	$(\Sigma_{i=1}^{m} \bar{W} \times \bar{X}) + b$	74%	5.9570
Random Forests	$Gini = 1 - \Sigma_{i=1}^{C}(p_i)^2$	66.50%	0.4024

TABLE III
METRICS OF FOUR CLASSIFIERS

Classifier	Parameters
Logistic Regression	default parameters
Support Vector Machine	C=3, max_iter=500
Multilayer Perceptron	max_iter=850
Random Forests	max_depth=100

TABLE IV
PARAMETERS FOR EACH MODEL USED WITH SCIKIT-LEARN FUNCTIONS

Classifier	Accuracy	Time to Fit Model (seconds)
Logistic Regression	60.50%	0.0382
Support Vector Machine	69%	0.0468
Multilayer Perceptron	69%	5.4346
Random Forests	64%	0.3346

TABLE V
METRICS OF FOUR CLASSIFIERS WITH FEATURE SELECTION CHOOSING BEST 23 FEATURES

A. Q-learning

Q-learning is a popular implementation of reinforcement learning where a Q-table that maps a state to an action is developed where the values in the Q-table (Q-value) represent the responses from the user [5]. When a piece of music is recommended to a user based on their response when prompted to choose one of the 10 genres in the GTZAN dataset, the user is given a chance to provide feedback on the recommendation. A 'Yes' response means the recommended piece of music fits in with the genre, whereas a 'No' response means the piece of music does not fit into the genre selected by the user [5]

The Q-value associated with a piece of music belonging to one of the 10 genres is updated based on the response from the user. The reward value is set to 1 with a Yes response, while a -1 reward value is set with a No response. The update rule for the Q-value is given by equation 6 [4],

$$Q(state, action) = Q(state, action) + \alpha R \tag{6}$$

where state represents the genre the user wants to be recommended while action is one of the songs the recommender engine will recommend to the user. In order to balance the requirements of exploitation (recommend piece of music with highest Q-value) and exploration (recommend a new song the user might like), some random choice is implemented in order for the engine to explore more of the action space [6].

Typically when implementing a Q-learning model, the initial values of the Q-table are small random values used to choose actions for the actor to take. In the music recommendation engine, the predictions made by the Support Vector Machine for a subset of music pieces are used to initialize the q-table instead.

The Q-value associated with the SVM predicted genre label of the piece of music is initialized with a value of 1 to give it a higher probability of being recommended for that genre. This value will be updated using the update rule in equation 6 to increase its recommendation probability further, or decrease it if it was incorrectly labelled.

The algorithm allows for the exploration of pieces of music in the list of songs used for testing the SVM classifier. This is is achieved by defining an ϵ variable, with a value of 0.6, which is compared with a generated random value between 0 and 1. If the random value is less than ϵ, a random piece of music within the requested genre is recommended to the user (exploration). If however, ϵ is smaller than the random value, the piece of music with the highest Q-value within the requested genre is recommended instead (exploitation).

As we develop our strategy, we have less need of exploration and more exploitation to get more utility from our policy. Hence why as the number of iterations increases, ϵ decreases by 0.0001 for the algorithm to favour exploitation over exploration. A total of 1000000 iterations of the algorithm are execute, with the ϵ value decreased every 100 iterations. Execution of the algorithm took a total of 821.2443 seconds (13.7 minutes) to complete after the 1000000 iterations.

B. Evaluation

Reinforcement learning is based on the algorithm exploiting its priors to obtain as many rewards as possible. After each iteration, the rewards are accumulated in order to evaluate their trend over the iterations. If the recommendation engine is learning which piece of music belongs to which genre, then it's expected that this plot will show steady increases in the reward over the iterations.

In order to evaluate the Q-learning implementation, the algorithm loops through all genres in each iteration and queries the recommender engine for a piece of music in that genre. Rewards of 1 are given when the recommendation fits the genre and -1 otherwise.

In Figure 3, the cumulative rewards increase overtime, indicating that the recommendations made by the recommendation engine match the user specified genre. The Q-table at the end of the algorithm is used to evaluate the accuracy of the genre labelling through a confusion matrix. This results in an accuracy score of 75.5%, similar to that of the Support Vector Machine, indicating no material increases in the accuracy

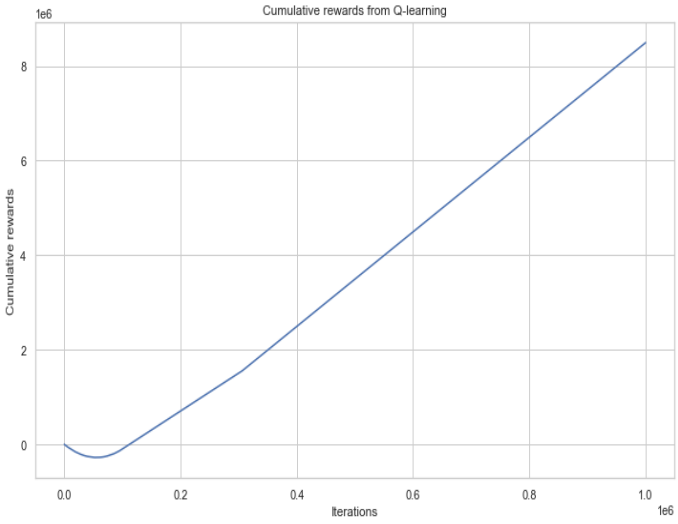


Fig. 3. Figure representing cumulative rewards over time

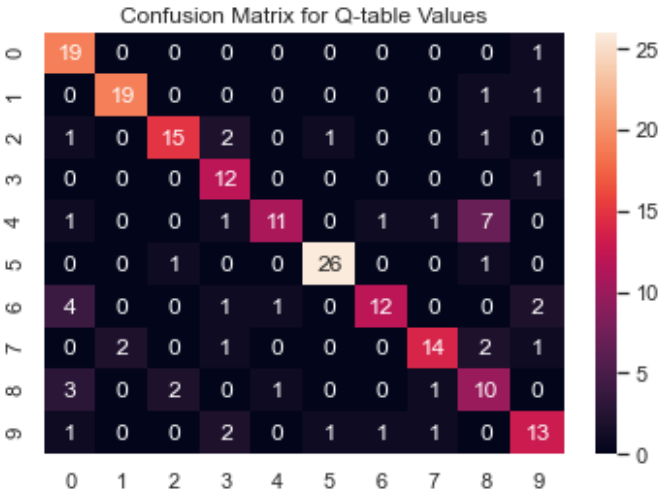


Fig. 4. Figure representing confusion matrix based on each piece of music's maximal values in the learned Q-table. These values are based on user feedback regarding the accuracy of its genre classification.

despite using user feedback on the recommendations. The confusion matrix in figure 4 indicates just how similar the results are to those of the SVM in figure 2.

VI. Conclusion and Recommendations

The results obtained from this experiment indicate that there is a lot of potential in using automatic classification of pieces of music based on their genre. This is an area of study that has been widely explored but it still shows that room for improvement is possible when such accuracy in prediction is obtained from using an aggregated feature set of pieces of music with short length. However, in this experiment lower accuracy scores were obtained precisely due to the aggregated nature of the features used in the models.

While Ajoodha et al. [3] obtained an accuracy high of 81.00% from the Logistic Regression classifier in their report, here the highest preforming model (Support Vector Machine) managed to only obtain 75.5%. While it is not far off, it shows that using hand crafted features that are not aggregated allows for much greater accuracy results at the cost convenience of modelling.

The same can be said for Nkambule and Ajoodha [2], where their highest performing obtained accuracies of 80.80% when also using a Support Vector Machine. Newer algorithms like XgBoost might do well here as well, since they are classifiers that rely on decision trees, perfect for this use case.

Love and Ajoodha [1] propose the use of ontologies, which extract influences between objects, to find relationships between the genres since pieces of music typically straddle between genres. This would allow for the creation of a network of genre classifications for a piece of music, allowing it to have probabilities being in each class rather than a single classification.

The use of the SVM predictions to initialize the Q-table of the Q-learning algorithm before execution resulted in accuracy scores exactly the same as that of the SVM itself (75.5%) after 1000000 iterations. This indicates that despite the algorithm trying to alter the genres based on user feedback, it is not enough to overcome the initial biasing introduced to the Q-table.

Reinforcement learning performance is usually dependent on the number of iterations used to train the agent, and in the case with only 1000000 iterations it took nearly 14 minutes to complete execution. By using a higher number of iterations, it could be possible to increase the accuracy score of the genre labelling of the pieces of music [5].

References

[1] T. Love & R. Ajoodha, "Building Undirected Infuence Ontologies using Pairwise Similarity Functions", In: 2020 International SUAPEC/RobMech/PRASAConference, 2019

[2] T. Nkambule & R. Ajoodha,"Classication of Music By Genre using Probabilistic Models and Deep Learning Models", In: Sixth International Congress on Information and Communication Technology (6th ICICT), 2021.

[3] R. Ajoodha, R. Klein and B. Rosman. "Single-labelled Music Genre Classification using Content-based Features", In: 2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech), 2015.

[4] T. De Mulder, S. Lippens JP. Martens and G. Tzanetakis. "A Comparison of Human and Automatic Musical Genre Classification". In: Acoustics, Speech and Signal Processing 1988 International Conference, 2004.

[5] J. Stockholm & P. Pasquier. "Reinforcement Learning of Listener Response for Mood Classification of Audio", In: International Conference on Computational Science and Engineering CSE, 2009.

[6] D. Hsu, X. Wang, Yi Wang & Ye Wang. "Exploration in Interactive Personalised Music Recommendation: A Reinforcement Learning Approach", In: ACM Transactions on Multimedia and Applications (TOMM), 2014.