Model-based Machine Learning (42186)

Music Genre Classification using Model-Based Machine Learning

DTU

Technical
University of
Denmark

May 2023

1 Abstract

Prediction of music genre represents an interesting modelling problem. This study aims to investigate how well logistic regression and feed forward neural networks can capture the complexity of a music genre dataset[1] with 11 different genre classes. The data was processed and analyzed ,missing values were handled. Ten different experiments were carried, during which five models were trained and tested on the given dataset. Inference was performed using Markov Chain Monte Carlo (MCMC) and Variational Inference (VI) and the results were compared. A brief discussion of the limitation of the experimental choices was discussed and future work considerations were presented.

2 Background

Our study aims to develop a model-based machine learning approach to predict the music genre of each music track by leveraging features such as 'Popularity,' 'danceability,', 'energy' etc. To achieve this goal, we aim to develop an accurate genre classification model. The main objective of this report is to explore the potential of model-based machine learning in predicting music genres and develop robust models capable of automatically assigning genre labels to music tracks, thereby enhancing music recommendation systems and aiding in music discovery.

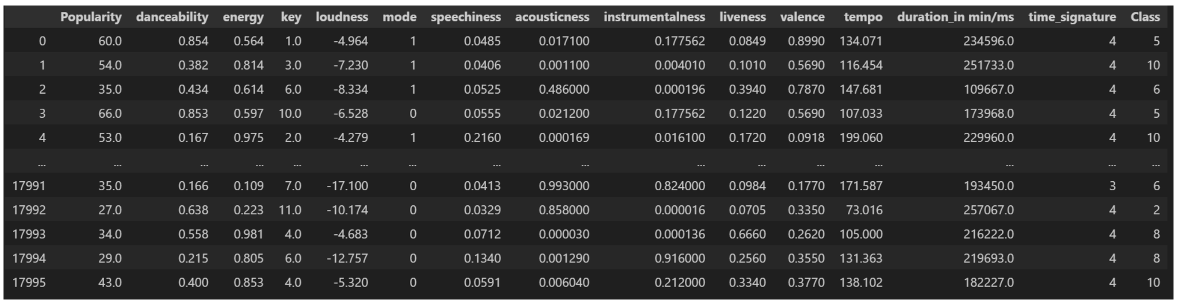
In particular, we conducted multiple experiments within a model-based machine learning framework to analyze the performance of prediction in a classification task. We utilized logistic regression models with different choices of priors, as well as ancestral sampling, neural networks, and Markov Chain Monte Carlo (MCMC) inference algorithms. These approaches helped us approximate the true distribution of the target variable, which, in our case, is the music genre. By evaluating different algorithms and performing extensive data analysis, we aimed to develop accurate models that can effectively predict music genres and provide valuable insights.

3 Data Description

The dataset used for this analysis comprises various attributes related to music tracks from diverse genres. The dataset includes information such as artist name, track name, popularity, acousticness, danceability, duration, energy, instrumentalness, key, loudness, liveness, valence, tempo, and time signature. The artist name and track name columns provide identification details for each track, allowing us to differentiate between different musical works. The popularity column indicates the relative popularity of each track, ranging from 0 to 100, with higher values indicating higher popularity. The acousticness attribute represents the degree to which a track is acoustic, measured on a scale from 0 to 1, where 1 indicates high acousticness. The danceability column quantifies the suitability of a track for dancing, with higher values indicating a greater tendency for danceability. The duration column specifies the length of each track in milliseconds, providing an insight into the temporal aspect of the music. The energy attribute represents the intensity and activity level of a track, with higher values indicating higher energy. The instrumentalness column measures the likelihood of a track being instrumental, ranging from 0 to 1, where 1 signifies a high likelihood of being instrumental. Additionally, the dataset includes the key attribute, which represents the key of each track, providing information about the musical pitch center or tonal center. The loudness attribute measures the overall loudness of the track in decibels (dB), indicating its volume. The liveness column signifies the probability of the track being performed live, ranging from 0 to 1, where higher values suggest a greater likelihood of being a live recording. The valence attribute measures the musical positiveness or happiness of a track, ranging from 0 to 1, with higher values representing more positive or happier tracks. The tempo column specifies the beats per minute (BPM) of each track, providing information about the track's speed or tempo. Lastly, the time signature attribute denotes the number of beats in each bar or measure of a track. These attributes collectively offer a comprehensive and accurate description of the considered data.

4 Data Selection

First, looking into the variables of the original dataset, Artist Name, and Track Name are text attributes. They could only be used if further feature engineering would have been applied in order to extract relevant features from it, which lead to the decision of eliminating them from the final dataset. The rest of the attributes except the two already mentioned are numerical attributes, and they were used as features in the final curated dataset which was used to train, validated and test the models selected. The class attribute is selected as the target attribute. In conclusion, the dataset has the shape of (17996,15), where 17996 corresponds to the number of observations and 15 to the number of features.



	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_in min/ms	time_signature	Class
0	60.0	0.854	0.564	1.0	-4.964	1	0.0485	0.017100	0.177562	0.0849	0.8990	134.071	234596.0	4	5
1	54.0	0.382	0.814	3.0	-7.230	1	0.0406	0.001100	0.004010	0.1010	0.5690	116.454	251733.0	4	10
2	35.0	0.434	0.614	6.0	-8.334	1	0.0525	0.486000	0.000196	0.3940	0.7870	147.681	109667.0	4	6
3	66.0	0.853	0.597	10.0	-6.528	0	0.0555	0.021200	0.177562	0.1220	0.5690	107.033	173968.0	4	5
4	53.0	0.167	0.975	2.0	-4.279	1	0.2160	0.000169	0.016100	0.1720	0.0918	199.060	229960.0	4	10
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
17991	35.0	0.166	0.109	7.0	-17.100	0	0.0413	0.993000	0.824000	0.0984	0.1770	171.587	193450.0	3	6
17992	27.0	0.638	0.223	11.0	-10.174	0	0.0329	0.858000	0.000016	0.0705	0.3350	73.016	257067.0	4	2
17993	34.0	0.558	0.981	4.0	-4.683	0	0.0712	0.000030	0.000136	0.6660	0.2620	105.000	216222.0	4	8
17994	29.0	0.215	0.805	6.0	-12.757	0	0.1340	0.001290	0.916000	0.2560	0.3550	131.363	219693.0	4	8
17995	43.0	0.400	0.853	4.0	-5.320	0	0.0591	0.006040	0.212000	0.3340	0.3770	138.102	182227.0	4	10

Figure 1: Preprocessed dataset

Next, the dataset in the dataframe 1 has been scaled by using StandardScaler for scaling all variables in the same range except class. The target variable class is ranged from 0 to 10, and this should not be scaled because the prediction of the algorithm would be the value of the class and it has to be checked if it is identical to the ground truth of the class, and base on that, the error rate and the accuracy could be calculated. Among the scaled observations, a certain subset was used for training. For getting the best prior and hyperparameter in logistic regression which is the main model in the project, a validation set was made by further spliting the training dataset. Technically, the validation set is used for choosing the best parameters by comparing the different validation errors. The models were then tested on the test dataset that was set aside.

5 Data Analysis And Modelling

5.0.1 Data Analysis

As a first step of analysis of the data if we take a look at the distribution of each feature in figure 2 we can see that most of the features in our data are normally distributed among their values. The rest of the features are just following a frequency count distribution.

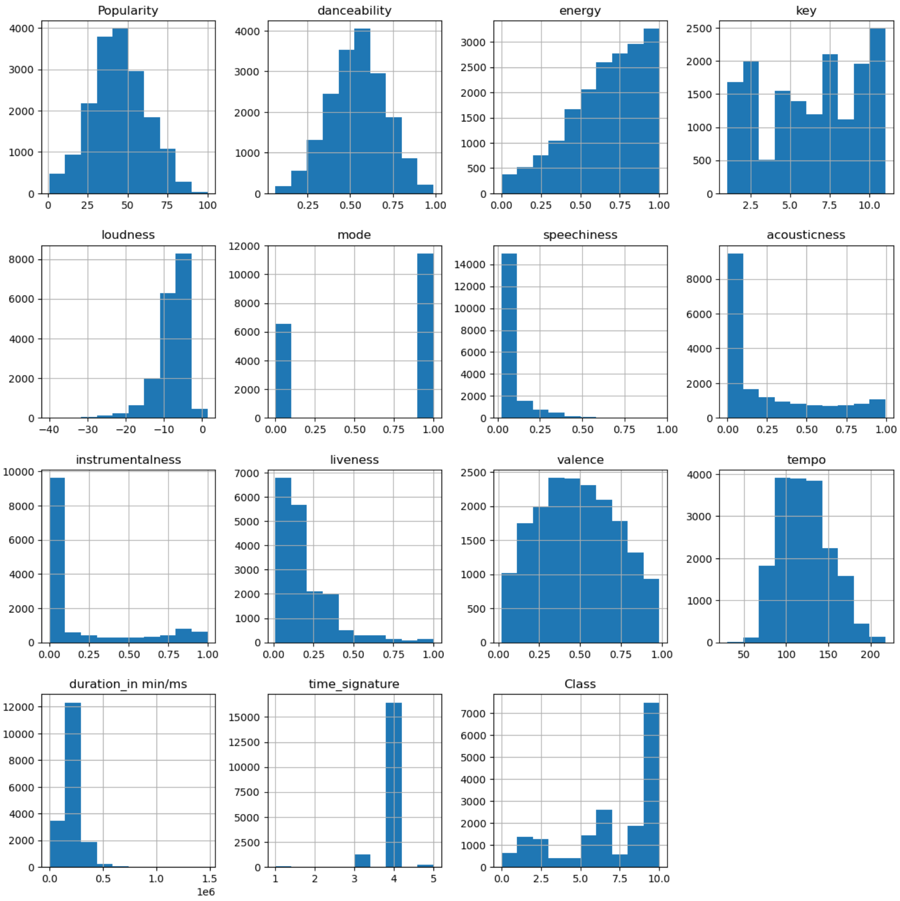


Figure 2: Different features distribution

Moreover, it is important to understand if we have an equal amount of observations for each one of the target attribute classes, which is rarely the case for real-life machine learning problems. As it can be observed from figure 3 some classes are better represented than others, with class 10 containing 5000 observations, which corresponds to approximately 1/3 of the entire dataset. Classes like 0, 3, 4 and 7 appear to be under represented in the dataset, with under 500 observations per class. Given that the values are not distributed homogeneously across the different classes would add to the challenge of finding a posterior distribution that can model well all the genre classes.

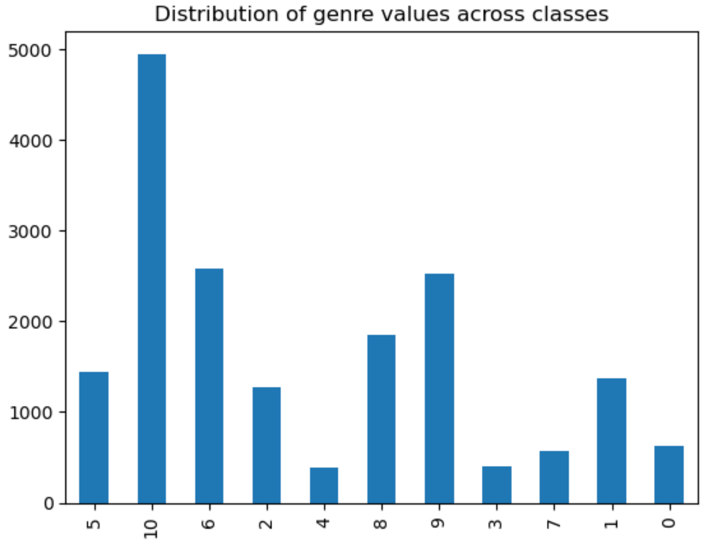


Figure 3: Distribution of values across the 11 music genre classes

Diving more into the details we can identify a high positive correlation between the "loudness" and "energy" of the soundtracks which is consistent with the basics of the music. Also quite a high negative correlation between "acousticness" with "energy" and "loudness" is observed in the data (4).

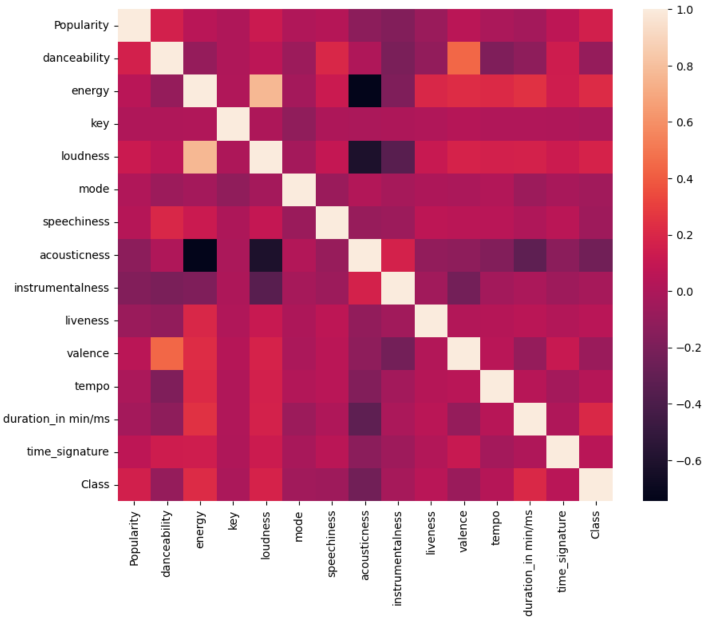


Figure 4: Different features distribution

Moving forward, we need to address any NaN values. We have NaN values in "key", "instrumentalness" and "popularity" features in our data and we decided to replace them with the mean value of each of those features. Also, there are two columns in our data called "Artist Name", and "Track Name" which we drop from the dataset as these columns do not provide valuable information to the prediction task.

5.1 Modeling Strategy

In our strategy for modeling, we first start with the simplest possible approach which is Ancestral sampling. In Ancestral sampling we just simply sample the parameters of our model (betas or weights) using Numpy's sampling function and without any use of the Pyro package. Also, in ancestral sampling, we do not consider any bias/intercept (alphas) and we do not sample them in the process. As the next step, we start implementing Pyro package to create models and train them to be used in the inference algorithms afterward. Since we are dealing with a classification problem, considering the data and also the model-based framework, our best approach would be to use a logistic regression model. We first start with very little data (100 data points each for train and test sets) and a simple Pyro model which samples alphas and betas (regression parameters) from normal distributions with mean 0 and standard deviation 5 as the priors. After sampling parameters, the model samples the observed variable from a categorical distribution. For the training process, in this experiment (as in most of our experiments except the one in which we use the MCMC method) we used Adam optimizer with 0.001 as the learning rate and Stochastic Variational Inference as the inference algorithm. Moving further in our analysis we conduct 9 other experiments, the details, and the results are described in the following sections of the report. In each of the experiments, we tried to tune the models and the parameters in a way to acquire an acceptable accuracy.

6 Modeling Approach

In this section, the models that were used in the project would be specified. First of all, since the target value is the music genre, which represents a discrete attribute, the algorithms for classification were used. Depending on the condition, many experiments have been conducted and the logistic regression and Feed Forward Neural Network(FFNN) models have been used together with and Markov Chain Monte Carlo and variational infernce (VI) methods for approximating the posterior distribution.

6.1 Ancestral Sampling

In the project, the array of betas is sampled from the Gaussian distribution and it is multiplied by the features of the dataset. Therefore, if the model is shown as B_c, then it can be said that the model is sampled from $\mathrm{P}(B_c|\beta)$. Ancestral sampling uses the probability given by its ancestor. Then, the sampled model is the result of the dot product of betas and the features, and it is used as an input of the Softmax, and the prediction of the class is the index that has the highest probability. Since the class has 11 labels from 0 to 10, the Softmax function and multinomial were in use.

6.2 Inference Methods

6.2.1 Markov Chain Monte Carlo

The MCMC is one of the inference algorithms for sampling from the posterior distribution and is one of the stochastic ways. By using sampled observations, the approximate representation of the true posterior could be derived, and also with respect to the posterior, expectations can be computed as well. In the project, for performing inference on the priors alpha and beta, it uses NUTS, and by its kernel, MCMC is performed. The training dataset is in use, and after training, the intercept and beta parameters are sampled from the posterior distribution. By using those samples, the test dataset can be used for measuring the accuracy of the models.

6.2.2 Variational Inference

From now on, the models performing Variational Inference (VI) would be explained. The goal of variational inference is to find the parameters of the simpler distribution that make it as similar as possible to the posterior distribution. By using Kullback-Leibler (KL) divergence, the inference problem becomes an optimisation problem, and the value of ELBO is used. In the project, the value of ELBO is measured for each step, and the number of steps is set to the point where the value of ELBO converges. After that, the ”Predictive” class is used for making predictions in the model, and by comparing predictions and ground truth class values, accuracy could be computed.

In the main model, the goal is to perform multi-class classification by logistic regression by using priors. There are four different variations depending on the condition of the prior and the way of variational inference, and they would be shown from 6.2.3 to 6.3.

6.2.3 The model without priors

In this model, for performing logistic regression, μ and λ are used as parameters. μ is used as mean and λ is used as the variance of beta, and beta is the coefficient of the logistic regression function. The model B_c is on the pyro plate and C stands for the class so it will be repeated 11 times. In the N plate (N stands for the number of features), there are numerical features for being used for training, and testing the model.

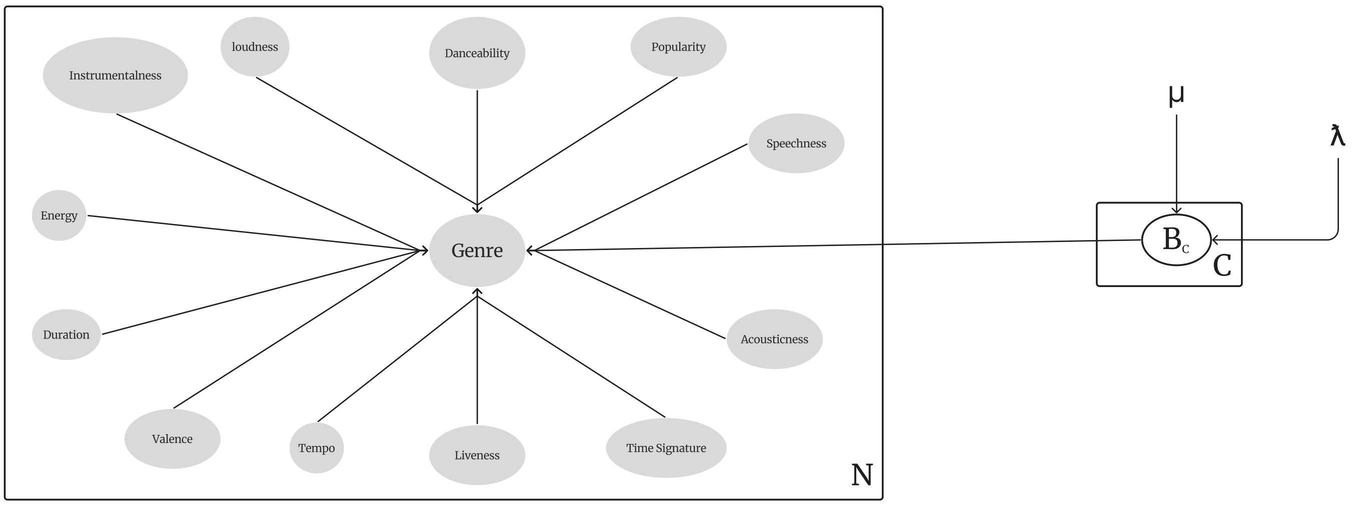


Figure 5: Initial Graphical Probabilistic Model

In the next model, the usage of priors V and τ would be specified.

6.2.4 Using priors V and τ

In this model, the V and τ have been used as priors of μ and λ of the beta. V and τ are defined by the following distributions of student-t and Halfcauchy for each. The reason why student-t distribution is used is it brings an advantage related to sampling from the posterior so that it makes the sampled data more informative.[2] By using μ and λ which have priors, the beta has been defined using the Gaussian distribution, and the model has been made on the pyro plate as B_c in the figure 6. For measuring the difference between using the normal distribution and student-t distribution, the other model used the normal distribution instead of student-t for the μ.

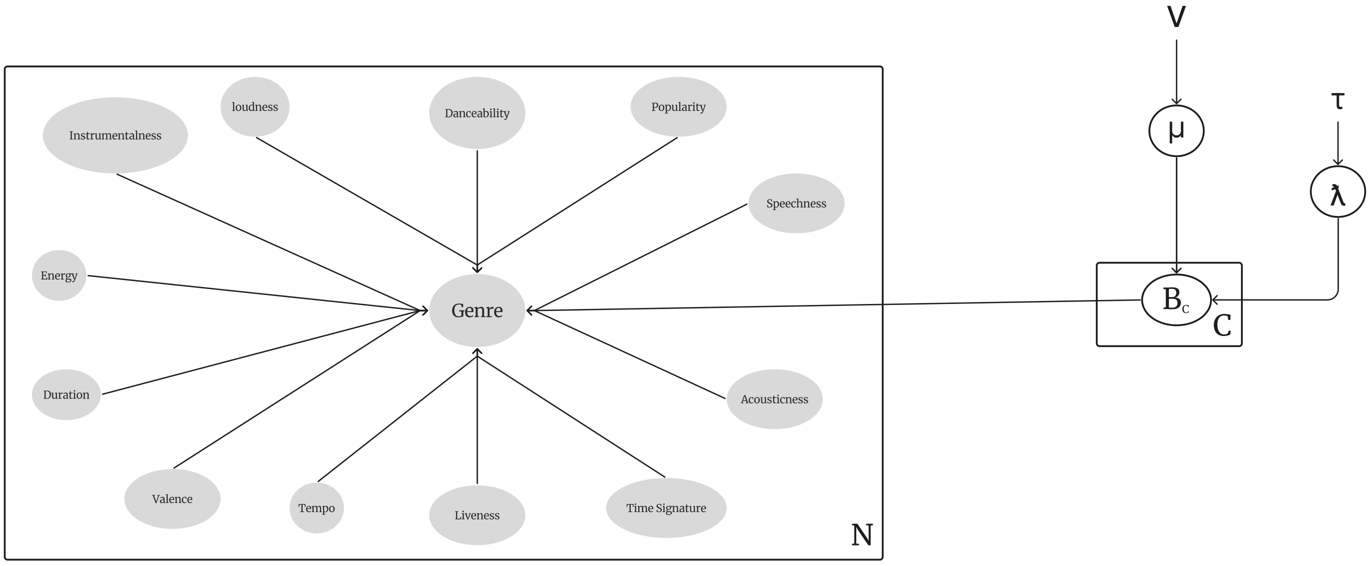


Figure 6: Graphical Probabilistic Model with Priors

For these main models, 100, 1000, and 10000 samples of observations have been used, and also some experiments were about prior and hyperparameter tuning so that the optimal parameters could be derived by validating the dataset.

6.2.5 Automatic Relevance Determination

For automatic feature selection, Automatic Relevance Determination (ARD) is in use. Looking into figure 7, the priors are the same as the last section, but the λ is in the pyro plate. It could show how each feature is relevant to the prediction, so the feature selection could be conducted.

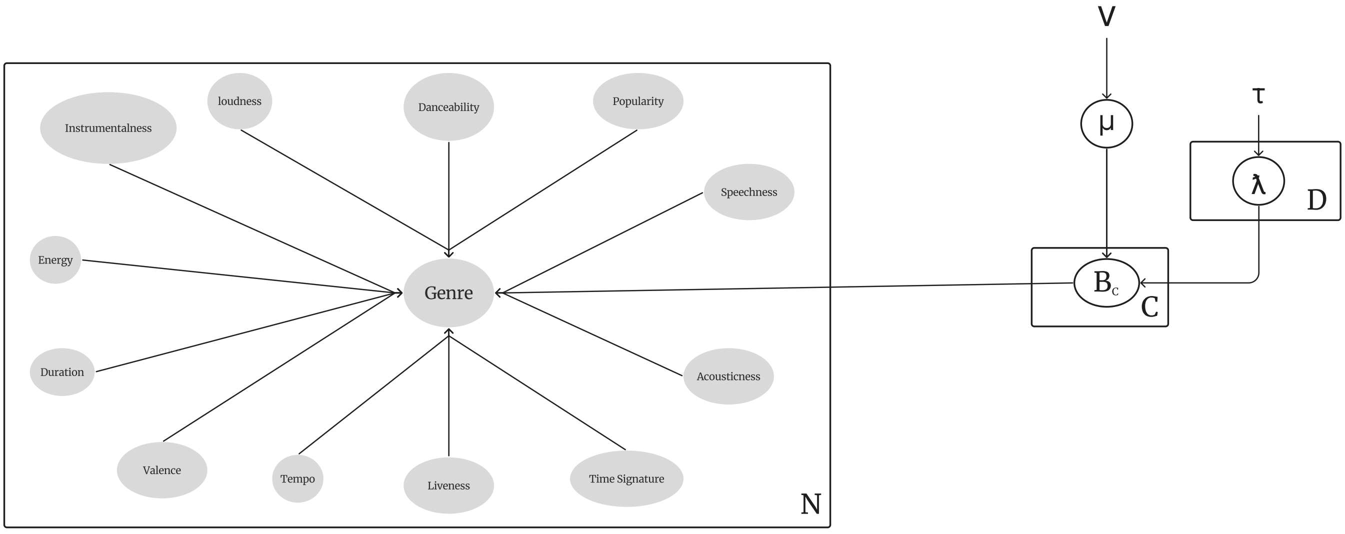


Figure 7: ARD Graphical Probabilistic Model

6.3 Feed Forward Neural Network

The FFNN is the neural network with some hidden layers and hidden units inside the layer. Each layer, it has its own weight, and it follows the normal distribution. μ works as mean and λ works as variance. Then, the FFNN model is B_c.

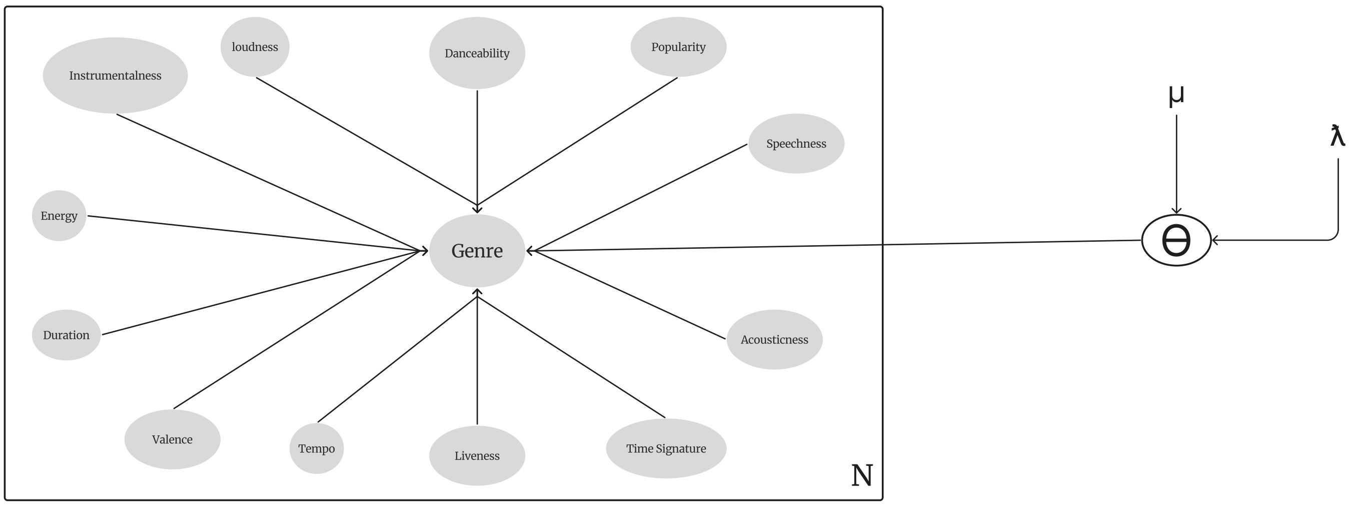


Figure 8: NN Graphical Probabilistic Model

The model is defined using the Pytorch class in Python, and each layer is specified, and in the forward function, each layer is using an activation function to compute the output. In the project, tanh and ReLU activation functions were in use, and the output goes through the softmax function for calculating the probability of the observation belonging to each class. Finally, an observation is assigned the class that belongs to the highest probability value and thus a prediction is obtained.

7 Results

Ten different experiments were carried in order to assess which of the selected models and inference methods produce the best results. The details of the models used are summarized in table 1. Due to a limitation of the compute resources, all experiments except experiment 1 and 6 were conducted using a train set and test set size of 1000 and 200 observations respectively.

The experimental results are shown in table 2. Experiment 1 can be considered as an initial test for understanding how well a simple logistic regression model can fit a very limited dataset of 200 observations. The training accuracy is 60%, while the accuracy obtained on the test set is 32%. As the data set size increased on 1000 (experiment 2) the test accuracy increased as well to 45%. The training accuracy is lower, which could mean that the initial train accuracy results obtained were by chance. Experiment 3 uses model 2 which makes use of priors for the mean and variance of the beta parameters. This increases train and accuracy results slightly to 51.7% and 45.5%.

Then, experiment 4 and 5 were meant to reveal which prior for mu_β works better, the one from the student-T distribution or the one from the normal distribution. Parameter tuning was also performed to understand what the optimal number of degrees of freedom (V) and τ is for the student-T distribution is. It was found that $V = 4$ and $\tau = 1 = 10$ gives best results. It was also observed that when using a normal distribution for mu_β a lambda variance value of 1 is optimal.
When mu_β follows a studentT distribution with $V = 4$ and $\tau = 1$, the test accuracy shows an increase of 1%. Experiment 6 was carried to understand if increasing the train set size by one order of magnitude will lead to a higher accuracy and, as it can be observed from the table the result obtained on the test set is 50.2%, which is roughly 4-5% better than for the previous experiments. Experiment 7 uses automatic relevance determination (ARD) to select which features are important for the prediction of the music genre when the training is carried. The test accuracy (46%) however is not better than the one of the models that are not making use of ARD.

During experiment 8, model 2 has been trained and tested for different learning rate values of 0.0001, 0.001 and 0.01. The training accuracy values obtained were optimal for a learning rate value of 0.001, which was the learning rate used for all other experiments. Experiment 9 makes use of a feed-forward neural network model for the classification task, where the weights for the different hidden layers were sampled from the normal distribution. Using this did not lead to better accuracy results (45.5% on the test set). For all experiments mentioned so far, variational inference (VI) was used in order to estimate the posterior predictive distribution.

Finally, in experiment 10 variational inference was exchanged for Markov Chain Monte Carlo and model 2 was trained on 1000 observations to understand if one method is superior to the other for the classification task at hand. The test set result obtained (47%) shows that there isn't a significant difference between VI and MCMC.

Model Id	Model	Alpha	Prior beta	Beta	ARD
1	logistic regression	$Alpha \sim$ (0 ,5)	N/A	$Beta \sim \mathcal{N}(0, 5)$	N/A
2	logistic regression	$Alpha \sim \mathcal{N}(0,5)$	$\mu_{beta} \sim$ StudentT(v) $\sigma_{beta} \sim$ HalfCauchy(τ)	$Beta \sim \mathcal{N}(\mu_{beta},\sigma_{beta})$	N/A
3	logistic regression	$Alpha \sim \mathcal{N}(0,5)$	$\mu_{beta} \sim \mathcal{N}(0, \lambda)$ $\sigma_{beta} \sim$ HalfCauchy(τ)	$Beta \sim \mathcal{N}(\mu_{beta},\sigma_{beta})$	N/A
4	logistic regression	$Alpha \sim \mathcal{N}(0,5)$	$\mu_{beta} \sim$ StudentT(v) $\sigma_{beta} \sim$ HalfCauchy(τ)	$Beta \sim \mathcal{N}(\mu_{beta},\sigma_{beta})$	Yes
5	Feed-Forward NN			$Beta \sim \mathcal{N}(0,1)$	

Table 1: Models Used for Experiments

Experiment	Model Id	Train set size	Inference Method	Acc. Train set (%)	Acc. Test Set (%)
1	1	100	VI	63	32
2	1	1000	VI	50.4	45
3	2	1000	VI	51.7	45.5
4	3	1000	VI	49.1	45
5	2	1000	VI	50.3	46
6	2	10000	VI	49.9	50.2
7	4	1000	VI	51.1	46
8	2	1000	VI	50.7	47
9	5	1000	VI	49.2	45.5
10	2	1000	MCMC	50.8	47

Table 2: Experimental Results

8 Conclusion and Discussion

In conclusion, the accuracy values obtained on the test sets show that the different models considered can capture to some extent the complexity of the dataset, and that the results obtained are not by chance which would correspond to an accuracy value of 9%. However, given that the best accuracy score obtained was 50.2% one can conclude that there is still room for improvement.

One consideration that is worth mentioning is that two of the attributes of the dataset, namely track name and artist name, were initially eliminated. This decision was taken due to the fact that these attributes were text attributes and the team did not have a clear strategy on how feature engineering could be used here to extract relevant features.

Furthermore, it is to be noted that in the dataset, 4377 observations for the attribute instrumentalness, 2014 observations for the attribute key and 428 observations for the popularity attribute had missing values. The way the missing values had been dealt with might deviate from the truth, and thus impact the accuracy. The distribution of observations in the dataset is also not homogeneous across all eleven classes, which might have been an impediment to finding a posterior distribution approximation that is close to the real distribution. And lastly, the number of observations used for most of the experiments was limited to 1000 due to scarce compute resources.

The considerations mentioned above could have an impact on the accuracy scores, and if further experiments are to be conducted, investigating them could lead to better approximations of the posterior distributions and thus higher accuracy scores.

Abstract	Background	D. Description	D. Selection	D. Analysis	Modelling	Results	Conclusion
1	2	3	4	5	6	7	8

References

[1] Kaggle. Music genre classification. *Music Genre Classification.*

[2] Stan. Prior choice recommendations. *Prior Choice Recommendations.*