Faculty of Computer Science
Major in Artificial Intelligence

Foundations of Machine Learning

Song classification by music genre using machine learning techniques

Rafael Balseiro Couceiro
Fiz Garrido Escudero
Guillermo Blanco Núñez
Miguel Pérez Francos

September 29, 2025

Contents

1 Introduction

In recent years, access to and consumption of music has grown exponentially thanks to the rise of streaming platforms such as Spotify, Apple Music, and YouTube Music. With catalogs containing millions of songs, the efficient classification and organization of this content has become essential to improve user experience and maximize the performance of both recommendation and search systems, while also representing a business opportunity for the music industry.

Although classifying songs by genre may sound like a trivial task for humans, it represents a significant challenge for computational systems. The inherent ambiguity of musical genres, combined with the high dimensionality of music data, necessitates the use of machine learning algorithms capable of effectively capturing complex patterns.

This work addresses the automatic classification of music tracks into genres using the "Spotify Audio Features" dataset from Kaggle [tomigelo, 2019], which contains musical features provided by Spotify [Spotify, 2024]. Several fundamental machine learning models will be evaluated, applying cross-validation techniques and hyperparameter optimization. The objective is to identify the most effective strategies for this type of problem and to propose potential improvements for future developments.

2 Problem Description

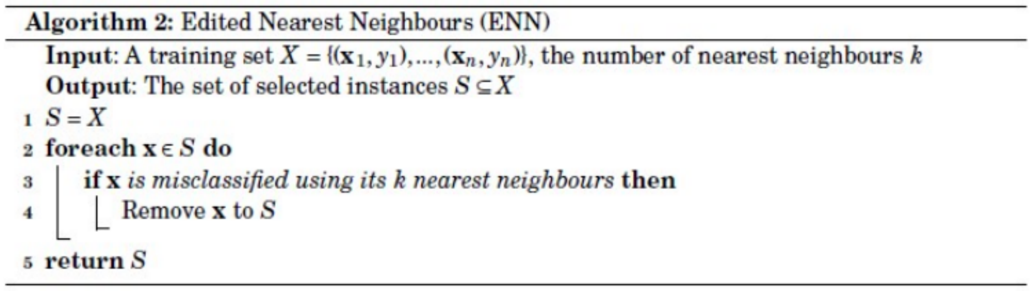
The problem under study is the classification of tracks by genre using features extracted from the Spotify Audio Features dataset [tomigelo, 2019] on Kaggle, which contains over 100,000 tracks along with their title, artist, and various attributes generated by Spotify to describe each song.

2.1 Dataset Elaboration

In this case, the original dataset did not explicitly contain the genre of each track, so Spotify's API was used to retrieve this information. Once all the necessary data were collected, we selected the ten most representative mutually exclusive genres (i.e., without including subgenres alongside their parent genres) in the dataset. These were: *classical*, *corrido*, *country*, *electronic*, *jazz*, *k-pop*, *lo-fi*, *metal*, *reggae* and *reggaeton.*

2.2 Dataset Reduction

Secondly, due to computational constraints Random Subsampling and ENN (Edited Nearest Neighbors) [Cubillos, 2022] were applied to the dataset in order to reduce the number of instances. ENN has enabled the selection of potentially representative instances by removing noise and potential outliers in the dataset.



Algorithm 2: Edited Nearest Neighbours (ENN)

```
Input: A training set $X = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, the number of nearest neighbours $k$
Output: The set of selected instances $S \subseteq X$
$S = X$
foreach $\mathbf{x} \in S$ do
    if $\mathbf{x}$ is misclassified using its $k$ nearest neighbours then
        Remove $\mathbf{x}$ to $S$
return $S$
```

Figure 1: Edited Nearest Neighbors (ENN) algorithm: pseudocode

The resulting dataset contains 3,500 balanced instances, compared to the initial dataset of over 20,000 instances. Both techniques were applied to the entire dataset with the goal of obtaining a balanced dataset, which led to more aggressive undersampling in certain classes containing more instances.

2.3 Data

The attributes contained in the dataset [Spotify, 2024] include both original acoustic features such as tempo, duration, or loudness, as well as Spotify-inferred features with more subjective definitions, such as danceability or energy. The features, their possible values, and their meanings are described below:

- **song**: name of the track.
- **artist**: name of the performing artist.
- **genre**: genre to which the track belongs.
- **acousticness**: confidence [0, 1] that the track is acoustic.
- **danceability**: confidence [0, 1] that the track is suitable for dancing.
- **duration**: duration of the track in milliseconds.

- **energy**: measurement [0, 1] of the intensity and activity of the track.
- **instrumentalness**: confidence [0, 1] that the track contains no vocals (higher values indicate fewer vocals).
- **key**: musical key of the track (categorical value, e.g., C, D, E. . .).
- **liveness**: likelihood [0, 1] that the track was performed live.
- **loudness**: average loudness of the track in decibels (dB).
- **mode**: mode of the track (1 for major, 0 for minor).
- **speechiness**: likelihood [0, 1] that the track contains spoken words.
- **tempo**: speed of the rhythm in beats per minute (BPM).
- **time signature**: estimated overall time signature of the track (e.g., 4/4, 3/4).
- **valence**: [0, 1] measure of musical positiveness conveyed by the track.
- **popularity**: popularity of the track in global rankings.

2.4 Exploratory Data Analysis

After performing a brief exploratory analysis of the data, aiming to study their distributions and potential discriminative qualities useful for classification, we found that the distributions are quite similar across all genres. This results in a high degree of overlap, leading to complex relationships with their respective genres. Furthermore, many of the attributes exhibit wide ranges, as shown in Table 1, due to the creative diversity within each musical genre.

In addition, when the distributions of the variables are visualized conditioned on each genre, the difficulty of identifying clear discriminative features becomes evident, highlighting the class overlap problem. For example, in Figure 2, it can be observed that tempo in BPM is not as discriminative as one might initially expect.

Regarding the dependence between variables, the Pearson correlation matrix for the continuous features was computed (Figure 3). Although dimensionality is not an issue in this dataset, notable correlations are observed between loudness and energy, as well as between energy and acousticness, which is consistent with the definitions of these features. However, these correlations are not sufficient to discard any of the variables.

Table 1: Statistical summary of continuous variables

Variable	**Min**	**Max**	**Mean**	**Std**
acousticness	0.0	0.996	0.3512	0.3459
danceability	0.0	0.983	0.587	0.1746
duration_ms	10587.0	1.90877e6	2.12012e5	1.04194e5
energy	0.0	0.999	0.5876	0.2637
instrumentalness	0.0	0.984	0.2354	0.3633
liveness	0.0	0.99	0.2034	0.1698
loudness	-60.0	0.636	-9.3521	6.367
speechiness	0.0	0.915	0.0986	0.102
tempo	0.0	239.553	119.461	30.896
valence	0.0	0.999	0.501	0.2669
popularity	0.0	94.0	31.3583	20.8113

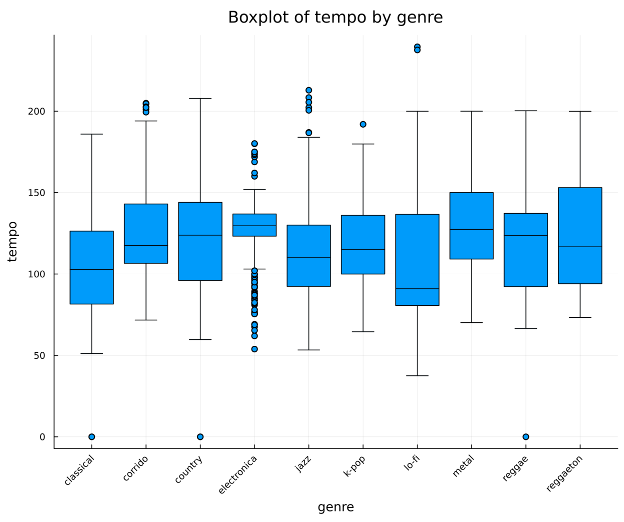


Figure 2: Boxplot of tempo conditioned on genre

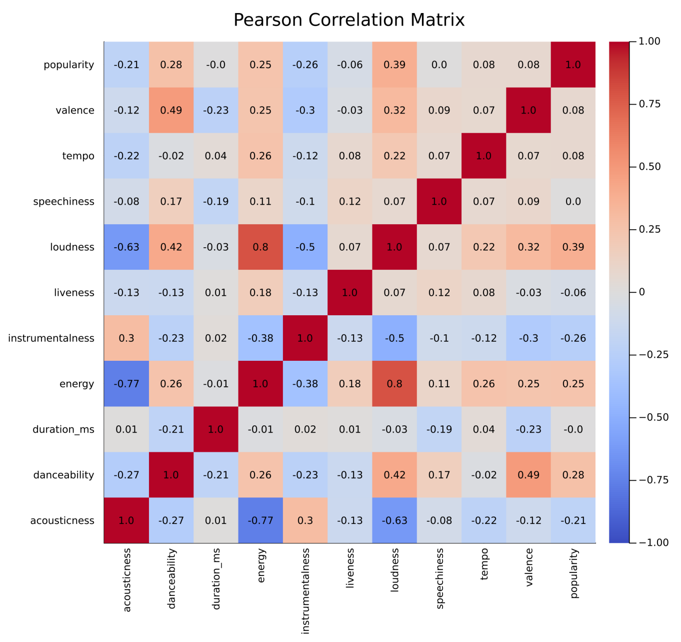


Figure 3: Pearson correlation matrix of numerical variables

2.5 Metrics used

- **Accuracy**: A reliable overall metric in balanced scenarios. In this case, it will be used as the primary decision metric due to the well-balanced classes (10 musical genres) in the dataset, as well as the empirical observation that **F1** and **accuracy** values are similar in the results analysis across all models.

  Accuracy measures the percentage of correctly classified instances over the total, making it very useful for this problem:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

  where:

  – TP (True Positives): correctly classified instances of a given genre.

  – TN (True Negatives): instances of other genres correctly excluded.

  – FP (False Positives): instances of other genres incorrectly classified as this one.

  – FN (False Negatives): instances of this genre incorrectly classified as another.

- **Weighted F1-score**: Provides a more detailed view of per-class performance, balancing the model's ability to correctly predict positives without generating too many false positives or false negatives, and accounts for class imbalance (although in our case the classes are balanced). First, for each genre i we define:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}, \qquad \text{Recall}_i = \frac{TP_i}{TP_i + FN_i},$$

$$F_1^{(i)} = 2 \cdot \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}. \tag{2}$$

Then, the *Weighted F1* is calculated as:

$$\text{Weighted}-\text{F1} = \sum_{i=1}^{C} w_i \, F_1^{(i)}, \quad w_i = \frac{N_i}{\sum_{j=1}^{C} N_j},$$

where N_i is the number of instances of genre i and $C = 10$ is the total number of genres.

- **Confusion matrix**: a table of (TP, FP, FN, TN) counts for all pairs of genres, wich allows the identification of systematic confusions (e.g., "jazz" vs. "lo-fi").

3 Literature Review

Various approaches, models, and data sources have been explored for music genre classification, ranging from traditional *machine learning*—as in this work—to deep learning and large language models (LLMs). [Ndou et al., 2021] provide a comparative review, showing that kNN on three-second GTZAN features achieves 92.69% accuracy, outperforming CNNs on spectrograms.

Simpler approaches include the work of Setiadi et al. [Rahardwika et al., 2020], which applies SVMs with Chi-square feature selection on a dataset similar to ours, achieving an accuracy of 0.8. Li Guo et al. [Guo et al.,] extract audio features and apply KNN, SVM, and neural networks, though with less promising results.

Deep learning approaches often rely on spectrograms. Pelchat and Gelowitz [Pelchat and Gelowitz, 2020] feed spectrograms into CNNs, while Elbir and Aydin [Elbir and Aydin, 2020] propose *MusicResNet*, using intermediate-layer representations as input for SVMs, reaching 0.97 accuracy.

Multimodal strategies combine audio, lyrics, and album artwork. Dammann and Haugh [Dammann and Haugh, 2017] integrate Naive Bayes for lyrics, LSTM for audio, and KNN on PCA/LDA-reduced images. Oramas et al. [Oramas et al., 2018] fuse convolutional networks for audio and images to enhance classification performance.

Recent advances explore LLMs for feature extraction directly from audio, followed by neural network classification, as in Meguenani et al. [Meguenani et al., 2024]. These works illustrate the wide spectrum of current techniques for automatic music genre classification.

4 Development

The procedure proposed in this work consists of testing different basic classification models, specifically artificial neural networks, support vector machines, decision trees, KNN, and DoME. Stratified **cross-validation** with k=10 was applied in all experiments, and the data were normalized as described in Section 4.1.

It should be noted that data normalization was performed within each cross-validation fold, using only the parameters obtained from the training set. These parameters were then applied to both the training and validation sets (when applicable). For certain models, experiments were conducted with and without normalization to assess its effect on model performance.

Randomness was also taken into account. To ensure experiment replicability, all experiments were executed using the same cross-validation splits, and the same seed was applied for the validation set.

4.1 Data Preprocessing

Before fitting the different models, the variables were normalized due to differences in scale (for example, loudness is measured in decibels while duration is measured in milliseconds). These discrepancies can negatively affect algorithms that are sensitive to the scale of the data.

The normalization strategy applied varies depending on the type of variable and its properties:

- **Time signature**: This variable represents the track's time signature and takes five possible values (1 to 5). Since there is no linear order relationship among them, *one-hot encoding* was applied, taking advantage of its low cardinality which does not significantly increase the dataset's dimensionality.

- **Key (pitch class)**: This variable encodes the pitch class on a cyclic 12-value scale (0 to 11). To preserve its circular nature, it was transformed using cyclic encoding with sine and cosine functions:

$$\text{key}_{\text{sin}} = \sin\left(2\pi \cdot \frac{\text{key}}{12}\right), \quad \text{key}_{\text{cos}} = \cos\left(2\pi \cdot \frac{\text{key}}{12}\right)$$

- **Mode**: A binary variable indicating the tonality (major or minor). Since it only contains values 0 and 1, no further transformation is required.

- **Other continuous variables**: These were standardized using Z-score normalization, based on the mean and standard deviation of each variable:

$$z = \frac{x - \mu}{\sigma}$$

4.2 Results

4.2.1 Artificial Neuronal Networks

In the experiments with ANNs (Artificial Neural Networks), different combinations of shallow network topologies (up to two hidden layers), transfer functions, and learning rates were tested. To prevent overfitting, training was performed using early stopping. All networks used common hyperparameters: 120 epochs, a patience of 10 epochs, and 30 training runs per fold.

Table 2: Performance of ANNs with different hyperparameters

LR	**Topology**	**Act. Funcs**	**Acc. Mean**	**Acc. Std**	**F1 Mean**	**F1 Std**
0.01	[48, 24]	[σ, σ]	0.6045	0.0163	0.5999	0.0163
0.01	[36, 18]	[σ, σ]	0.5920	0.0163	0.5864	0.0169
0.1	[32, 16]	[ReLU, ReLU]	0.5646	0.0124	0.5639	0.0127
0.01	[32, 16]	[tanh, tanh]	0.5267	0.0119	0.5260	0.0114
0.01	[64, 32]	[tanh, ReLU]	0.5463	0.0125	0.5461	0.0116

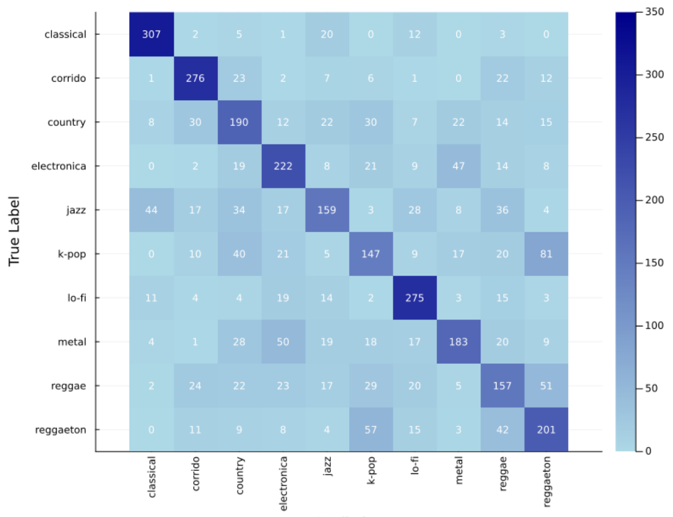


Figure 4: Confusion matrix of the neural network with topology [48, 24] and tranfer functions [σ, σ]

The experiment shows that activation functions play an important role in improving accuracy, with the best results achieved using [σ, σ]. More complex topologies do not necessarily yield better results; for instance, the network with topology [64, 32], despite having more neurons, does not achieve higher accuracy (0.5463).

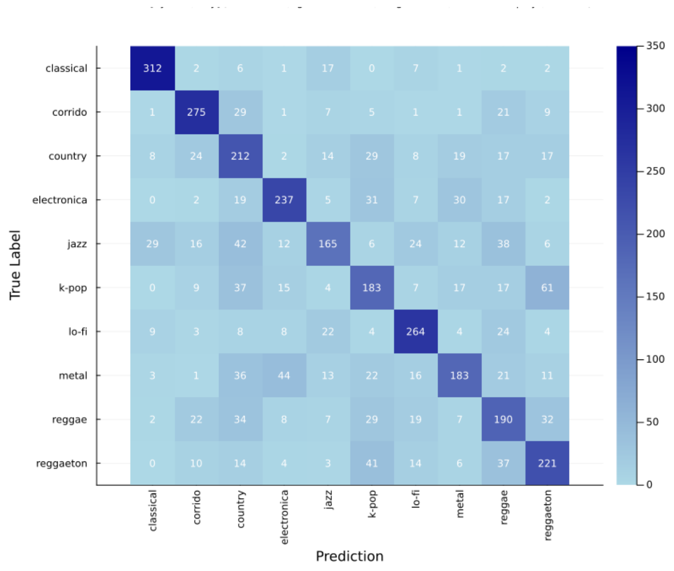
This behavior may be due to the fact that the dataset is not large enough to fully exploit the capacity of more complex architectures, leading to overfitting or inefficient training. Additionally, sigmoid activations help stabilize learning in these smaller networks by keeping outputs bounded, which can improve convergence and overall performance.

4.2.2 Support Vector Machines

In the experiments conducted with support vector machines, different combinations of kernels, C values (penalty for misclassified instances), and gamma values (for kernels that require it) were tested, yielding the following results 3.

Table 3: Performace of SVMs with different hyperparameters

C	**Kernel**	**Gamma**	**Coef0**	**Degree**	**Acc. Mean**	**Acc. Std**	**F1 Mean**	**F1 Std**
0.1	poly	0.1	1	2	0.6149	0.0173	0.6132	0.0169
0.1	poly	0.1	1	3	0.6363	0.0181	0.6361	0.0172
0.1	linear	–	–	–	0.5894	0.0153	0.5840	0.0192
10	poly	0.1	1	2	0.6203	0.0184	0.6182	0.0169
1	poly	0.1	1	2	0.6406	0.0169	0.6400	0.0159
1	poly	0.01	1	3	0.5991	0.0196	0.5955	0.0210
1	poly	0.1	1	3	0.6277	0.0147	0.6269	0.0138
1	poly	0.1	1	4	0.6074	0.0238	0.6051	0.0233
1	sigmoid	0.1	1	–	0.3643	0.0203	0.3612	0.0186
1	rbf	0.01	–	–	0.5877	0.0252	0.5831	0.0273
1	linear	–	–	–	0.5963	0.0152	0.5915	0.0179



True \ Prediction	classical	corrido	country	electronica	jazz	k-pop	lo-fi	metal	reggae	reggaeton
classical	312	2	6	1	17	0	7	1	2	2
corrido	1	275	29	1	7	5	1	1	21	9
country	8	24	212	2	14	29	8	19	17	17
electronica	0	2	19	237	5	31	7	30	17	2
jazz	29	16	42	12	165	6	24	12	38	6
k-pop	0	9	37	15	4	183	7	17	17	61
lo-fi	9	3	8	8	22	4	264	4	24	4
metal	3	1	36	44	13	22	16	183	21	11
reggae	2	22	34	8	7	29	19	7	190	32
reggaeton	0	10	14	4	3	41	14	6	37	221

Figure 5: Confusion matrix for SVM with polynomic kernel, C=1, gamma=0.1, coef0=1 and degree=2

Support vector machines for classification show notable differences depending on the kernel used. Starting with the sigmoid kernel, it achieves very low accuracy, highlighting its limited applicability in contexts as complex as this one. Meanwhile, polynomial kernels perform quite well. The nature of this kernel, which allows projecting the data into higher-dimensional spaces and capturing more complex relationships, enables better performance on this problem. However, increasing the degree does not necessarily improve classification accuracy; in fact, higher-degree polynomials may lead the model to overfit the training noise, reducing its effectiveness on the test set. Finally, the radial kernel appears less capable of capturing the relationships in the data, achieving lower classification accuracy compared to polynomial kernels. This may suggest that the nonlinearity of this dataset is not particularly high. The linear kernel performs roughly at the same level as the radial kernel, while being a considerably simpler architecture.

Regarding the optimal C and gamma values for SVCs in this problem, these depend entirely on the kernel used. The role of gamma is also crucial: if it is set too low, information is lost. According to the results, the optimal gamma value is 0.1.

4.2.3 Decision Trees

In the case of decision trees, experiments were conducted with different maximum depth values to test various levels of model complexity. Additionally, the model was trained both with and without normalization, as decision trees are not sensitive to feature scales.

Table 4: Results for decision trees.

	With normalization				**Without normalization**			
max_depth	**acc_mean**	**acc_std**	**f1_mean**	**f1_std**	**acc_mean**	**acc_std**	**f1_mean**	**f1_std**
5	0.4751	0.0290	0.4586	0.0399	0.4837	0.0189	0.4709	0.0236
6	0.5009	0.0331	0.4953	0.0361	0.5077	0.0259	0.4977	0.0348
8	0.5269	0.0373	0.5276	0.0364	0.5369	0.0227	0.5381	0.0213
9	0.5297	0.0291	0.5296	0.0264	0.5420	0.0245	0.5415	0.0230
10	0.5234	0.0237	0.5248	0.0220	0.5326	0.0250	0.5321	0.0230
11	0.5114	0.0420	0.5123	0.0407	0.5297	0.0252	0.5311	0.0247
12	0.5191	0.0275	0.5207	0.0238	0.5300	0.0305	0.5308	0.0292
13	0.5094	0.0246	0.5105	0.0234	0.5277	0.0159	0.5295	0.0170
14	0.5083	0.0280	0.5095	0.0259	0.5280	0.0162	0.5283	0.0133
15	0.5114	0.0276	0.5126	0.0246	0.5314	0.0178	0.5327	0.0176
20	0.5131	0.0303	0.5142	0.0282	0.5300	0.0198	0.5304	0.0187

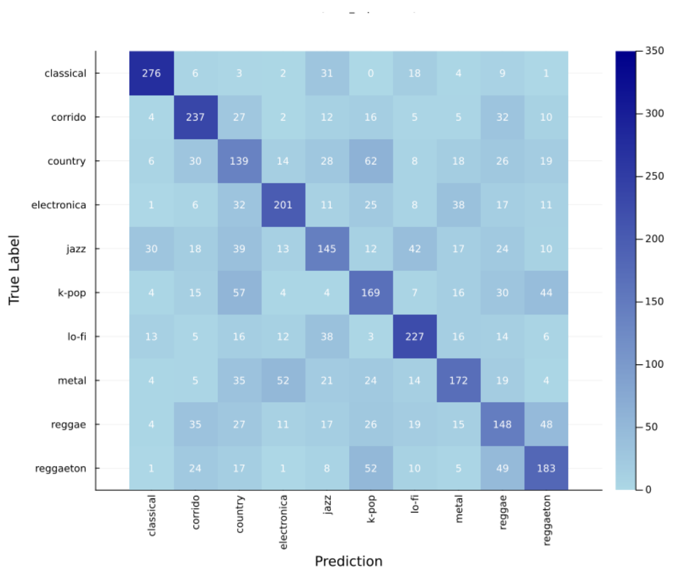


Figure 6: Confusion matrix for `max_depth`=9, no normalization

A configuration with max depth=9 without normalization achieved the highest average accuracy (0.542) because it balances the tree's capacity to capture interactions between features without incurring excessive overfitting. When the depth is too low, the tree is too shallow and fails to model non-linear relationships between features such as "energy" and "acousticness"; conversely, very large depths (over 12) introduce branches that fit the noise in the data, increasing variance and reducing generalization. The model's invariance to feature scaling explains why normalization provides no improvement: the tree only compares the order of values, not their absolute magnitudes.

4.2.4 K-Nearest Neighbours

For the KNN model, multiple configurations of K (number of nearest neighbors used for decision making) were tested, yielding the following results 5.

Table 5: Results of kNN for different values of k.

k	Acc. mean	Acc. std	F1 mean	F1 std
3	0.5337	0.0169	0.5299	0.0171
5	0.5594	0.0257	0.5580	0.0244
7	0.5706	0.0272	0.5684	0.0272
10	0.5734	0.0285	0.5718	0.0297
15	0.5766	0.0233	0.5747	0.0238
18	0.5797	0.0206	0.5782	0.0206
20	0.5794	0.0274	0.5773	0.0268
22	0.5777	0.0248	0.5753	0.0245

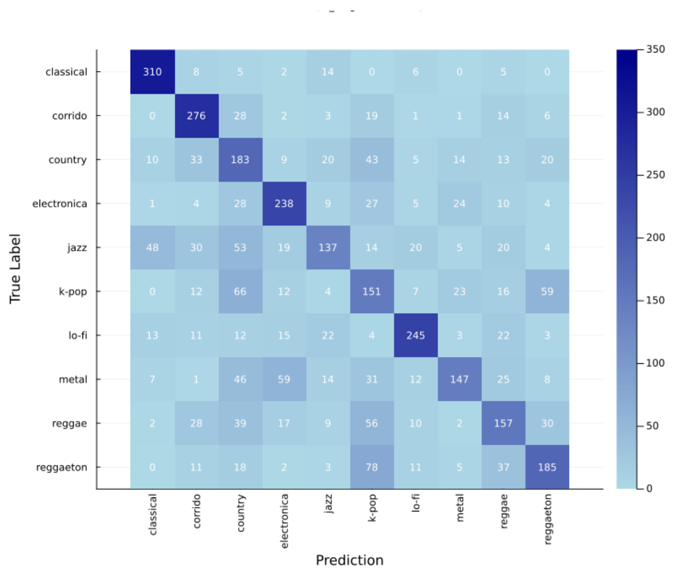


Figure 7: Confusion matrix for `k`=18.

The results of the experiment show that accuracy increases slightly as the value of k rises, reaching 0.5706 with k=7. Beyond this point, accuracy remains largely stable, with only minor increases, reaching a maximum of 0.5797 at k=18. This indicates that KNN was unable to capture the complex relationships present in the data, and despite increasing model complexity, the results did not improve, making it poorly suited for this problem.

4.2.5 DoME (Developement of Mathematical Expressions)

The DoME (*Developement of Mathematical Expressions*) model is a symbolic regression model that builds a tree to form arithmetic expressions. The hyperparameters adjusted in this experiment were the maximum depth and the learning rate.

Table 6: Results for DoME

Learning Rate	Max. Nodes	Acc. Mean	Acc. Std	F1 Mean	F1 Std
0.005	5	0.3249	0.0223	0.2475	0.0192
0.015	15	0.4563	0.0252	0.3948	0.0266
0.01	10	0.3863	0.0210	0.3211	0.0228
0.02	20	0.4714	0.0246	0.4198	0.0242
0.02	30	0.4960	0.0205	0.4526	0.0241
0.03	20	0.4714	0.0246	0.4198	0.0242
0.04	20	0.4714	0.0246	0.4198	0.0242
0.01	35	0.5154	0.0192	0.4791	0.0193
0.01	40	0.5223	0.0188	0.4862	0.0185
0.01	43	0.5283	0.0181	0.4920	0.0185
0.01	45	0.5274	0.0161	0.4920	0.0171
0.01	50	0.5340	0.0146	0.5001	0.0163

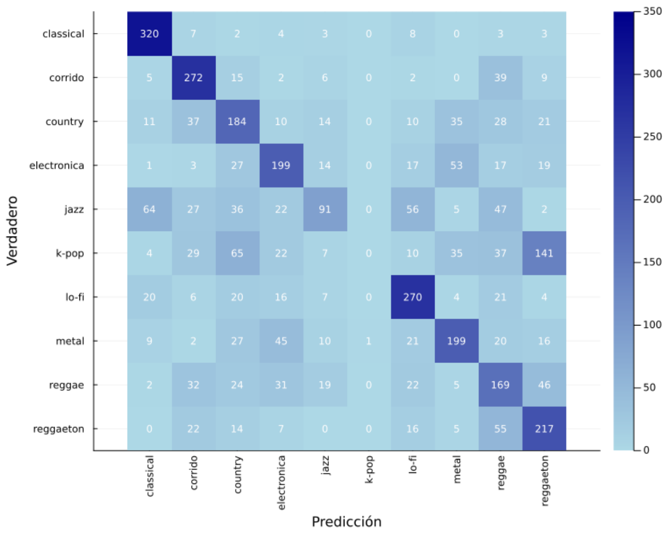


Figure 8: Confusion matrix for ($maximumNodes$=50)

In this experiment with DoME, the results were notably poor, requiring a significant increase in complexity to surpass an accuracy of 0.5, which was achieved with a maximum of 35 nodes. Although performance improves as complexity increases, the improvement is slow, reaching a maximum accuracy of 0.5340 with 50 nodes. These results are understandable given the nature of the model: constructing a tree representing simple arithmetic operations (addition, subtraction, multiplication, and division) is insufficient to successfully capture the complex relationships present in a dataset for musical classification, which involves diverse and abstract features, as observed in the exploratory data analysis. Furthermore, the training process for this model is computationally expensive, limiting the feasibility of experimenting with higher complexity levels.

4.3 Statistical Significance Tests

After training and evaluating the models, it is necessary to perform statistical significance tests in order to determine on a solid basis which classifier performs best in the experiments. For this purpose, the five classifiers with the highest mean *accuracy* were selected and subjected to the following statistical tests.

Friedman Test. The Friedman test is a non-parametric test that allows comparison of more than two classifiers executed on the same dataset, without assuming normality in the distributions. The Friedman statistic is defined as:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left(\sum_{j=1}^{k} R_j^2 \right) - 3N(k+1),$$

where N is the number of datasets, k the number of algorithms, and R_j the sum of ranks for classifier j. Under the null hypothesis that all algorithms have the same performance, this statistic follows a χ^2 distribution with $k-1$ degrees of freedom.

Paired Wilcoxon Test. After detecting significant differences with the Friedman test, the Wilcoxon signed-rank test is applied as a post-hoc analysis. This test compares two algorithms based on the differences in their performance. The test statistic is defined as:

$$W = \sum_{i=1}^{n} \operatorname{sign}(x_{2,i} - x_{1,i})\, R_i$$

$$\operatorname{sign}(x_{2,i} - x_{1,i}) = \begin{cases} +1, & x_{2,i} - x_{1,i} > 0 \\ -1, & x_{2,i} - x_{1,i} < 0 \\ 0, & x_{2,i} - x_{1,i} = 0 \end{cases}$$

$$R_i = \text{rank of } |x_{2,i} - x_{1,i}| \text{ among all nonzero differences}$$

where $x_{1,i}$ and $x_{2,i}$ are the paired observations, R_i are the ranks of the absolute differences, and zeros are excluded. The statistic W corresponds to the signed sum of ranks, with positive differences contributing positively and negative differences negatively. The Holm-Bonferroni correction is be applied to adjust the *p-values* when performing multiple comparisons, reducing the probability of Type I errors.

Selected Models.

1. **SVM 1:** Polynomial kernel, degree 2, $C = 1$, coef0 = 1, $\gamma = 0.1$
2. **SVM 2:** Polynomial kernel, degree 2, $C = 10$, coef0 = 1, $\gamma = 0.1$
3. **ANN 3:** Topology $[48, 24]$, activation functions $[\text{sigmoid}, \text{sigmoid}]$, $LR = 0.01$
4. **SVC 4:** Polynomial kernel, degree 3, $C = 1$, coef0 = 1, $\gamma = 0.1$
5. **SVC 5:** Linear kernel, $C = 1$

Table 7: Mean Accuracy and Std of the Models Selected

Model	Mean Accuracy	Std
SVC 1	0.6406	0.0169
SVC 2	0.6203	0.0184
ANN 3	0.6045	0.0163
SVC 4	0.5991	0.0196
SVC 5	0.5963	0.0152

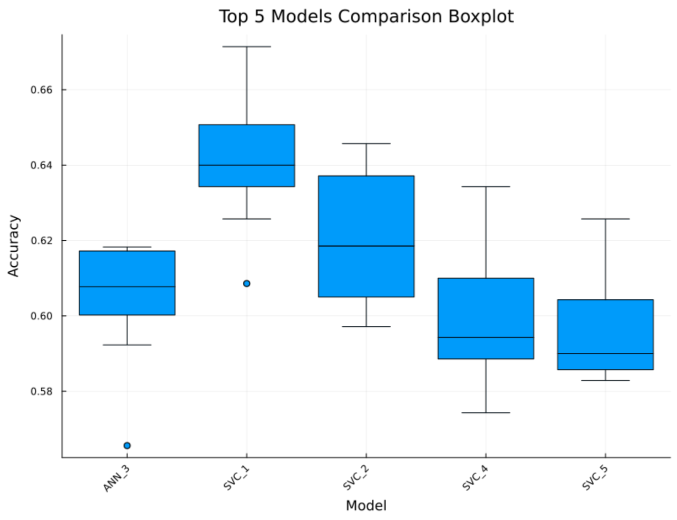


Figure 9: Visual representation of models's performace

Table 8: Friedman Test Results

Statistic	Value	Interpretation
Friedman statistic	25.04	
Degrees of freedom	4	
p-value	4.9×10^{-5}	< 0.05 (significant)
Significance level	0.05	Differences detected

Table 9: Average Ranking of Models (Friedman)

Model	Average Ranking
SVC 1	1.2
SVC 2	2.3
ANN 3	3.3
SVC 4	4.0
SVC 5	4.2

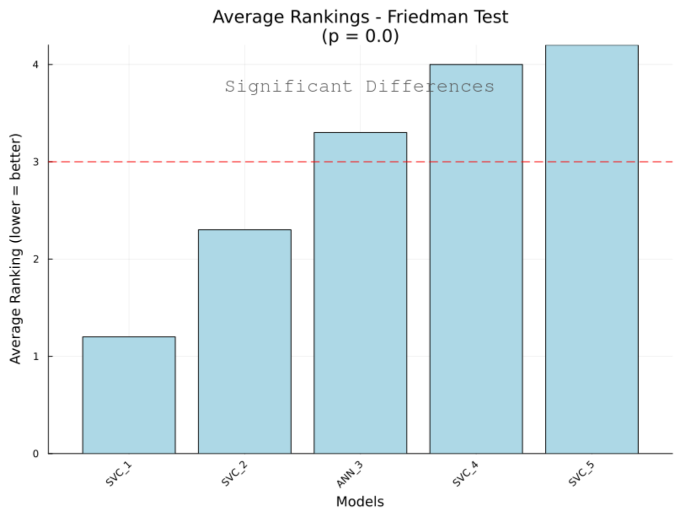


Figure 10: Graphical representation of Friedman rankings

Friedman Test Results. The Friedman test indicates that there are significant differences in model performance. Consequently, pairwise comparisons are performed using the Wilcoxon test.

Table 10: Paired Wilcoxon Test Results with Holm-Bonferroni Adjustment

Model1	Model2	Statistic	Raw p-value	Adj p-value	Significant
SVC 1	SVC 2	51.0	0.0117	0.0703	No
SVC 1	ANN 3	55.0	0.0020	0.0195	Yes
SVC 1	SVC 4	55.0	0.0020	0.0176	Yes
SVC 1	SVC 5	55.0	0.0020	0.0156	Yes
SVC 2	ANN 3	45.0	0.0840	0.3359	No
SVC 2	SVC 4	41.0	0.0273	0.1367	No
SVC 2	SVC 5	53.0	0.0059	0.0410	Yes
ANN 3	SVC 4	40.0	0.2324	0.4648	No
ANN 3	SVC 5	42.0	0.1602	0.4805	No
SVC 4	SVC 5	27.5	0.5898	0.5898	No

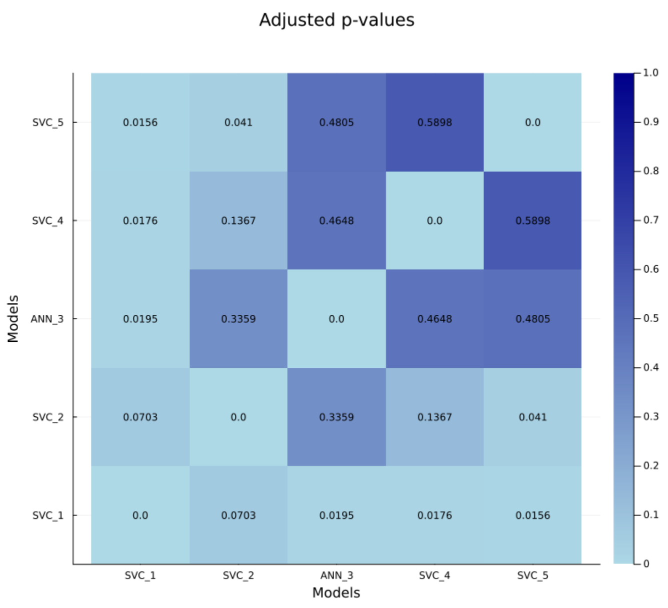


Figure 11: Heatmap of adjusted p-values (Wilcoxon test)

Paired Wilcoxon Test Results with Holm-Bonferroni Adjustment.

Conclusion. The results show that, although the difference between **SVM 1** and **SVM 2** is not statistically significant after adjustment, the differences between **SVM 1** and the other models are significant. Therefore, it is concluded that the best-performing classifier is the **degree-2 polynomial SVM with** $C = 1$, as it achieves better results without increasing computational cost compared to **SVM 2**.

4.4 Discussion

Based on the results obtained from the experiment, including the statistical significance tests, the strength of SVMs in handling classification tasks is evident. A careful hyperparameter search, for instance using a grid search to be more exhaustive, can yield powerful classifiers with a much lighter training process compared to neural networks or DoME.

Neural networks did not achieve such high performance, despite clearly being the second most powerful family of models. It is likely that experimenting with deeper and more complex networks could achieve higher performance, but this was not the focus of the current experiment.

On the other hand, the other models performed poorly in the classification task. As observed throughout this work, this is a complex dataset and likely not highly descriptive for the task of classifying songs into their respective musical genres.

5 Conclusions

In conclusion, this study addresses a problem with inherently complex nature, and the chosen data source may not have been optimal for tackling it. Nevertheless, the experiment highlighted both the limitations and the strengths of various fundamental machine learning models.

The main takeaway is the robustness of SVCs for this type of task, combined with the optimization achievable through careful hyperparameter tuning and efficient training, which facilitates experimentation with such models.

From a personal perspective, this project has taught us to experiment with maximum rigor, and above all, it has highlighted the inherent complexity of machine learning problems and their application in real-world scenarios. We have also gained insights into the music business by considering various possibilities and musical features that may define a song's genre.

Finally, we want to emphasize the interest generated by investigating prior work in this field, especially the combination of deep learning with classical models such as classification heads, in particular support vector machines. This aligns with part of our project and, coincidentally, with the best-performing models in this specific study.

6 Future Work

As evidenced by the review of related work and the results obtained in our experiments, music classification using machine learning techniques remains an open field with significant room for improvement.

There are multiple approaches to tackle this problem, as well as a growing range of models and techniques that can be applied. In our case, a clear direction for improvement lies in expanding our data sources, including not only metadata but also the audio signals themselves or song lyrics.

As our knowledge and technical skills advance, it would be interesting to explore more complex models, such as deep neural networks or multimodal approaches, with the aim of developing classifiers that are more robust and accurate in the face of the high variability inherent to music genres, styles, and fusions.

It is also worth noting the practical relevance of such models in the current context, where digital music consumption and production are massive, and their integration into audiovisual content—especially on social media—requires automatic and efficient solutions for music organization and recommendation.

References

[Cubillos, 2022] Cubillos, M. A. P. (2022). *Técnicas de selección de instancias en aprendizaje automático. Estudio y análisis empírico.* Trabajo de fin de máster, Universidad de Huelva, Huelva, Andalucía. Available at `https://www.uhu.es/mecofin/documents/docencia/tfm/20212022/MECOFIN_20212022_tfm_20212022-PCMA_TSIA.pdf`.

[Dammann and Haugh, 2017] Dammann, T. and Haugh, K. (2017). Genre classification of spotify songs using lyrics, audio previews, and album artwork. *CS229 Final Project, Stanford Univeristy, Fall.*

[Elbir and Aydin, 2020] Elbir, A. and Aydin, N. (2020). Music genre classification and music recommendation by using deep learning. *Electronics Letters*, 56(12):627–629.

[Guo et al.,] Guo, L., Gu, Z., and Liu, T. Music genre classification via machine learning.

[Meguenani et al., 2024] Meguenani, M. E. A., Britto Jr, A. d. S., and Koerich, A. L. (2024). Music genre classification using large language models. *arXiv preprint arXiv:2410.08321.*

[Ndou et al., 2021] Ndou, N., Ajoodha, R., and Jadhav, A. (2021). Music genre classification: A review of deep-learning and traditional machine-learning approaches. In *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–6.

[Oramas et al., 2018] Oramas, S., Barbieri, F., Nieto Caballero, O., and Serra, X. (2018). Multimodal deep learning for music genre classification. *Ubiquity Press.*

[Pelchat and Gelowitz, 2020] Pelchat, N. and Gelowitz, C. M. (2020). Neural network music genre classification. *Canadian Journal of Electrical and Computer Engineering*, 43(3):170–173.

[Rahardwika et al., 2020] Rahardwika, D. S., Rachmawanto, E. H., Sari, C. A., Susanto, A., Mulyono, I. U. W., Astuti, E. Z., Fahmi, A., et al.

(2020). Effect of feature selection on the accuracy of music genre classification using svm classifier. In *2020 International seminar on application for technology of information and communication (iSemantic)*, pages 7–11. IEEE.

[Spotify, 2024] Spotify (2024). Spotify web api documentation. Accessed: 2025-05-07.

[tomigelo, 2019] tomigelo (2019). Spotify audio features. Accessed: 2025-05-07.