

Mestrado em Engenharia Informática

Reconhecimento de Padrões

2022/2023

# Relatório

# Music Genre Classification

Miguel Faria | 2019216809



# Índice

Introdução.....	3
1. Pré-Processamento.....	4
1.1 Tratamento de dados.....	4
1.2 Seleção de Features.....	4
1.2.1 Teste de Kruskal-Wallis.....	5
1.2.2 Indicadores de Correlação.....	5
1.3 Redução de Features.....	6
1.3.1 Principal Component Analysis (PCA).....	6
1.3.2 Linear Discriminant Analysis (LDA).....	7
2. Classificadores.....	7
2.1 Minimum Distance Classifier.....	8
Conclusão.....	9

# Introdução

Este trabalho foi realizado no âmbito da cadeira de Reconhecimento de Padrões, enquadrada no Mestrado em Engenharia Informática da Universidade de Coimbra, e tem como principal objetivo desenvolver e analisar métodos que permitam classificar músicas consoante o seu estilo musical. Para isso, foi utilizado um *dataset* com 1000 músicas e mais de 190 *features*, que serve de base para a elaboração do projeto.

Nesta meta, foram considerados 2 cenários. No cenário A pretende-se realizar uma classificação binária, ou seja, adotando uma postura “um-contra-todos”. Assim, será feita a classificação das músicas de teste tendo em conta cada um dos gêneros musicais, determinando se elas pertencem ou não a esse estilo. No cenário B, é necessário classificar todas as diferentes categorias musicais em conjunto, o que torna o problema mais complexo.

De modo a realizar a classificação recorreu-se a vários métodos, nomeadamente, *Minimum Distance Classifier*, *Fisher LDA Classifier*, *Bayesian Classifier* e *k-NN Classifier*. No entanto, para poder aplicá-los, foi necessário realizar previamente algum processamento dos dados: divisão dos dados para treino e para testagem, normalização/standardização dos dados e redução/seleção de *features*, com recurso a testes *Kruskal-Wallis*, indicadores de correlação e *PCA*.

# 1. Pré-Processamento

## 1.1 Tratamento de dados

Os dados utilizados na resolução deste trabalho encontram-se no ficheiro “dados.csv”, presente no link fornecido no enunciado. Para facilitar o seu manuseamento, algumas das colunas foram alteradas de modo a poderem ser tratadas numericamente, nomeadamente as que envolvem as *labels* de classificação dos áudios (colunas “filename” e “label”). Deste modo, foi estabelecida a seguinte relação:

- Blues – 1
- Classical – 2
- Country – 3
- Disco – 4
- Hip-Hop – 5
- Jazz – 6
- Metal – 7
- Pop – 8
- Reggae – 9
- Rock – 10

Estes dados foram divididos de maneira a gerar diferentes *datasets* para a fase de treino e de teste, adotando uma separação de 80% e 20%, respetivamente. Para isso, foi tido em consideração que os *datasets* deveriam possuir um mesmo número de músicas referentes aos diversos gêneros. Os dados foram então colocados numa estrutura de dados igual à usada pelo STPRTool e estandardizados/normalizados, após remover as *features* sem variância. A separação dos dados foi feita antes de realizar qualquer tipo de alteração, pois assim é garantido que nenhuma característica dos dados que irão ser usados na testagem influêncie as modificações que irão ser aplicadas; desta maneira, as técnicas de seleção e redução de *features* que melhor se adaptaram ao *dataset* de treino vão ser replicadas no *dataset* de testagem.

Nesta fase inicial, foram realizados alguns testes para analisar características das *features* (matriz de correlação e histogramas de distribuição), no entanto, devido ao seu número bastante elevado, não foi possível retirar conclusões.

## 1.2 Seleção de Features

Como referido, os dados possuem demasiadas *features*, o que torna inviável a sua utilização nos classificadores, devido à sua redundância e irrelevância. Por essa razão, é

necessário aplicar técnicas de seleção, como o teste Kruskal-Wallis ou indicador de correlação, que ajudam a reduzir a dimensionalidade dos dados e a melhorar a eficiência computacional do modelo.

### 1.2.1 Teste de Kruskal-Wallis

O teste de Kruskal-Wallis é um teste não paramétrico, utilizado para verificar se há diferenças significativas entre as medianas de três ou mais grupos independentes de dados amostrais. A hipótese nula considerada neste teste é que as medianas dos grupos são iguais, enquanto a hipótese alternativa é que pelo menos uma das medianas é diferente das demais, usando como nível de significância o valor 0.05. Se o *p-value* obtido para uma determinada *feature* for menor do que o nível de significância, a hipótese nula é rejeitada; caso contrário, é aceite.

Para a análise das *features*, foram calculados os *ranks* do teste *KW*, apresentando também os respetivos valores de *chi-squared* e *p-value*. A tabela seguinte apresenta apenas alguns dos valores obtidos, devido ao elevado número de características:

rank	feature_ID	feature	chi_sq	p_value
1	7	chroma_stft_qt2	523.98	4.1932e-107
2	2	chroma_stft_mean	504.66	5.7658e-103
3	17	spectral_centroid_mean_max	495.9	4.337e-101
...				
194	43	harmony_mean_mean	41.735	3.6741e-06
195	130	mfcc_mean_10_min	41.356	4.309e-06
196	47	harmony_mean_median	39.081	1.1139e-05

Tabela 1 - Ranks obtidos com o teste KW

Uma vez que todos os *p-values* são baixos e que os valores *chi squared* não são bem separáveis, foi implementada uma técnica que seleciona apenas uma determinada percentagem das *features* com maior valor *chi squared* e *p-value* abaixo de 0.05; neste caso, essa percentagem definiu-se a 0.9, tendo selecionado 176 *features*.

### 1.2.2 Indicadores de Correlação

Com os dados selecionados com o teste KW, foram calculados os coeficientes de correlação, que permitem medir a relação entre as *features*. Bastantes valores eram elevados, pelo que se optou por selecionar as *features* que possuem coeficientes de correlação abaixo de 0.75, ficando com apenas 42 das 176 anteriores. Embora ainda seja um número alto, está ilustrada de seguida a matriz de correlação, onde se pode ver através da coloração que já não existe valores tão elevados:

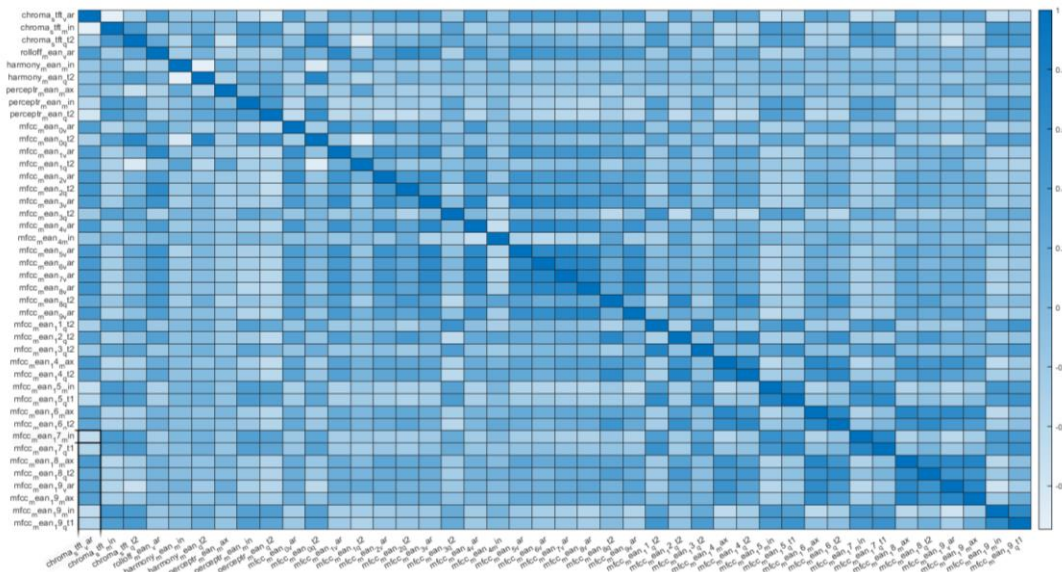


Figura 1 - Matriz de correlação entre as features selecionadas

## 1.3 Redução de Features

Para além das técnicas de seleção de *features*, também foram aplicados métodos de redução, particularmente PCA e LDA, de modo a descartar as características menos importantes, mantendo a estrutura essencial dos dados.

### 1.3.1 Principal Component Analysis (PCA)

O PCA (Principal Component Analysis) é uma técnica usada para reduzir a dimensão de um conjunto de dados de alta dimensionalidade, mantendo as informações mais relevantes. Após gerar o PCA Model, foram representados os Eigenvalues e a percentagem de variância relativas a cada uma das componentes principais, como ilustrado de seguida:

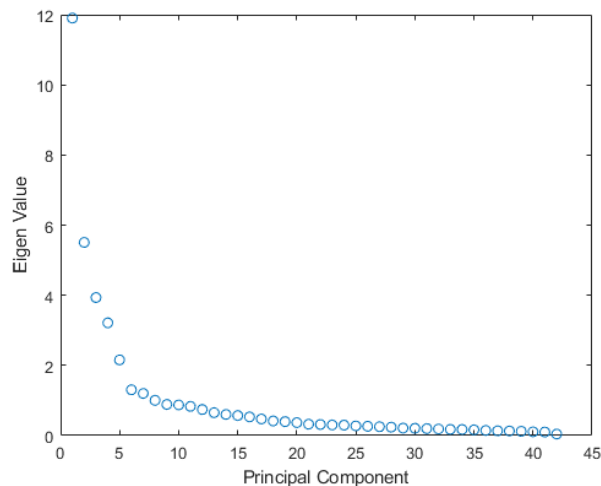


Figura 2 - Representação dos Eigen Values associados a cada uma das componentes principais

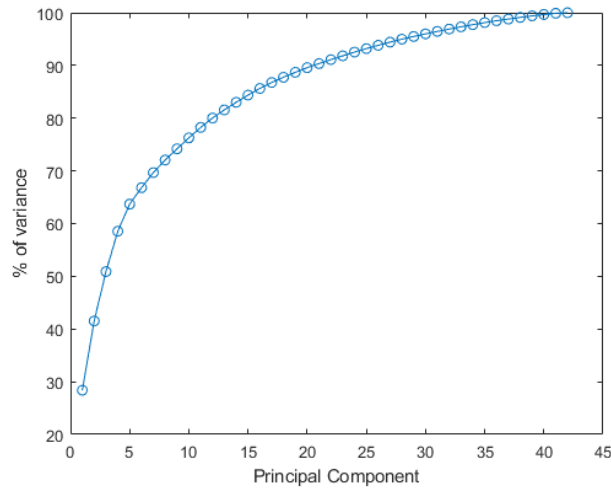


Figura 3 - Representação da percentagem de variância associada a cada uma das componentes principais

Após os testes, decidiu-se escolher as componentes principais que explicam pelo menos 0.95 da variância dos dados, que revelaram ser 29.

### 1.3.2 Linear Discriminant Analysis (LDA)

O LDA (Linear Discriminant Analysis) é também um método de redução de dimensionalidade, que visa encontrar uma transformação linear que maximize a separação entre classes num *dataset*. Este processo foi aplicado após o PCA, usando as mesmas componentes principais, 29.

## 2. Classificadores

Os classificadores têm como objetivo determinar a que classe cada um dos dados de teste pertence, com base nos seus atributos. Neste estudo, serão utilizados cinco classificadores diferentes: Minimum Distance Classifier, Fisher LDA Classifier, Bayesian Classifier, k-NN Classifier, com o objetivo de comparar e avaliar a eficiência de cada um na classificação de géneros musicais.

O seu desempenho irá ser medido através de métricas, tais como:

- $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- $Sensitivity = \frac{TP}{TP+FN}$
- $Specificity = \frac{TN}{TN+FP}$
- $F - measure = 2 \times \frac{Accuracy \times Sensitivity}{Accuracy + Sensitivity}$

## 2.1 Minimum Distance Classifier

Este classificador baseia-se na distância euclidiana para determinar a classe de uma amostra, atribuindo, para cada caso de teste, a classe que se encontra a menor distância do vetor médio de características. Os resultados obtidos encontram-se nas tabelas seguintes:

Minimum Distance Classifier – Cenário A				
Género	Accuracy	Sensitivity	Specificity	F-measure
Blues	0,61089	0,81667	0,5	0,785
Classical	0,98477	0,96667	1	0,97
Country	0,83571	0,76667	0,9	0,78
Disco	0,67675	0,65	0,7	0,655
Hip-Hop	0,68014	0,80556	0,6	0,785
Jazz	0,7	0,86667	0,6	0,84
Metal	0,83582	0,88333	0,8	0,875
Pop	0,96907	0,93333	1	0,94
Reggae	0,6965	0,85556	0,6	0,83
Rock	0,71053	0,66667	0,75	0,675

*Tabela 2 – Métricas de desempenho usando Minimum Distance Classifier para o cenário A*

Minimum Distance Classifier – Cenário B				
Género	Accuracy	Sensitivity	Specificity	F-measure
Blues	0,88	0,25	0,95	0,38938
Classical	0,86087	0,57143	0,91282	0,6869
Country	0,82569	0,48276	0,87831	0,60928
Disco	0,82843	0,31818	0,89011	0,45977
Hip-Hop	0,89583	0,0625	0,97159	0,11685
Jazz	0,88679	0,42308	0,95161	0,57285
Metal	0,85922	0,34783	0,9235	0,49519
Pop	0,86486	0,51613	0,92147	0,64646
Reggae	0,85922	0,34783	0,9235	0,49519
Rock	0,80882	0,31818	0,86813	0,4567

*Tabela 3 – Métricas de desempenho usando Minimum Distance Classifier para o cenário B*



## Conclusão

Para o classificador usado, os resultados foram razoáveis. Os processos de seleção e redução de *features* permitiram melhorar a sua *performance*, mesmo com dimensionalidade mais baixa.

O objetivo do trabalho não foi atingido, uma vez que deveriam ter sido testados diversos outros classificadores. Pessoalmente, a falta de destreza com MatLab e a documentação de difícil compreensão tornaram o processo de resolução de problemas mais demorado, nomeadamente na correção de erros da primeira meta, o que contribuiu para a incompletude do trabalho.