

Classificação de Gênero Musical com técnicas de Machine Learning

Milena de Matos Siqueira

Ciência da Computação, UNIFESP
São José dos Campos, São Paulo, Brasil
milena.matos@unifesp.br

Resumo— Este artigo tem como objetivo aplicar os conhecimentos e alguns algoritmos estudados em Inteligência Artificial, sendo eles KNN, SVC, e Naive Bayes. O conjunto de dados escolhido foi “Top Hits Spotify from 2000-2019”

Palavras-chave— Inteligência Artificial, Machine Learning, Python

I. INTRODUÇÃO

A Inteligência Artificial tem crescido muito nos últimos anos, e chamando cada vez mais atenção de estudantes e profissionais de diversas áreas. Uma de suas técnicas é Aprendizado de Máquina (“Machine Learning”), que será utilizada nesse projeto, mais especificamente, Aprendizado de Máquina Supervisionado.

II. OBJETIVOS

Para conseguir aplicar os conceitos estudados, foi selecionada uma base de dados de tamanho médio (2000 linhas) que contém diversas informações independentes, tais como índice de energia, batidas por minuto (BPM) e dançabilidade de uma música. Com base nisso, queremos que o algoritmo seja capaz de prever o gênero musical (pop, country, R&B, entre outros).

III. METODOLOGIA

Para atingir o objetivo, foram selecionados 3 algoritmos de Machine Learning:

A. KNN - K Nearest Neighbor

Este método calcula a distância entre os elementos (geralmente Distância Euclidiana é utilizada) e seleciona os “k” vizinhos mais próximos. O parâmetro “k” escolhido impacta no resultado das previsões, podem ser testados alguns valores até atingir o desejado.

B. SVC - Support Vector Classifier

Este método utiliza as diferenças de posição e disposição no espaço dos atributos para classificar os pontos. Para isto é obtido um objeto geométrico separador de classes, no caso deste projeto, uma linha, pois foi utilizado a função de kernel linear.

C. Naive Bayes

Classificador probabilístico baseado no Teorema de Bayes, leva o “ingênuo” (naive) no nome por desconsiderar correlação entre as variáveis, tratando-as de forma independente.

IV. PRÉ-PROCESSAMENTO

A base de dados “Top Hits Spotify from 2000-2019” completa contém 2000 linhas e 18 colunas. Das 2000 linhas, 70% será separada para treino e 30% para teste. E as colunas serão analisadas e filtradas, pois nem todas as informações serão úteis para os algoritmos.

De imediato foram removidas as colunas “artist” (nome do artista) e “song” (nome da música) do conjunto de dados, visto que são informações muito distintas e únicas. Em seguida foram removidas “year” (ano de lançamento), “duration_ms” (duração da música em milissegundos) e “popularity” (índice de popularidade). Esses dados não irão auxiliar na classificação. Na figura [1], o formato final da base de dados.

```
genre_classifier/analysis.py
explicit danceability energy ... valence tempo genre
0 False 0.751 0.834 ... 0.894 95.053 pop
1 False 0.434 0.897 ... 0.684 148.726 rock, pop
2 False 0.529 0.496 ... 0.278 136.859 pop, country
3 False 0.551 0.913 ... 0.544 119.992 rock, metal
4 False 0.614 0.928 ... 0.879 172.656 pop

[5 rows x 13 columns]
genre count
```

Fig. 1 Prévia da base de dados após tratamento e remoção de colunas inadequadas

Analizando os diferentes valores da coluna “genre”, que desejamos classificar, foi observado que grande parte das músicas se encaixam em mais de um gênero, o que acabou gerando 58 classes, uma variedade muito alta para o objetivo do projeto. Na figura [2], as 5 classes mais frequentes:

	genre	count
0	pop	428
1	hip hop, pop	277
2	hip hop, pop, R&B	244
3	pop, Dance/Electronic	221
4	pop, R&B	178

Fig. 2 As 5 combinações de gêneros mais frequentes na base

Para simplificar a análise, a alternativa escolhida para contornar esse problema foi considerar apenas o primeiro gênero da lista como principal. Dessa forma, a quantidade de classes foi reduzida para 11, conforme figura [3].

	genre	count
0	pop	936
1	hip hop	776
2	rock	162
3	Dance/Electronic	41
4	latin	15
5	R&B	13
6	country	11
7	World/Traditional	10
8	metal	9
9	Folk/Acoustic	4
10	easy listening	1

Fig. 3 Contagem dos gêneros considerando apenas o primeiro

V. ANÁLISE EXPLORATÓRIA DOS DADOS

Antes de iniciar a implementação, foram realizadas algumas análises a fim de conhecer um pouco melhor a distribuição dos dados. Como muitas variáveis serão utilizadas, um gráfico útil é o de Confusão da Matriz, conforme figura [4]. As variáveis utilizadas são:

- explicit: Indica se a música contém uma ou mais letra considerada ofensiva ou inapropriada para crianças.
- danceability: Valor entre 0 e 1 que indica o quão adequada a música é para dançar.
- energy: Valor entre 0 e 1 representando a intensidade e atividade da música.
- key: tonalidade da música

- loudness: volume da música em decibéis.
- mode: Valor entre 0 e 1 indicando o modo da música, sendo 0 o menor e 1 maior.
- speechiness: Presença de vocais na música (valores entre 0 e 1)
- acousticness: Medida de acústico na música, também valores entre 0 e 1
- instrumentalness: Valores entre 0 e 1 que prevêem a ausência de vocais
- valence: Indica a positividade da música com valores de 0 até 1
- tempo: Valor estimado das BPM (batidas por minuto)
- genre: Gênero musical

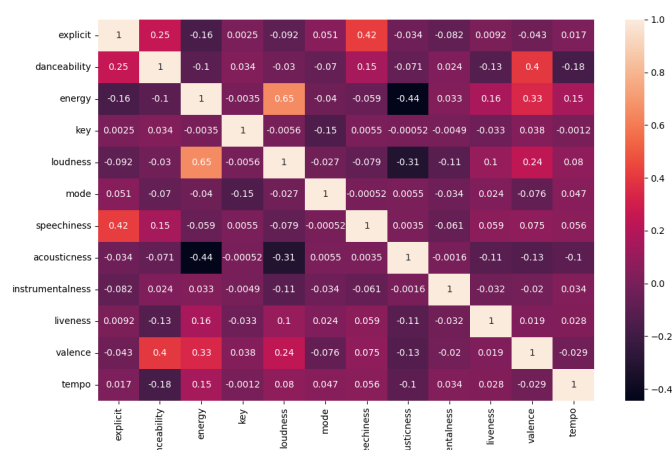


Fig. 4 Matriz de correlação de todas as variáveis, exceto gênero, que queremos prever, utilizadas no estudo

Algumas correlações que podem ser destacadas são:

- 1) “acousticness” e “energy”: Os índices de acústico e energia são inversamente proporcionais, possuindo -0.44 de relação.
- 2) “loudness” e “energy”: Volume e energia estão 0.65 relacionados.
- 3) “speechiness” e “explicit”: Quantidade de falas e conteúdo explícito na música também estão relacionados em 0.42.

Analizando “loudness”, “energy” e “genre”, temos a seguinte distribuição na figura [5], onde é possível notar a concentração de músicas acompanhando o volume e energia, entretanto os gêneros não estão muito dispersos, o que pode ser um ponto de atenção para os resultados da previsão.

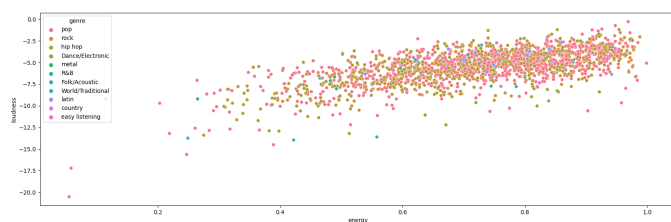


Fig. 5 Distribuição dos gêneros de acordo com energia e volume da música

VI. IMPLEMENTAÇÃO

Partindo para a implementação, foram utilizadas as seguintes tecnologias:

- 1) *Visual Studio Code*: ferramenta utilizada para edição e execução do código.
- 2) *Python*: linguagem de programação escolhida por possuir diversas bibliotecas e funções úteis, desde a plotagem dos gráficos até a manipulação dos dados e predição do modelo. As principais bibliotecas que foram essenciais para a implementação foram:

A. Pandas

Biblioteca utilizada para manipular e analisar o “dataset” (conjunto de dados lido do arquivo .csv). Ela possibilita realizar queries, agrupamentos e obter correlações em cima dos dados.

B. Matplotlib e Seaborn

Utilizadas para criação dos gráficos em geral, sendo que a Seaborn é baseada no Matplotlib. A combinação das duas possibilitou a geração de gráficos mais complexos, tais como a Matriz de Confusão.

C. Scikit-learn

Disponibiliza todas as funcionalidades de Machine Learning que foram necessárias para a construção dos modelos. A biblioteca é de fácil acesso e uso, sendo possível encontrar diversos conteúdos e suporte na comunidade.

Sendo assim, foi utilizada uma função da biblioteca “sklearn” (Scikit-learn) para normalizar os dados do eixo X, que contém todas aquelas 13 colunas já filtradas no pré-processamento, exceto pelo gênero (prévia na figura [6]). O eixo Y terá apenas a coluna gênero.

```
[0. 0.73522459 0.82523038 ... 0.40108238 0.91549898 0.23227167]
[0. 0.36052009 0.8919606 ... 0.71016236 0.69087603 0.58811791]
[0. 0.47281324 0.46721746 ... 0.27600722 0.25660498 0.50944097]
...
[0. 0.84869976 0.65999364 ... 0.30366807 0.82671944 0.25170388]
[0. 0.72340426 0.49263849 ... 0.24113049 0.33040967 0.28494617]
[0. 0.66903073 0.74896727 ... 0.07793145 0.55075409 0.39794606]
```

Fig. 6 Prévia dos dados após normalização

Conforme mencionado também no pré-processamento, foi utilizado a função “train_test_split” para separar os dados em treino e teste, sendo 70% e 30% respectivamente.

Para cada algoritmo foi criado um arquivo Python que importa a lib (biblioteca) necessária, realiza o treinamento, predição, e mostra na tela os resultados obtidos. Foram criados também arquivos auxiliares para preparar os dados e interpretar os resultados obtidos. Na figura [7], organização do projeto, e na figura [8], exemplo de funcionamento de um dos algoritmos.

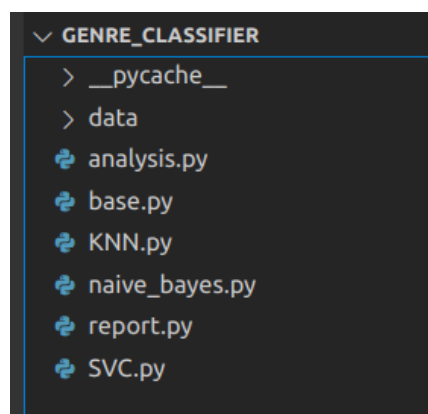


Fig. 7 Organização final do projeto, através de imagem da ferramenta Visual Studio Code

```
from sklearn.svm import SVC
from base import get_normalized_data
from report import print_report

X_train, X_test, y_train, y_test = get_normalized_data()

svc = SVC(kernel="linear")
svc.fit(X_train, y_train)

y_predict_SVC = svc.predict(X_test)
print_report(svc, X_test, y_test, y_predict_SVC)
```

Fig. 8 Imagem ilustrando trecho de código do método SVC, contendo importação das libs e utilização das funções “fit” e “predict” do Scikit-learn e “print_report” do projeto

VII. RESULTADOS

A. Método KNN

Foi utilizado um k igual 5 e obtido cerca de 61% de acurácia. Mais detalhes da classificação na figura [9] e na figura [10] a respectiva matriz de confusão.

	precision	recall	f1-score	support
Dance/Electronic	0.00	0.00	0.00	13
Folk/Acoustic	0.00	0.00	0.00	2
R&B	0.00	0.00	0.00	2
World/Traditional	0.00	0.00	0.00	4
country	0.00	0.00	0.00	2
hip hop	0.64	0.64	0.64	232
latin	0.00	0.00	0.00	7
metal	0.00	0.00	0.00	2
pop	0.61	0.73	0.67	284
rock	0.33	0.15	0.21	46
accuracy			0.61	594
macro avg	0.16	0.15	0.15	594
weighted avg	0.57	0.61	0.59	594

Fig. 9 Detalhes dos resultados obtidos com o método KNN.

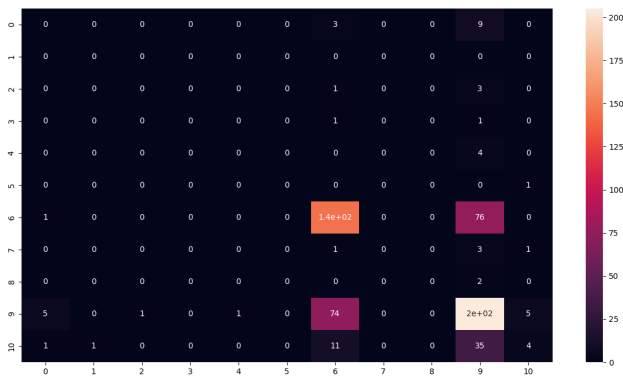


Fig. 10 Matriz de confusão obtida com o método KNN. Podemos destacar os resultados para os gêneros 6 (hip hop) e 9 (pop).

B. Método SVC

A função utilizada foi a linear, e a acurácia obtida foi cerca de 64%. Mais detalhes da classificação na figura [11] e na figura [12] a respectiva matriz de confusão.

	precision	recall	f1-score	support
Dance/Electronic	0.00	0.00	0.00	10
Folk/Acoustic	0.00	0.00	0.00	2
R&B	0.00	0.00	0.00	4
World/Traditional	0.00	0.00	0.00	4
country	0.00	0.00	0.00	1
hip hop	0.81	0.57	0.67	242
latin	0.00	0.00	0.00	5
metal	0.00	0.00	0.00	3
pop	0.58	0.90	0.70	271
rock	0.00	0.00	0.00	52
accuracy			0.64	594
macro avg	0.14	0.15	0.14	594
weighted avg	0.59	0.64	0.59	594

Fig. 11 Detalhes dos resultados obtidos com o método SVC.



Fig. 12 Matriz de confusão obtida com o método SVC. Podemos destacar novamente os resultados para os gêneros 6 (hip hop) e 9 (pop).

C. Método Naive Bayes

A acurácia obtida não ficou muito distante das anteriores, mas foi a maior, com cerca de 66%.

	precision	recall	f1-score	support
Dance/Electronic	0.00	0.00	0.00	15
Folk/Acoustic	0.00	0.00	0.00	1
R&B	0.00	0.00	0.00	3
World/Traditional	0.00	0.00	0.00	1
country	0.00	0.00	0.00	2
easy listening	0.00	0.00	0.00	1
hip hop	0.77	0.60	0.68	225
latin	0.00	0.00	0.00	3
metal	0.00	0.00	0.00	3
pop	0.62	0.88	0.73	293
rock	0.00	0.00	0.00	47
accuracy			0.66	594
macro avg	0.13	0.13	0.13	594
weighted avg	0.60	0.66	0.61	594

Fig. 13 Detalhes dos resultados obtidos com o método KNN.

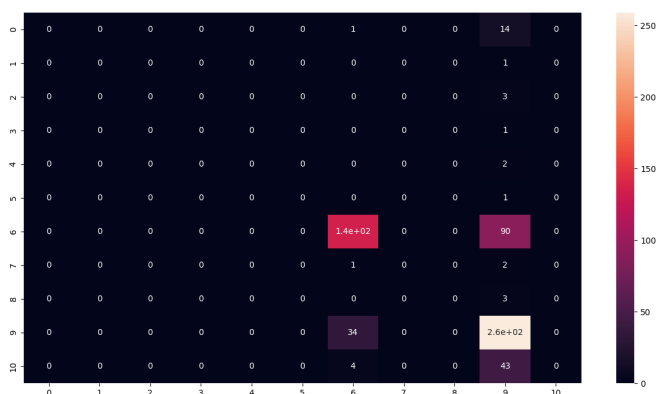


Fig. 14 Matriz de confusão obtida com o método Naive Bayes.

VIII. CONCLUSÃO

Com os resultados obtidos, podemos observar que a diferença entre os algoritmos é baixa, porém o algoritmo que obteve a maior taxa de acertos foi o Naive Bayes. Isso pode ser por conta do fato de que ele não assume que os atributos têm alguma relação, e conforme visto no tópico de pré-processamento, pelo menos alguns pares de atributos tem alguma relação, como o de volume e energia. Mas de fato, se olharmos para a distribuição de gênero da figura [5], notamos que não há muita relação entre o gênero de acordo com volume/energia.

O que é possível observar também é que grande parte da taxa de acertos estava concentrada nos gêneros pop e hip hop, que são os 2 mais frequentes na base (vide figura [3]). Se a base tivesse gêneros melhores distribuídos, o resultado poderia ser um pouco melhor.

REFERENCES

- [1] [1] HUANG, J.; LING, C. X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, v. 17, n. 3, p. 299-310, Mar, 2005, doi: 10.1109/TKDE.2005.50.
- [2] PAHWA, K.; AGARWAL, N. Stock Market Analysis using Supervised Machine Learning. In: *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 1., 2019, Faridabad. *Anais eletrônicos*. Faridabad: IEEE, 2019. p. 197-200.
- [3] SARAVANAN, R.; SUJATHA, P. A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification. In: *Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2., 2018, Madurai. *Anais eletrônicos*. Madurai: IEEE, 2019. p. 945-949.
- [4] "Top Hits Spotify from 2000-2019" Mark Koverha
- [5] MARK KOVERHA. (2022) Top Hits Spotify from 2000-2019. [Online]. Available: <https://www.kaggle.com/datasets/paradisejoy/top-hits-spotify-from-20002019>

- [6] TIAGO B. MARINHO. (2022) Atividade K-means com PCA. [Online]. Available: <https://www.kaggle.com/code/tiagobmarinho/atividade-k-means-com-pca-tiago-bezerra-marinho>
- [7] CHANCE V. (2022) Spotify SVM Analysis w/ Gridsearch. [Online]. Available: <https://www.kaggle.com/code/chancev/spotify-svm-analysis-w-gridsearch>
- [8] JOÃO P. SANTOS. (2022) Trabalho de IRA. [Online]. Available: <https://www.kaggle.com/code/laamze/trabalho-de-ira>