Lucas Balieiro Maciel, 800534
Departamento de Computação
UFSCar
lucas.balieiro@estudante.ufscar.br

Rafael Naoki Arakaki Uyeta, 800207
Departamento de Computação
UFSCar
rafael.uyeta@estudante.ufscar.br

Paula Vitoria Martins Larocca, 769705
Departamento de Computação
UFSCar
paula.larocca@estudante.ufscar.br

Vinícius Gonçalves Perillo, 800219
Departamento de Computação
UFSCar
vinicius.perillo@estudante.ufscar.br

Resumo - O presente documento tem como objetivo discutir a respeito de um modelo de classificação de gênero musical.

I. INTRODUÇÃO

A música é uma das formas de artes mais antigas que a humanidade conhece. Possui inúmeros gêneros musicais diferentes, e representativos de cada cultura . Na base de dados que será utilizada neste estudo, temos diversas informações de diferentes músicas, como por exemplo o artista, o nome do álbum, dentre outras informações.

Assim, esse trabalho visa utilizar técnicas de aprendizado de máquina para classificar corretamente o gênero musical das músicas, através dos atributos contidos na base utilizada.

II. BASES DE DADOS

Para a aplicação dos algoritmos de classificação de aprendizado de máquina, utilizamos uma base de dados com as seguintes informações:

- **track_id:** É um identificador da tupla;
- **artista:** Nome do artista que compôs a faixa musical;
- **album_name:** Nome do álbum em que a música aparece;
- **track_name:** Nome da faixa musical;
- **popularity:** Consiste em um valor entre 0 e 100 (sendo este o mais popular), o qual é calculado através de um algoritmo que utiliza principalmente o número de vezes que determinada faixa foi tocada e o quão recente ela foi tocada. Assim, faixas muito tocadas antigamente recebem uma baixa popularidade quando comparadas com faixas que são muito tocadas atualmente.
- **duration_ms:** O tamanho da música em milisegundos;
- **explicity:** Se a música possui sua letra explícita para o ouvinte acompanhar.(true = sim, false = não ou desconhecido);
- **danceability:** Consiste em um valor de 0 a 1, sendo este o mais "dançável", que representa o quanto determinada música é propícia para dançar;
- **energy:** Consiste em um valor entre 0 e 1, sendo este o mais energético, que representa uma medida de intensidade e atividade que determinada música possui;
- **key**: É a nota base que música utiliza (por exemplo, se o valor for 0 significa que a música foi escrita com base na nota dó)
- **loudness:** O nível de barulho da faixa musical em decibéis;
- **mode:** Indica se a música está em escala maior ou menor;
- **speechiness:** Detecta a presença de palavras faladas na faixa analisada (quanto mais próximo de 1, mais provável de a faixa ser um *podcast*, um *talk show*, enquanto mais perto de 0 é muito provável de ser uma música instrumental);
- **acousticness:** Consiste em uma medida de 0 a 1, sendo que quanto mais próximo de 1 a música é mais acústica (sem presença de instrumentos elétricos ou amplificações sonoras);
- **instrumentalness:** Consiste em uma medida de 0 a 1, sendo que quanto mais próximo de 1, a música apresenta menos vocalização, sendo uma música instrumental ("ooh" e "aah" são considerados sons instrumentos nesse caso);
- **liveness:** Consiste em uma medida de 0 a 1 que capta a presença de audiência na faixa musical. Dessa forma, quanto mais próximo de 1, maior a chance de que essa faixa musical tenha sido armazenada em uma apresentação ao vivo;

- **valence:** Consiste em uma medida de 0 a 1 que representa o quão positiva (alegre, eufórica) ou negativa (triste, depressiva) é a faixa musical, sendo que quanto mais próximo de 1 mais positiva é a música
- **tempo:** Representa o ritmo da música em bpm (batidas por minuto);
- **time_signature:** Consiste em uma medida aproximada de número de notas, ou batidas, em um compasso (por exemplo, se o valor for 7, isso significa que, em média, há 7 batidas em um compasso);
- **track_genre:** Indica o gênero musical que determinada faixa musical pertence;

A base de dados utilizada pode ser encontrada no link. Ela contém 114000 instâncias e 114 gêneros musicais diferentes. Esse *dataset* foi criado por Maharshi Pandya, e é de uso livre e aberto, sendo recomendado para criação de "Sistemas de recomendação", baseados em valores de entrada de algum usuário qualquer, ou então para "Classificação" dos gêneros musicais através das demais informações contidas na base de dados.

```
Data columns (total 20 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   track_id          114000 non-null  object
 1   artists           113999 non-null  object
 2   album_name        113999 non-null  object
 3   track_name        113999 non-null  object
 4   popularity        114000 non-null  int64
 5   duration_ms       114000 non-null  int64
 6   explicit          114000 non-null  bool
 7   danceability      114000 non-null  float64
 8   energy            114000 non-null  float64
 9   key               114000 non-null  int64
 10  loudness          114000 non-null  float64
 11  mode              114000 non-null  int64
 12  speechiness       114000 non-null  float64
 13  acousticness      114000 non-null  float64
 14  instrumentalness  114000 non-null  float64
 15  liveness          114000 non-null  float64
 16  valence           114000 non-null  float64
 17  tempo             114000 non-null  float64
 18  time_signature    114000 non-null  int64
 19  track_genre       114000 non-null  object
dtypes: bool(1), float64(9), int64(5), object(5)
```

Imagem 1: *DataSet* com todos os atributos

Após entender melhor o que cada atributos significa, retiramos alguns atributos que não são relevantes para a classificação do gênero, como o nome da faixa e seu ID, visto que eles são apenas identificadores. Também foi retirado o nome do artista e do álbum para evitar um possível *overfitting*, visto que seria possível que o algoritmo aprendesse a classificar o gênero a partir dessas informações, e caso um artista novo, ou álbum, aparecesse, ele já não seria mais capaz de realizar a classificação corretamente. Após reestruturar a base de dados principal, foi criada uma matriz de correlação a fim de entender melhor o comportamento desses atributos quando relacionados entre si.

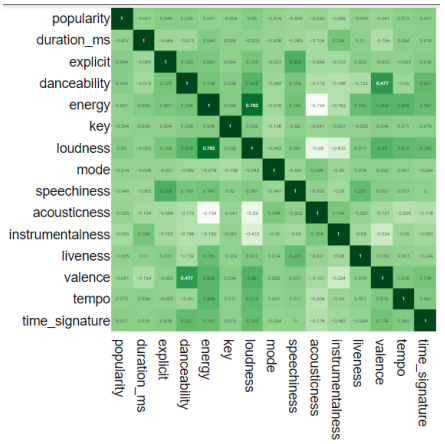

Imagem 2: matriz de correlação dos dados

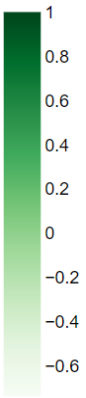

Imagem 3: Legenda das cores da matriz

Após realizar uma análise dessa matriz, é possível verificar que os atributos *energy* e *loudness*, *acousticness* e *energy* são os que apresentam maior correlação, sendo respectivamente 0.782 e -0.754. Isso indica que *energy* e *loudness* estão muitos relacionados positivamente, ou seja, se uma música é muito energética então ela tende a ser mais barulhenta, enquanto a correlação entre *acousticness* e *energy* expressa um valor negativo, significando que quanto mais energética a música, menor a presença do instrumento

puro, de forma que a presença de instrumentos eletrônicos, ou com amplificação sonora, se torna mais expressiva.

III. SUBCONJUNTO DE DADOS CRIADOS

A base de dados é bastante extensa, contendo mais de 110 gêneros musicais diferentes, e como o "gênero" será utilizado como a classe para o algoritmo de classificação, optou-se pela redução da base em dois conjuntos distintos com 3 gêneros cada. O primeiro conjunto será denominado *Diff_spf*, sendo composta por gêneros sonoramente diferentes. O segundo, será denominado *Sim_spf*, composto por gêneros sonoramente semelhantes.

III.I. HIPÓTESE CRIADA

A hipótese que será testada neste trabalho é de que os algoritmos não terão dificuldade de classificar os tipos musicais mais distintos, no caso a base *Diff_spf*, contudo ela terá dificuldade em realizar a classificação quando os tipos musicais forem mais semelhantes, como ocorre na base *Sim_spf*.

III.II. CONJUNTO DE DADOS DIFF_SPF

Para a base *diff_spf*, inicialmente pensamos em escolher os gêneros pop, rock e clássica, já que é relativamente fácil diferenciar esses estilos ao ouvir eles. Porém, ao realizar uma análise dos valores dos atributos, foi possível ver que apesar de o pop e o rock serem facilmente identificados ao ouvir, eles são muito semelhantes na análise de seus dados, como as duas próximas imagens demonstram.

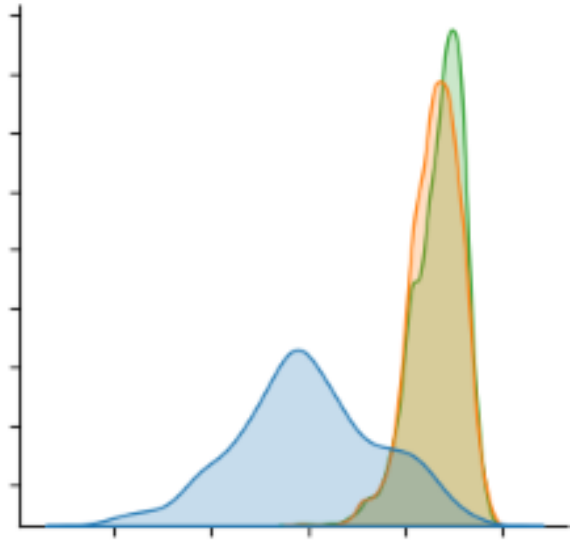

Imagem 4: Gráfico de *loudness* x *loudness*

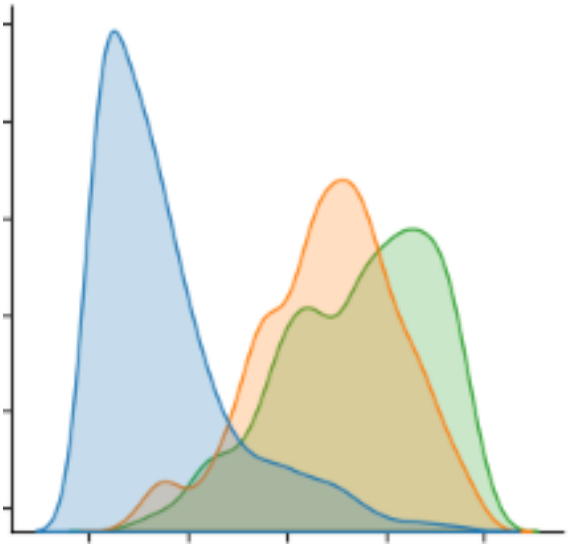

Imagem 5: Gráfico de *energy* x *energy*

Em ambos os gráficos, o azul representa o estilo Clássico, o amarelo representa o estilo Pop e o ver o estilo Rock. Esses dois gráficos demonstram o quão semelhantes os dados de pop e rock são, ao mesmo passo em que o estilo clássico acaba destoando um pouco deles. Tendo isso em vista, alteramos os gêneros que seriam utilizados para compor esse subconjunto de dados.

Por fim, escolhemos os gêneros jazz, hip-hop e clássica, pois uma análise mais a fundo desses 3 gêneros mostrou o quão diferentes eles são, apesar de ainda apresentarem semelhanças em algumas ocasiões.

Feita a criação da base *diff_spf*, foi criada uma matriz de correlação com os gêneros

selecionados, para entender como cada atributo se relaciona com cada um dos gêneros selecionados.

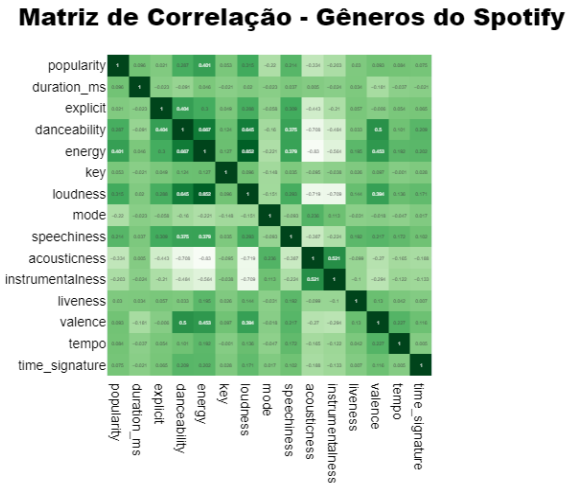

Imagem 6: Gráfico de correlação com os 3 gêneros selecionados

Após analisar a matriz de correlação, é possível visualizar semelhanças com a matriz de correlação com todos os gêneros da base.. Os atributos *loudness* e *energy* continuam tendo a maior correlação, contudo os atributos *energy* e *danceability* passaram a apresentar um valor elevado de correlação também, enquanto *energy* e *acousticness* continuam tendo a maior correlação negativa, mas os atributos *acousticness* e *danceability*, *instrumentalness* e *loudness*, também passaram a apresentar uma correlação mais significativa. Isso demonstra principalmente que, para os gêneros selecionados, o *danceability* e o *instrumentalness* passam a ser mais relevantes do que quando analisamos todos os gêneros de uma vez.

III.III. CONJUNTO DE DADOS SIM_SPF

Analogamente, após a criação da base de dados *sim_spf*, também foi criada a matriz de correlação dos atributos em relação aos 3 gêneros selecionados. Nessa base de dados, o objetivo é avaliar o desempenho dos algoritmos quando são submetidos a 3 gêneros extremamente semelhantes. Aqui serão utilizados o gênero Pop, Indie-Pop e K-Pop.

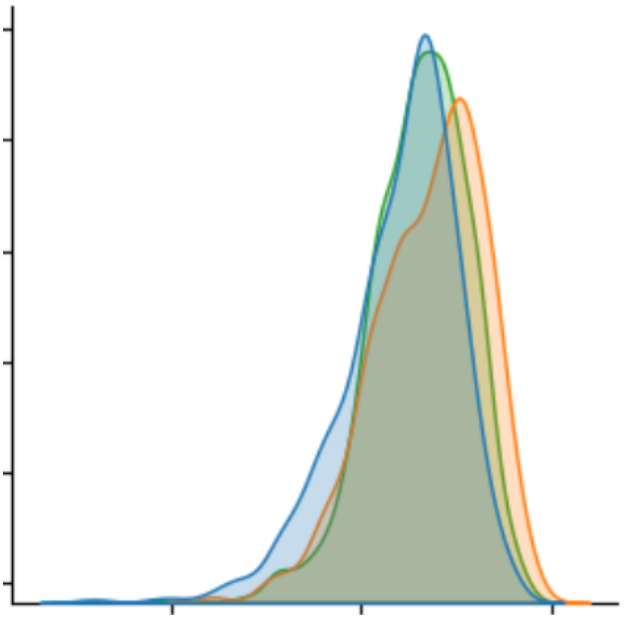

Imagem 7: Gráfico de *loudness* x *loudness*

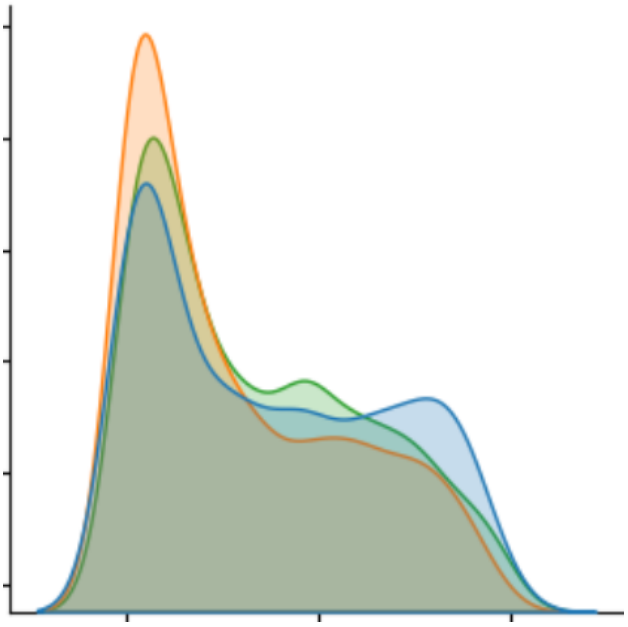

Imagem 8: Gráfico de *acousticness* x *acousticness*

Através desses gráficos, é possível perceber o quão semelhantes são esses modelos, e é curioso que eles são tão semelhantes que a distribuição deles, o formato da onda, chega até a ser semelhante. Assim, a escolha dos gêneros foi condizente com o objetivo da base.

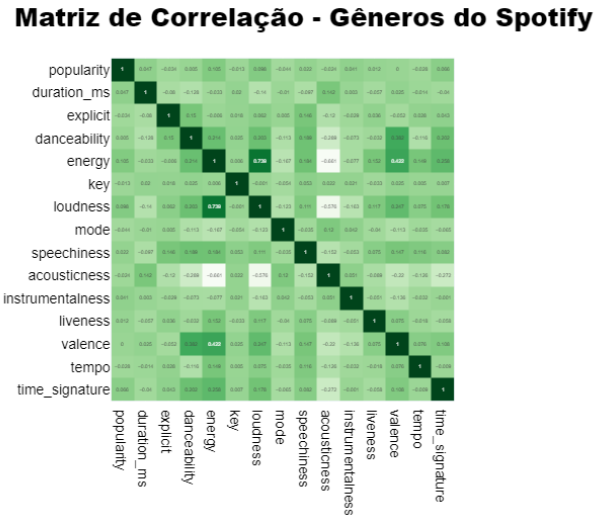

Imagem 9: Matriz de correlação com os 3 gêneros selecionados

IV. MODELOS DE APRENDIZADO DE MÁQUINA UTILIZADOS

Para realizar a classificação das faixas musicais de acordo com os gêneros, serão utilizados os modelos de *Naive Bayes* e *Decision Tree*.

IV.I NAIVE BAYES

O *Naive Bayes* é um algoritmo supervisionado, o qual tem como objetivo realizar classificações, ou previsões, com base no teorema de *Bayes*. Esse teorema necessita que os atributos utilizados não contenham nenhuma correlação, mas o *Naive Bayes*, ou "bayes ingênuo", utiliza esse teorema de forma a assumir que os atributos são independentes, ou seja, que não há nenhum tipo de correlação entre eles. Por conta dessa suposição que ele é conhecido como "bayes ingênuo", visto que ela raramente é verdadeira na prática.

IV.II. ÁRVORE DE DECISÃO

A árvore de decisão é outro algoritmo supervisionado utilizado para realizar classificação, além de regressão. Seu funcionamento consiste em cada nó interno representar uma característica, ou atributo, cada ramo representa uma decisão frente ao atributo selecionado no nó e cada folha representa o resultado final da classificação, ou seja, a classe final. O objetivo é dividir a base de dados em hiperplanos homogêneos, ou seja, que dentro deles contenha a menor variação possível de classes, de forma a obter a melhor eficiência de classificação possível.

V. APLICAÇÃO DOS MODELOS DE APRENDIZADO DE MÁQUINA NA BASE DIFF_SPF

Os modelos mencionados foram testados exaustivamente, mudando alguns parâmetros específicos de cada algoritmo (como os atributos no *Naive Bayes*, ou então a altura na "Árvore de Decisão"), visando resultados estáveis.

V.I. APLICAÇÃO NAIVE BAYES

Em primeiro lugar, importante ressaltar que o *Naive Bayes* foi o algoritmo escolhido inicialmente, visto que a base de dados escolhida possui muitos atributos não relacionados, como pode ser visto na imagem 9, e ele trabalha muito bem com atributos não relacionados.

Inicialmente, foi feita uma análise com todos os atributos da base, lembrando que o ID, o nome da música, o nome do artista e do álbum já foram retirados da base, e com diferentes porcentagem de treinamento e teste, a fim de identificar a proporção que apresenta melhor desempenho. Para cada porcentagem selecionada, foram realizadas 100 instâncias de testes, com o objetivo de verificar a consistência apresentada.

```
40%
0.7301000000000002
0.019518965136502513

35%
0.7293
0.01834420889545255

30%
0.7316999999999999
0.018923794545492204

25%
0.7342
0.020257344347174452

20%
0.7372
0.021216974336601357
```

Imagem 10: Porcentagem de teste, precisão média e variância

A partir da imagem acima, é possível ver que a alteração na proporção de teste e treinamento não foi significativa na precisão do modelo e nem na variância, pois os resultados obtidos são muito próximos. Tendo isso em vista, a proporção de 25% para teste e 75% para treinamento foi escolhida para prosseguir nas demais análises.

Depois de realizar a análise com todos os atributos, foram selecionados manualmente alguns atributos para ver qual seria o novo desempenho do modelo. Os atributos escolhidos foram *explicit, danceability, energy, loudness, acousticness* e *instrumentalness*, por conta de serem atributos relevantes para a classificação do gênero musical. Nesse caso, é de se esperar que o desempenho do algoritmo não seja elevado, visto que contém atributos com um nível de correlação mais elevado.

	precision	recall	f1-score	support
classical	0.85	0.82	0.83	250
hip-hop	0.96	0.75	0.84	250
jazz	0.65	0.82	0.73	250
accuracy			0.80	750
macro avg	0.82	0.80	0.80	750
weighted avg	0.82	0.80	0.80	750

```
Average F1-score
0.6678000000000001
Standard Deviation F1-score
0.029071635660898067
```

Imagem 11: Precisão no melhor caso, média de precisão e variância com atributos relevantes

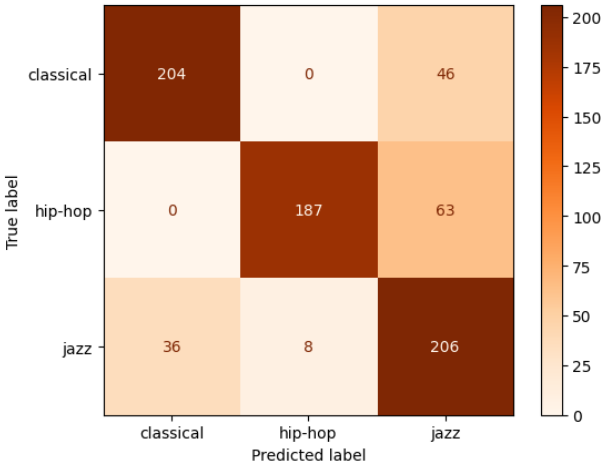

Imagem 12: Matriz de confusão do melhor caso com atributos relevantes

Dos resultados obtidos, viu-se que filtrar apenas os atributos relevantes de fato não foi muito significativo para o desempenho no modelo, tendo uma queda para 66,78% de precisão. Como dito anteriormente, tal resultado já era esperado devido à alta correlação entre os dados e o fato de que o *Naive Bayes* tem como princípio lidar com dados independentes.

Assim, realizou-se um teste utilizando apenas atributos não correlacionados, ou com valores muito baixos de correlação. Para isso, foram retirados os atributos *energy, acousticness* e *instrumentalness*, os quais são fortemente relacionados com *danceability* e *loudness*, fato que foi comentado na análise da imagem 10. Como tratam-se de atributos não correlacionados, e o modelo foi feito para realizar classificação através de atributos independentes, espera-se uma melhora significativa no desempenho do algoritmo. Contudo, um ponto importante é que esses atributos não são 100% correlacionados, apresentando um

nível de independência que será perdido ao serem retirados da base, então é possível que isso atrapalhe no desempenho do algoritmo.

	precision	recall	f1-score	support
classical	0.88	0.76	0.81	250
hip-hop	0.78	0.86	0.82	250
jazz	0.70	0.72	0.71	250
accuracy			0.78	750
macro avg	0.79	0.78	0.78	750
weighted avg	0.79	0.78	0.78	750

Average F1-score
0.7294000000000003
Standard Deviation F1-score
0.01999099797408825

Imagem 13: Precisão do melhor caso, média das precisões e variância com atributos não correlacionados

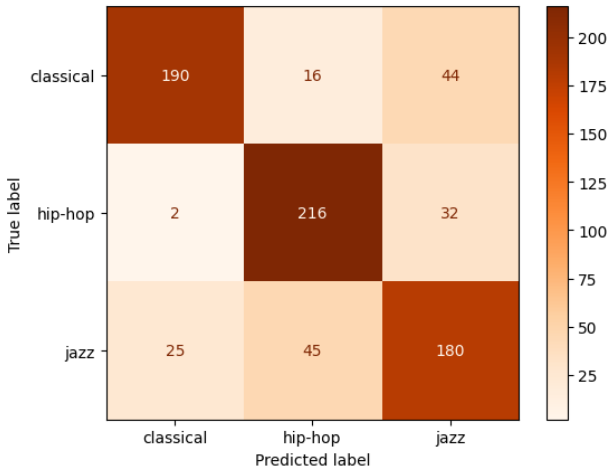

Imagem 14: Matriz de confusão do melhor caso

Como pode ser visto na imagem, o modelo não obteve uma melhora tão significativa quanto o esperado, de forma que seu desempenho seja abaixo dos 73%, que foi a precisão média com todos os atributos, o que mostra que a perda de informação ao retirar os atributos de fato atrapalhou no desempenho do modelo.

Como esse resultado obtido ainda não foi satisfatório, utilizou-se a técnica de PCA para combinar as informações de diferentes atributos em um único, ou seja, é uma possível técnica para solucionar o problema da dimensionalidade. Ela combinou informações de atributos altamente relacionados (*loudness, energy, acousticness, instrumentalness, danceability, valence*) em um único atributo, de forma a reduzir a relação de dependência entre os atributos na base. Ao utilizar essa técnica, espera-se uma melhora significativa no desempenho do algoritmo.

	precision	recall	f1-score	support
classical	0.91	0.74	0.82	250
hip-hop	0.77	0.82	0.80	250
jazz	0.66	0.74	0.70	250
accuracy			0.77	750
macro avg	0.78	0.77	0.77	750
weighted avg	0.78	0.77	0.77	750

Average F1-score
0.7306
Standard Deviation F1-score
0.020239565212721362

Imagem 15: Precisão do melhor caso, média das precisões e variância com PCA

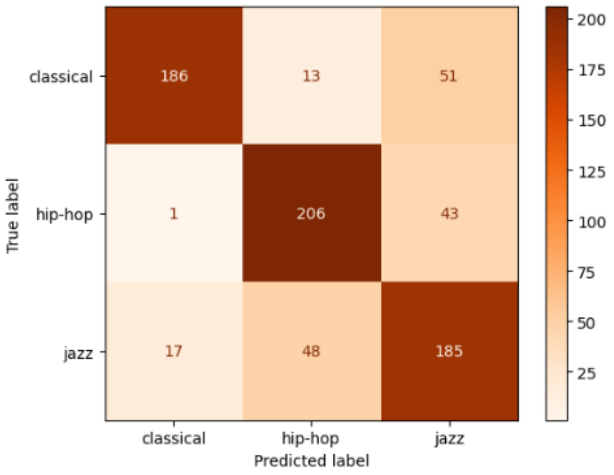

Imagem 16: Matriz de confusão do melhor caso

Novamente as expectativas foram quebradas. Ao comparar com a aplicação utilizando todos os atributos, a utilização do PCA foi praticamente irrelevante no desempenho do modelo, apresentando valores levemente superiores em alguns casos e valores abaixo em outros, como é possível observar na imagem 15.

Algo interessante a se destacar é que em todas as abordagens o gênero Jazz foi que obteve menor eficácia, tendo sempre uma taxa de falsos positivos e falsos negativos muito maior que o das outras classes. A hipótese é de que os atributos da base são muito bons para diferenciar o Clássico do Hip-Hop, dado os atributos *explicity, loudness, acoustcness* por exemplo e Jazz sempre possui ou uma sobreposição maior com um dos gêneros ou com o outro, dificultando o modelo.

V.II. APLICAÇÃO ÁRVORE DE DECISÃO

Posterior à análise de dados, e verificação de os dados apresentam uma sobreposição relativamente grande, já que apresentam objetos de classes diferentes que podem compartilhar o mesmo hiperplano, a outra base de dados a ser utilizada foi a "Árvore de Decisão". A princípio, acreditou-se que seu desempenho não seria muito satisfatório justamente por conta das sobreposição dos hiperplanos, de forma que uma boa classificação necessitaria de hiperplanos específicos, o que geraria um *overfitting* no uso do modelo.

Usando 40% para teste: - F1-score médio: 0,892 - Desvio padrão: 0,009
Usando 35% para teste: - F1-score médio: 0,895 - Desvio padrão: 0,010
Usando 30% para teste: - F1-score médio: 0,898 - Desvio padrão: 0,009
Usando 25% para teste: - F1-score médio: 0,899 - Desvio padrão: 0,011
Usando 20% para teste: - F1-score médio: 0,902 - Desvio padrão: 0,012

Ao contrário do que era esperado inicialmente, a árvore de decisões apresentou um resultado muito melhor do que o previsto, tendo um desempenho até melhor do que o modelo utilizado anteriormente. Após refletir pelo ocorrido pode-se perceber que isso ocorreu pois mesmo com atributos com bastante sobreposição existem muitos outros que separam bem as classes como *loudness* e *explicity,* e, como a árvore de decisão aplica a seleção de atributos, sua eficácia foi muito maior.

O primeiro teste do algoritmo foi feito sem limitação de altura da árvore de decisão, mas sabe-se que quanto maior a altura maior o overfitting do modelo, dessa forma testou-se o algoritmo iterativamente, aumentando a altura máxima da árvore em cada uma, analisando seu f1-score médio e parando quando ele decrescesse.

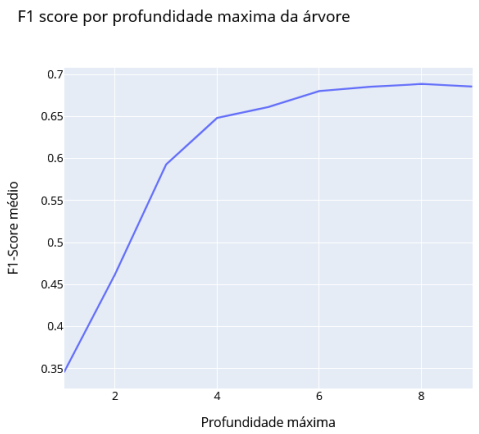

Imagem 17: Gráfico do crescimento do f1-score ao longo das iterações

Com esse teste pode-se encontrar que a altura máxima que maximiza o f1-score médio é 9, assim conseguiu-se subir consideravelmente a eficácia do algoritmo devido a maior generalização.

	precision	recall	f1-score	support
classical	0.92	0.94	0.93	250
hip-hop	0.97	0.96	0.96	250
jazz	0.91	0.90	0.91	250
accuracy			0.93	750
macro avg	0.93	0.93	0.93	750
weighted avg	0.93	0.93	0.93	750

Imagem 18: Precisão do melhor caso

```
Tree com maxdeaph 9
Average F1-score
0.9052999999999999
Standard Deviation F1-score
0.010242558274181317
```

Imagem 19: , média das precisões e variância com atributos não correlacionados

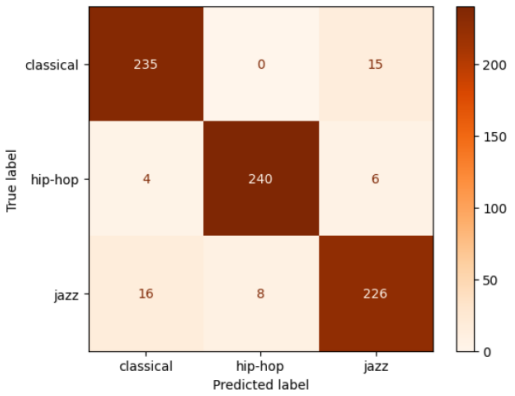

Imagem 20: Matriz de confusão do melhor caso

VI. APLICAÇÃO DOS MODELOS DE APRENDIZADO DE MÁQUINA NA BASE SIM_SPF

Relembrando a hipótese gerada no tópico III.I, supôs-se que com caso seja utilizado gêneros semelhantes, os modelos de classificação apresentariam um desempenho insatisfatório e bem abaixo de quando é utilizado gêneros diferentes, como o caso da base *dif_spf*. No caso, realizamos uma implementação do *Naive Bayes* utilizando essa base de dados, com os todos os atributos juntos

```
40%
0.4542
0.02627470266244701

35%
0.4488
0.026543549122150192

30%
0.4483
0.022937087870956944

25%
0.44800000000000006
0.024372115213907884

20%
0.4411
0.022973680593235384
```

Imagem 17: Porcentagem utilizada para teste, precisão média e variância

A partir das imagens, é possível ver que o primeiro algoritmo teve o comportamento esperado, ou seja, apresentou um resultado pior quando comparado com a utilização da base *dif_spf*.

Após isso, também realizamos uma análise utilizando a técnica de PCA para ver qual seria a melhora no caso.

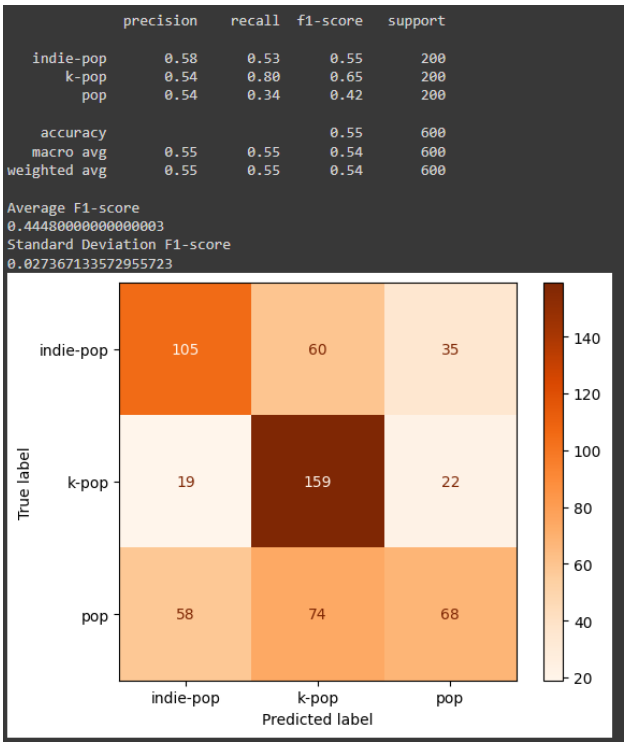

Imagem x: Precisão no melhor caso, média das precisões, variância e matriz de confusão do melhor caso

A utilização do PCA não foi tão significativa positivamente, mantendo um resultado bem baixo. Interessante ressaltar que a maior parte dos falsos positivos e falsos negativos ocorreram com o gênero Pop, do qual os outros dois derivam. Isso realmente demonstra que o Indie-Pop e K-Pop são variações do Pop, e ainda apresentam características dele.

Posteriormente, também foi testada a hipótese com o algoritmo de Árvore de Decisões. Neste caso, também é esperado um desempenho abaixo em relação a utilização da outra base. Como há maior sobreposição dos dados, também é esperado que os hiperplanos tenham um menor desempenho na classificação, e que caso a profundidade seja muito grande o *overfitting* crescerá consideravelmente.

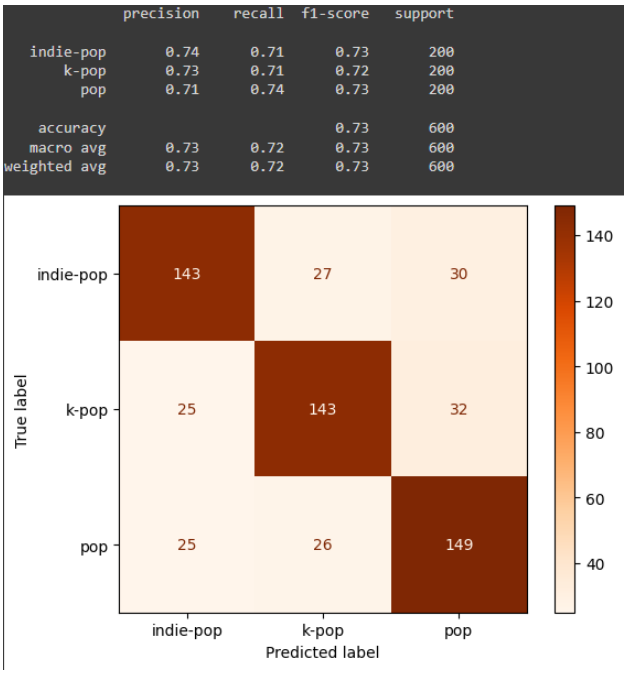

Imagem x: Precisão do melhor caso, média das precisões, variância e matriz de confusão do melhor caso

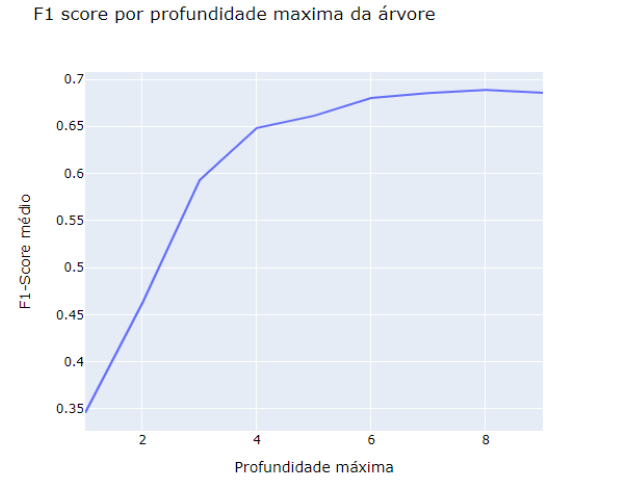

Imagem x: Gráfico *profundidade máxima* x *precisão*

É possível verificar que o último algoritmo também obteve um desempenho próximo do esperado, apesar de não ter tido uma piora tão significativa quanto a do *Naive Bayes*. Além disso, diferentemente do *Naive Bayes*, os falsos positivos e os falsos negativos foram mais distribuídos, de forma que o fato de o Indie-Pop e K-Pop serem do gênero Pop não seja tão relevante assim na classificação. Outra coisa que é possível visualizar na figura x, é que a partir do oitavo nível de profundidade já começa a ocorrer o *overfitting*, deixando o algoritmo mais específico e menos genérico, piorando assim o seu desempenho.

VI. CONCLUSÃO

Por fim, a partir dos resultados obtidos, é possível visualizar que de fato, a hipótese gerada no tópico III.I é verdadeira, ou seja, gêneros semelhantes são consideravelmente mais difíceis de serem classificados, enquanto gêneros diferentes são mais facilmente classificados.

Além disso, ao contrário do que era esperado, a Árvore de Decisões, em ambos os subconjuntos de dados, apresentou um melhor desempenho em comparação ao *Naive Bayes*, sendo, inclusive, menos afetada pela semelhança dos gêneros do que o primeiro algoritmo foi.

Finalmente, também foi possível concluir que os métodos de pré-processamento utilizados não tiveram o efeito esperado, ou então um efeito tão significativo. Contudo, foi possível confirmar que a seleção de atributos significativos foi teria um efeito negativo no desempenho da análise, indo de encontro com o resultado esperado.

VII. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Link do DataSet

[2] https://scikit-learn.org

[3] https://www.mathworks.com

[4] https://stackoverflow.com

[5] Ferreira, A. C. P. L. (2011). Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina. LTC.