Introdução e Contextualização

Em um mundo cada vez mais moldado por grandes volumes de informação digital, a capacidade de organizar, compreender e extrair padrões de dados não estruturados torna-se essencial. Entre esses dados, os arquivos de áudio ocupam uma posição especial por sua riqueza cultural, diversidade de formatos e desafios técnicos específicos. No contexto musical, a identificação de covers — ou seja, versões alternativas de músicas já existentes — representa uma tarefa interessante tanto do ponto de vista artístico quanto computacional. Covers podem variar drasticamente em estilo, ritmo, voz e instrumentação, o que torna sua detecção automatizada um problema real de análise de dados, bastante desafiador e repleto de nuances.

Este projeto surgiu como uma proposta prática dentro da disciplina de Computação Científica e Análise de Dadods, cursada no âmbito da graduação em Ciência da Computação. O objetivo principal foi investigar métodos capazes de diferenciar músicas originais de suas versões cover, utilizando estratégias de organização de dados, extração de características relevantes e técnicas matemáticas para redução e análise dimensional. A ideia central era simples, mas ambiciosa: seria possível agrupar automaticamente músicas similares, mesmo que tenham arranjos distintos, apenas a partir de suas propriedades sonoras?

A motivação foi além de uma simples tarefa de agrupamento. A partir da observação de bancos de dados como o Covers80 Dataset, surgiu a curiosidade de entender como características computacionais — como espectros, curvas melódicas ou outras representações numéricas — poderiam ser manipuladas para evidenciar similaridades que o ouvido humano percebe de forma intuitiva, mas que um algoritmo precisa aprender a reconhecer. Em outras palavras, como traduzir algo tão subjetivo como "soar parecido" em um critério matemático e cional válido?

O projeto também teve o intuito de reforçar, na prática, os conteúdos vistos em sala de aula sobre pré-processamento de dados, construção de vetores de documentos (ou, neste caso, vetores de áudio), redução de dimensionalidade e interpretação de agrupamentos via visualização. Isso nos levou ao uso de técnicas clássicas de aprendizado não supervisionado, como o PCA (Principal Component Analysis), que desempenhou um papel fundamental na representação visual dos dados e na identificação dos grupos de músicas similares.

Por fim, vale destacar que este relatório foi escrito com o intuito de refletir o processo real de desenvolvimento do projeto: as decisões certas e os erros cometidos, as

dificuldades encontradas ao lidar com dados de áudio em larga escala e, principalmente, a tentativa constante de conectar teoria e prática. Mais do que buscar a solução perfeita, o projeto serviu como laboratório para a aplicação de conceitos estudados e para a experimentação livre — algo raro, porém valioso, no ambiente acadêmico.

Organização dos Dados e Pré-Processamento

Após ter estruturado as primeiras ideias do projeto, percebi rapidamente que a organização dos dados seria uma etapa essencial para que tudo funcionasse de forma estável e coerente. Embora parecesse simples criar uma pasta com músicas, nomeá-las e começar a análise, a realidade foi bem mais trabalhosa. A primeira coisa que fiz foi separar as músicas em duas categorias: originais e covers. Coloquei esses arquivos dentro de uma pasta chamada data/, garantindo que cada par de original e cover tivesse exatamente o mesmo nome — uma estratégia simples, mas que me ajudou bastante a parear as músicas posteriormente.

Como os arquivos estavam em formato .mp3, precisei me certificar de que a biblioteca que usaria fosse compatível. Escolhi a librosa para extrair as características dos áudios, mas logo descobri que ela não lida diretamente com .mp3 sem o suporte do ffmpeg. Tive que instalar essa dependência e ajustar o ambiente, o que me fez perder algum tempo até conseguir configurar tudo corretamente. Esses problemas de infraestrutura me ensinaram que, muitas vezes, os maiores atrasos de um projeto não estão nos cálculos matemáticos, mas nas ferramentas que precisamos fazer funcionar primeiro.

Depois de superar essa etapa, comecei o processo de leitura e extração dos vetores MFCC (Mel Frequency Cepstral Coefficients), que serviriam como a representação matemática dos áudios. Cada vetor precisava ter uma dimensão semelhante, então decidi aplicar uma interpolação nos vetores menores para que pudessem ser comparados com os demais. Não cortei ou padronizei a duração dos áudios, porque queria preservar o máximo possível da integridade do material sonoro. Preferi lidar com vetores mais longos e normalizados do que perder trechos relevantes.

Durante os testes, percebi que o tempo de leitura dos arquivos estava se tornando um gargalo. Cada vez que eu rodava o programa, ele demorava vários minutos só para extrair novamente os vetores das músicas — mesmo quando os arquivos não tinham mudado. Para resolver isso, implementei um sistema de cache. Gravei os vetores MFCC extraídos em arquivos .pkl e, em execuções seguintes, apenas carregava esses dados prontos. Isso reduziu drasticamente o tempo de espera e tornou a experiência

de uso muito mais leve. Também incluí uma função para limpar esse cache, caso eu precisasse forçar uma nova leitura, especialmente útil durante os testes e ajustes finos.

Além disso, criei uma estrutura de classes para representar as músicas no código. Cada música possuía atributos como nome, se era original ou cover, e qual vetor de características ela carregava. Isso me ajudou a manter o código mais modular e organizado, facilitando também o tratamento de grupos e comunidades mais adiante. Ao fim dessa etapa, senti que tinha em mãos um conjunto de dados limpo, padronizado e otimizado, pronto para ser analisado com as ferramentas matemáticas que eu gostaria de aplicar.

Essa fase me mostrou, com clareza, que grande parte do sucesso de uma análise não vem apenas da matemática usada, mas da preparação dos dados. Um bom pré-processamento poupa trabalho e garante que as interpretações posteriores façam sentido. Foi cansativo e exigiu paciência, mas foi uma das etapas mais importantes do projeto.

Estratégias Matemáticas

Com os dados organizados e os vetores MFCC já extraídos, percebi que precisava de alguma técnica matemática que me permitisse visualizar e entender a relação entre as músicas. Sabia que os vetores extraídos eram de alta dimensão e, portanto, difíceis de interpretar visualmente ou mesmo de comparar diretamente. Foi nesse ponto que decidi aplicar o PCA (Análise de Componentes Principais) como estratégia central de análise dimensional.

Antes de tudo, revisei os conceitos por trás do PCA. Eu sabia que essa técnica serve para projetar dados de alta dimensão em um espaço de menor dimensão, preservando ao máximo a variância dos dados. Entendi que, ao reduzir a dimensão, conseguiria representar os vetores MFCC em duas ou três dimensões, facilitando a visualização das semelhanças entre músicas. Cada música, que antes era uma matriz complicada de coeficientes, passaria a ser um ponto no espaço — e, com isso, eu poderia tentar identificar grupos, padrões e relações.

Implementei a redução com a ajuda da biblioteca sklearn, utilizando o método PCA(n_components=2). Escolhi apenas duas dimensões para que eu conseguisse gerar um gráfico bidimensional simples, que poderia ser visualizado com o matplotlib. A primeira vez que vi os pontos se agrupando no gráfico, percebi que o PCA tinha funcionado: músicas que eram covers de uma mesma original realmente estavam próximas entre si no plano. Isso me deu confiança para seguir explorando.

Ao observar a distribuição dos pontos, percebi que surgiam pequenas “nuvens” no espaço, indicando uma proximidade entre músicas. Interpretei esses grupos como possíveis comunidades de covers, ou seja, conjuntos de músicas que pareciam cobrir a mesma original ou que compartilhavam características sonoras próximas. Apesar de o PCA não usar nenhuma informação semântica, apenas matemática, ele foi capaz de refletir esses agrupamentos naturais de maneira surpreendentemente eficaz.

Para melhorar ainda mais a leitura do gráfico, associei cores diferentes a cada comunidade detectada. Isso me ajudou a visualizar como os covers se distribuíam em relação aos originais. O projeto não incluía uma técnica automática de clusterização, mas a própria redução dimensional já sugeria agrupamentos visíveis, o que me levou a pensar em futuras melhorias — como integrar algoritmos de agrupamento, como DBSCAN ou KMeans, para reforçar essa ideia de comunidades musicais.

Ao longo dessa etapa, aprendi que a matemática pura, mesmo que sem contexto sonoro direto, pode revelar relações escondidas dentro de um conjunto de dados complexos. A aplicação do PCA foi, sem dúvidas, o ponto em que o projeto deixou de ser apenas uma leitura de arquivos e passou a se tornar uma ferramenta real de análise. Senti que, finalmente, estava dando forma visual a uma ideia abstrata — e que a matemática estava ali, agindo como ponte entre som e interpretação.

Resultados Obtidos e Interpretação Visual

Depois de aplicar o PCA e representar visualmente os dados, pude finalmente observar os primeiros resultados concretos do projeto. Olhar para aquele gráfico foi como ver uma versão física das conexões musicais que antes só existiam na minha cabeça. Pela primeira vez, tive a oportunidade de enxergar, de forma objetiva, o quanto as músicas covers estavam realmente próximas das originais em termos de suas características matemáticas. Essa visualização se tornou o ponto de virada da minha compreensão:

agora, eu não dependia apenas da minha intuição ou da análise manual das letras ou melodias — os dados estavam falando por si.

Notei que várias músicas que eu já sabia serem covers estavam agrupadas em pequenas regiões do plano. Essas músicas compartilhavam padrões semelhantes nos coeficientes MFCC e, por isso, acabaram próximas após a redução dimensional. Isso confirmou que o PCA estava funcionando como esperado, preservando a essência sonora e permitindo uma comparação direta. Em alguns casos, covers de uma mesma música apareceram quase sobrepostos no gráfico. A partir disso, comecei a identificar essas regiões como potenciais comunidades musicais, conceito que adotei inspirado em trabalhos acadêmicos da área.

O mais interessante foi perceber que nem todos os agrupamentos eram óbvios. Algumas músicas covers que tinham estilos bastante diferentes — como versões acústicas ou remixadas — ainda assim apareciam próximas das suas versões originais. Isso me surpreendeu. Eu esperava que essas variações estilísticas distorcessem os vetores MFCC de tal forma que afastassem as músicas no espaço vetorial. No entanto, ao contrário, percebi que muitos elementos essenciais da música original se mantinham, e o PCA conseguia capturar essa similaridade mesmo quando a instrumentação era totalmente diferente. Isso me levou a refletir sobre a profundidade dos coeficientes MFCC e sobre como eles vão além da melodia superficial.

Além disso, encontrei casos em que músicas não relacionadas acabaram ficando próximas no gráfico. Em alguns desses casos, percebi que isso se devia a limitações do meu processo de extração: por exemplo, faixas com introduções longas ou com qualidade de gravação inferior acabavam gerando vetores pouco representativos. Isso me ensinou a importância do pré-processamento e da padronização dos dados, etapas que, por enquanto, optei por não aprofundar por limitações de tempo e conhecimento técnico. Reconheço, no entanto, que seriam essenciais em versões futuras do projeto.

Utilizei gráficos com cores e legendas que ajudassem a separar visualmente os grupos identificados. Para cada música, eu exibi o nome no gráfico, o que tornou o processo de interpretação mais acessível e direto. Com isso, consegui criar uma interface visual que me permitia navegar pelas relações musicais com facilidade. Essa visualização não só facilitou a validação dos resultados como também me motivou a pensar em outras formas de representação — talvez um grafo interativo, em que cada música fosse um nó conectado por arestas de similaridade, ou mesmo um sistema web com busca por similaridade vetorial.

Em resumo, os resultados visuais me mostraram que o caminho que escolhi estava dando frutos. Com uma metodologia simples, mas bem pensada, fui capaz de representar e interpretar padrões escondidos nos dados. Vi com clareza que a combinação de técnicas matemáticas com representação gráfica pode revelar conexões que vão além do óbvio. E, mais importante, senti que criei algo que não apenas processava dados, mas contava uma história — a história de como músicas se relacionam entre si, mesmo quando cantadas por vozes diferentes.

Conclusão e Possíveis Melhorias

Ao concluir este projeto, percebi que aprendi mais do que apenas aplicar técnicas de análise de dados ou escrever código funcional. Vivi, na prática, as dificuldades e conquistas que acompanham qualquer processo investigativo em Ciência da ção. Desde o momento em que escolhi trabalhar com músicas covers, sabia que estava entrando em um campo mais sensível, onde as informações não são sempre estruturadas e onde a subjetividade artística desafia a matemática. Mas mesmo assim, escolhi seguir, e acredito que tomei a decisão certa.

Senti que o uso do PCA foi essencial para transformar dados abstratos em algo visualmente interpretável. Com ele, consegui representar a semelhança sonora entre músicas de forma clara, intuitiva e até elegante. A redução dimensional, que no início era apenas uma solução matemática para lidar com vetores grandes, acabou se tornando a chave para criar uma linguagem comum entre som e dados. Isso me mostrou o poder das ferramentas estatísticas e como elas podem ser utilizadas para traduzir fenômenos complexos em padrões compreensíveis.

Apesar disso, reconheço que o projeto ainda tem limitações importantes. Não apliquei nenhuma técnica de normalização na entrada dos áudios, como padronizar o volume, cortar introduções ou uniformizar a duração das faixas. Esses fatores impactaram diretamente a qualidade da extração de características. Além disso, percebi que alguns resultados no gráfico de clusters poderiam estar enviesados por ruídos de gravação ou efeitos especiais presentes em certas versões cover. Esses ruídos, embora comuns na música, afetam negativamente a precisão dos vetores MFCC, e isso é algo que eu gostaria de corrigir em versões futuras.

Outra melhoria que considerei foi a adoção de métodos de clusterização mais robustos. No momento, usei o KMeans apenas como ferramenta de agrupamento visual, mas sei que existem técnicas mais sofisticadas, como DBSCAN ou HDBSCAN, que talvez se adequassem melhor ao tipo de distribuição que observei. Também gostaria de explorar o uso de autoencoders ou UMAP no lugar do PCA, principalmente para ver se conseguiria capturar estruturas mais profundas e não lineares entre os dados.

Além disso, senti falta de um sistema que permitisse classificar novas músicas automaticamente. Embora tenha criado uma função para sugerir músicas parecidas, ela ainda é muito simples. Em uma nova versão, planejo treinar um modelo supervisionado com exemplos reais de covers e originais, a fim de automatizar a detecção com mais precisão e confiança. Também penso em criar uma interface gráfica mais sofisticada ou mesmo uma aplicação web interativa onde qualquer pessoa possa fazer upload de uma música e visualizar sua posição no mapa de similaridade.

Por fim, compreendi que a beleza desse projeto está justamente em sua abertura para expansão. Comecei com uma ideia simples: descobrir se um cover soa parecido com sua música original. Mas ao longo do caminho, vi que isso se desdobrou em muitos aprendizados — de programação, de estatística, de engenharia de dados e, acima de tudo, de pensamento estruturado. Não criei uma solução perfeita, mas criei algo funcional, com propósito e que reflete o que aprendi até aqui.

Acredito que esse seja o tipo de projeto que nunca termina de verdade. Sempre haverá uma nova técnica para testar, uma nova base de dados para importar, ou uma nova pergunta para responder. E isso me anima. Porque mais do que entregar um trabalho, eu construí uma ferramenta que pode evoluir junto comigo.

Observações

Mais informações sobre o projeto estão no arquivo README.md, onde mostro as screenshots do projeto e detalho mais sobre a aplicação prática.