Rodrigo Santos Cordeiro

SUMARIZAÇÃO AUTOMÁTICA DE LETRAS MUSICAIS: IDENTIFICAÇÃO DE TEMAS E EMOÇÕES COM TEXT MINING

Projeto no âmbito do Mestrado em Ciência de Dados

Orientado pelo Professor Doutor Ricardo Malheiro do Instituto Politécnico de Leiria

Setembro de 2025

Rodrigo Santos Cordeiro

SUMARIZAÇÃO AUTOMÁTICA DE LETRAS MUSICAIS: IDENTIFICAÇÃO DE TEMAS E EMOÇÕES COM TEXT MINING

Projeto no âmbito do Mestrado em Ciência de Dados

Orientado pelo Professor Doutor Ricardo Malheiro do Instituto Politécnico de Leiria

Setembro de 2025

“Tudo o que REALMENTE queremos acontece”

Rodrigo Cordeiro

Resumo

A identificação de emoções em música é um dos principais desafios da área de *Music Emotion Recognition* (MER), um subcampo do *Music Information Retrieval* (MIR). No caso particular das letras musicais, estas desempenham um papel determinante na forma como as emoções são transmitidas e percecionadas pelos ouvintes. A representação emocional adotada neste trabalho baseia-se no modelo dimensional de Russell, que organiza as emoções segundo as dimensões de valência e ativação, permitindo posicionar cada letra num dos quatro quadrantes emocionais.

A motivação central deste estudo surge da necessidade de reduzir a complexidade textual das letras musicais sem comprometer a informação emocional nelas contida. A aplicação de técnicas de sumarização automática, extrativas e abstrativas, constitui uma estratégia promissora para diminuir redundâncias e simplificar o texto, possibilitando ao mesmo tempo ganhos de eficiência computacional e, potencialmente, melhorias na generalização dos modelos de classificação.

Para investigar esta hipótese, foram utilizadas 180 letras musicais anotadas segundo o modelo de Russell. Implementaram-se técnicas de sumarização extrativa (como TextRank e z.BERT) e abstrativa (como PEGASUS, LED e GPT-2), seguidas da extração de *features* lexicais, estilísticas e semânticas. Os textos originais e sumarizados foram depois utilizados no treino de diferentes classificadores, abrangendo tanto modelos clássicos de machine learning (SVM, *Random Forest*) como abordagens de *deep learning* (MLP, H2O MLP, BERT e GPT-2). O desempenho foi avaliado com base em métricas como *accuracy*, *precision*, F1-*score* e AUC.

Os resultados mostraram que, embora as letras completas continuem a assegurar o melhor desempenho, algumas técnicas de sumarização, em particular a extrativa com BERT e TextRank, produziram resultados muito próximos, demonstrando que a redução textual não compromete de forma significativa a classificação emocional. Entre os modelos testados, o *Multilayer Perceptron* (MLP) destacou-se como o mais robusto, confirmando a superioridade das abordagens de *deep learning* face aos métodos clássicos.

Este trabalho contribui para a área de MER ao demonstrar a viabilidade da integração entre sumarização automática e classificação de emoções musicais. Os resultados obtidos abrem perspetivas para aplicações práticas em sistemas de recomendação personalizados, análise de tendências culturais e apoio a processos criativos, além de fornecerem bases metodológicas para futuras investigações em contextos multimodais que combinem letra e áudio.

Abstract

The identification of emotions in music is one of the main challenges in the field of Music Emotion Recognition (MER), a subfield of Music Information Retrieval (MIR). In the particular case of song lyrics, they play a decisive role in how emotions are conveyed and perceived by listeners. The emotional representation adopted in this work is based on Russell's dimensional model, which organizes emotions along the dimensions of valence and arousal, allowing each lyric to be positioned within one of the four emotional quadrants.

The central motivation of this study stems from the need to reduce the textual complexity of song lyrics without compromising the emotional information they contain. The application of automatic summarization techniques, both extractive and abstractive, represents a promising strategy to reduce redundancy and simplify the text, while at the same time enabling computational efficiency gains and potentially improving the generalization capacity of classification models.

To investigate this hypothesis, 180 song lyrics annotated according to Russell's model were used. Extractive summarization techniques (such as TextRank and z.BERT) and abstractive approaches (such as PEGASUS, LED, and GPT-2) were implemented, followed by the extraction of lexical, stylistic, and semantic features. Both the original and summarized texts were then used to train different classifiers, covering classical machine learning models (SVM, Random Forest) as well as deep learning approaches (MLP, H2O MLP, BERT, and GPT-2). Performance was evaluated using metrics such as accuracy, precision, F1-score, and AUC.

The results showed that, although complete lyrics still ensure the best overall performance, some summarization techniques—particularly extractive methods with z.BERT and TextRank—produced results that were very close, demonstrating that textual reduction does not significantly compromise emotional classification. Among the tested models, the Multilayer Perceptron (MLP) stood out as the most robust, confirming the superiority of deep learning approaches over classical methods.

This work contributes to the field of MER by demonstrating the feasibility of integrating automatic summarization with emotion classification in music lyrics. The results open perspectives for practical applications in personalized recommendation systems, cultural trend analysis, and creative support processes, while also providing methodological foundations for future research in multimodal contexts that combine lyrics and audio.

Agradecimentos

Ao longo da vida, cada etapa traz os seus desafios e aprendizagens, tornando o caminho, por vezes, mais complexo do que seria esperado. No entanto, também é verdade que nenhum percurso se faz sozinho, e que muitas vezes é o impulso certo, dado no momento certo, que nos permite avançar. Reconhecendo isso, e sabendo que seria injusto destacar apenas alguns nomes quando tantos contribuíram de diferentes formas, deixo aqui um agradecimento genuíno e abrangente a todos os que, direta ou indiretamente, fizeram parte desta caminhada.

Grato.

Índice

Índice de Figuras

Índice de Tabelas

1. INTRODUÇÃO

A música é uma das formas mais universais de expressão artística, desempenhando um papel central na comunicação de sentimentos e estados emocionais. O Reconhecimento de Emoções em Música (*Music Emotion Recognition* – MER) é uma área interdisciplinar que combina elementos de processamento de linguagem natural, *machine learning* e psicologia para identificar emoções transmitidas em músicas. No mundo atual, onde plataformas de *streaming* dominam o consumo musical, a identificação de emoções tornou-se essencial para personalizar recomendações e melhorar a experiência do utilizador. Além do impacto na personalização, o MER tem aplicações em terapia musical, marketing e até no desenvolvimento de sistemas de inteligência artificial com maior capacidade de interação emocional (Kim et al., 2010).

Embora grande parte dos trabalhos em MER se concentre no sinal áudio, as letras das músicas desempenham um papel crucial, dado que transportam significados semânticos e simbólicos que intensificam ou até reconfiguram a experiência emocional do ouvinte (Hu et al., 2010). No entanto, as letras apresentam desafios adicionais, como a diversidade estilística, o uso frequente de metáforas e a elevada dimensionalidade textual. Para lidar com estes aspetos, a sumarização automática de texto tem surgido como uma técnica promissora, permitindo reduzir a complexidade textual enquanto se preserva a informação essencial (Nenkova & McKeown, 2012)

1.1 Problema de Investigação

Apesar dos avanços no Music Emotion Recognition (MER), a investigação tem-se focado predominantemente no sinal áudio, sendo a utilização das letras musicais e o impacto da sua transformação ainda pouco explorados. As letras contêm informação semântica e simbólica que complementa o áudio e podem alterar significativamente a perceção emocional de uma música. No entanto, a sua elevada dimensionalidade e uso frequente de metáforas dificultam a modelação automática das emoções.

A sumarização automática surge como uma abordagem promissora para reduzir a complexidade textual, mas o seu efeito na qualidade da classificação emocional ainda não está bem documentado. Este estudo pretende colmatar essa lacuna, respondendo à seguinte questão central de investigação:

- Será que a utilização de letras sumarizadas (por métodos extrativos e abstrativos) permite treinar modelos de classificação de emoções que mantenham ou até melhorem o

desempenho obtido com letras completas, sem comprometer a integridade da informação emocional?

Adicionalmente, pretende-se compreender de que forma diferentes métodos de sumarização e classificação influenciam o desempenho de algoritmos de *machine learning* e *deep learning*, fornecendo evidências experimentais para orientar futuras investigações e aplicações em sistemas de recomendação musical.

1.2 Objetivos

O objetivo principal deste trabalho é investigar de forma sistemática o impacto da sumarização automática de letras musicais, utilizando abordagens extrativas e abstrativas, na tarefa de reconhecimento automático de emoções.

Pretende-se compreender se a redução da complexidade textual, obtida através da sumarização, pode melhorar a eficiência computacional e a capacidade de generalização dos modelos de classificação, sem comprometer a integridade da informação emocional transmitida nas letras.

Além disso, este estudo procura:

- Avaliar a robustez de diferentes técnicas de sumarização (extrativas e abstrativas) no contexto de MER, identificando quais melhor preservam os sinais emocionais relevantes.
- Explorar de que forma a seleção de features influencia o desempenho dos modelos de classificação quando aplicada a letras originais e resumidas.
- Analisar os ganhos e perdas no desempenho de modelos de *machine learning* e *deep learning* (incluindo *LLMs* como *BERT* e *GPT-2*) quando treinados com letras sumarizadas.

Para atingir o objetivo principal, definem-se os seguintes objetivos secundários, organizados em etapas que refletem o fluxo lógico da investigação:

- Aplicação de Técnicas de Sumarização
  - Implementar algoritmos de sumarização extrativa e abstrativa.
  - Comparar o desempenho das abordagens
- Extração e Seleção de *Features*
  - Extrair atributos das letras originais e resumidas.

- Treino e Otimização de Modelos de Classificação
    - Implementar e treinar modelos clássicos de *machine learning* e modelos baseados em *deep learning* e LLMs.
- Comparação e Avaliação de Desempenho
    - Avaliar os modelos através de várias métricas.
    - Identificar quais combinações de técnicas de sumarização e classificação geram os melhores resultados.
- Análise Crítica e Recomendações
    - Discutir os resultados obtidos, identificando as vantagens e limitações do uso de letras sumarizadas no MER.
    - Propor recomendações para investigação futura e possíveis aplicações práticas, como sistemas de recomendação baseados em emoção.

1.3 Metodologia Resumida

A metodologia seguida neste trabalho foi desenvolvida em várias etapas, procurando responder à questão central da investigação: avaliar o impacto da sumarização de letras musicais no reconhecimento automático de emoções. O processo incluiu a aplicação de técnicas de sumarização, a extração e seleção de features, e a implementação de diferentes modelos de classificação.

Durante a execução do projeto, alguns constrangimentos influenciaram o planeamento inicial. O tratamento e adaptação das sumarizações, especialmente as abstrativas, exigiram mais tempo do que o previsto, uma vez que os modelos testados produziram frequentemente resultados incoerentes ou redundantes, obrigando à aplicação de pós-processamento adicional. Esta situação implicou uma revisão do cronograma, prolongando as fases de limpeza e preparação dos dados.

Outro desafio esteve relacionado com a disponibilidade de recursos computacionais. A utilização de modelos de linguagem de larga escala, como BERT e GPT-2, exigiu uma capacidade de processamento significativa, que nem sempre estava disponível no ambiente de execução. Para contornar estas limitações, foi necessário ajustar o processo experimental, nomeadamente através da redução do número de parâmetros e do tamanho das épocas de treino, o que permitiu garantir a execução dos modelos e a realização de testes consistentes dentro dos prazos estipulados.

Apesar dos constrangimentos, a metodologia beneficiou desta adaptação, pois permitiu explorar diferentes combinações de sumarização e classificação em cenários reais de limitação

de recursos. A diversidade de métodos aplicados — desde algoritmos clássicos de *Machine Learning* até arquiteturas baseadas em *transformers* — possibilitou uma análise comparativa mais rica e representativa.

Para garantir uma execução organizada e coerente do projeto, foi elaborado um plano de atividades representado no Diagrama de Gantt (Figura 1). Este diagrama apresenta a calendarização de todas as etapas metodológicas, aplicação das técnicas de sumarização, extração e seleção de *features*, treino e otimização dos modelos de classificação, bem como a análise e discussão dos resultados. A utilização do Gantt permitiu acompanhar a evolução do trabalho, ajustar prazos quando necessário e assegurar que as diferentes fases fossem concluídas de forma sequencial e controlada, mesmo perante os constrangimentos identificados.

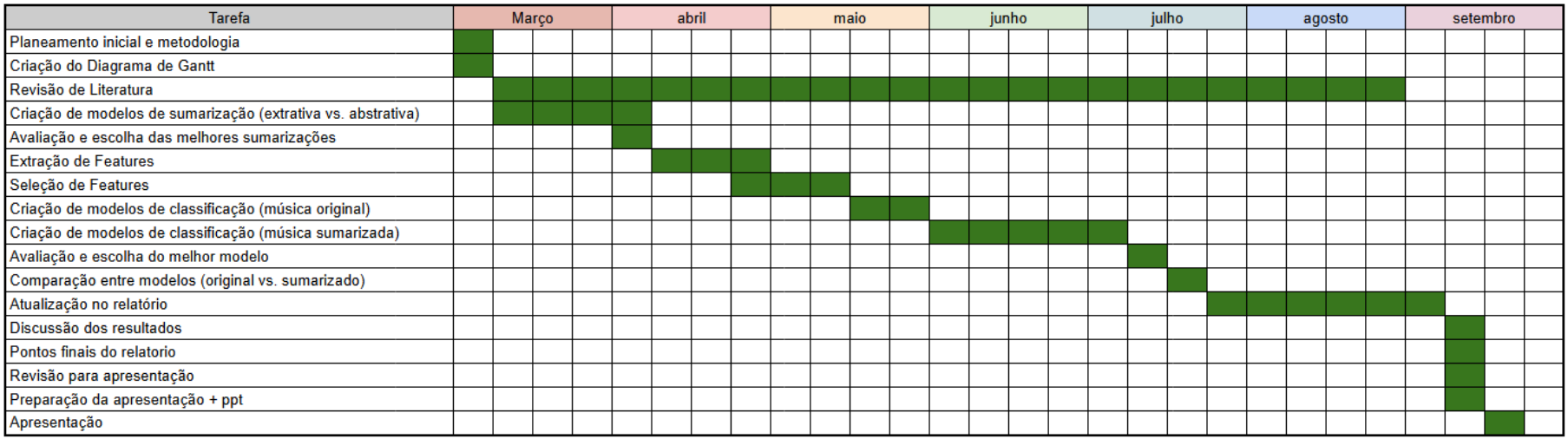

Figura 1- Diagrama de Gantt

1.4 Estrutura do relatório de Projeto

A presente relatório de projeto está organizada da seguinte forma:

No capítulo 1 temos a introdução onde são apresentados o enquadramento do tema, a problemática em estudo, os objetivos, a metodologia e as contribuições do trabalho. De seguida, o capítulo 2 é onde é feita a revisão de literatura para conseguir descrever os conceitos fundamentais de Psicologia das Emoções, abordando definições, modelos emocionais, a distinção entre emoção expressa, percebida e sentida, bem como técnicas de *Music Emotion Recognition* (MER), sumarização automática de texto e métodos de classificação. No capítulo 3 segue a descrição do corpus que caracteriza o conjunto de dados utilizado, detalhando o processo de construção e anotação do *dataset* de letras musicais. A partir do capítulo 4 começa o desenvolvimento prático do projeto onde neste capítulo temos o processo de sumarizações onde explica os métodos de sumarização extrativa e abstrativa aplicados às letras, bem como os desafios encontrados no pré e pós-processamento, de seguida o capítulo 5 que corresponde ao processo de classificação onde apresenta as etapas de extração e seleção de *features*, a implementação dos algoritmos de classificação e a definição das métricas de avaliação. No capítulo 6 encontramos a comparação de modelos que é onde se faz, se analisa e se compara o desempenho dos modelos de classificação aplicados às letras originais e às versões sumarizadas, por fim no capítulo 7 é onde tiramos as conclusões e trabalho futuro que sintetiza os principais resultados obtidos, destacando as contribuições do estudo, e apresenta perspetivas para investigações futuras.

2. REVISÃO DE LITERATURA

O reconhecimento de emoções tem sido amplamente estudado em áreas como psicologia, música e inteligência artificial, refletindo a importância das emoções na experiência musical. No âmbito do *Music Emotion Recognition* (MER), embora grande parte da investigação se concentre no áudio, as letras assumem também um papel essencial, acrescentando significado semântico que pode influenciar a perceção emocional.

Para sustentar esta investigação, importa abordar a definição de emoção, a distinção entre emoção expressa, percebida e sentida e os principais modelos de representação emocional. Seguidamente, destaca-se a relevância da sumarização automática de textos na redução da complexidade das letras, bem como o papel da extração de *features* na sua transformação em representações computacionais. Por fim, são apresentados os métodos de classificação que permitem avaliar o impacto de letras originais e sumarizadas no reconhecimento automático de emoções.

2.1 A Importância das Letras no MER

As letras desempenham um papel essencial na transmissão de emoções, sendo um complemento ao áudio. Frases poéticas, metáforas e palavras-chave frequentemente carregam significados emocionais que definem o tom de uma música. Segundo PN Juslin & P Laukka (2004), 29% das pessoas referem que as letras são um fator importante na forma como a música transmite emoções. Este dado demonstra que, embora o áudio tenha um peso significativo, as letras possuem um papel expressivo e não negligenciável na experiência emocional. Como sublinha Ricardo Malheiro (2016), a integração de ambas as dimensões, áudio e letra, em sistemas bimodais tende a produzir resultados mais robustos e fidedignos no reconhecimento de emoções musicais, quando comparados com abordagens que utilizam apenas uma das modalidades.

Empresas como o Spotify têm investido em modelos de *Music Emotion Recognition* (MER) e análise de conteúdo musical para melhorar a experiência do utilizador e aumentar o tempo de permanência na plataforma. A partir da sua *API* pública, sabe-se que são disponibilizados atributos derivados da análise de áudio, tais como *valence*, *energy* e *danceability*, os quais são calculados por modelos internos da empresa. Contudo, mais recentemente, estas plataformas têm demonstrado interesse em explorar também a dimensão textual, nomeadamente as letras, de forma a enriquecer os sistemas de recomendação e captar nuances emocionais que não estão presentes apenas no sinal de áudio.

2.2 Definição de Emoção

A palavra emoção origina-se do latim *emotio*, que significa "movimento" ou "impulso". Este termo deriva do verbo *emovere*, composto por e- ("para fora") e movere ("mover"), traduzindo-se literalmente como "mover para fora" ou "impulsionar para fora". Essa etimologia sugere que as emoções são forças internas que nos "movem", motivando-nos a agir ou reagir de maneira espontânea e, muitas vezes, involuntária. Este entendimento da emoção enquanto um impulso que nos leva à ação reflete a sua natureza intensa e transformadora.

Importa ainda distinguir entre *mood* (humor) e *emotion* (emoção). Enquanto a emoção corresponde a uma reação afetiva intensa e de curta duração, geralmente provocada por um estímulo específico, o humor refere-se a estados afetivos mais duradouros, difusos e de menor intensidade, muitas vezes sem causa claramente identificável (Gross et al., 1998; Russel, 2003). Apesar de ambos influenciarem a perceção e a experiência musical, a investigação em *Music Emotion Recognition* (MER) centra-se maioritariamente no estudo das emoções, dado que estas podem ser mais diretamente relacionadas com características musicais e mais facilmente operacionalizadas em modelos computacionais (Yang et al., 2012) .

Neste sentido, William James, em The Principles of Psychology (James, 2000), contribuiu de forma significativa para o estudo das emoções ao propor a sua conhecida Teoria de James-Lange. Segundo esta teoria, a emoção não é a causa inicial da resposta fisiológica, mas sim uma consequência desta. Por outras palavras, James defende que "a emoção é a perceção das mudanças fisiológicas que ocorrem no corpo em resposta a um evento externo". De acordo com esta visão, ao experienciar um estímulo externo – como uma situação de perigo – o corpo reage primeiro (por exemplo, com aumento do ritmo cardíaco e/ou suor), e é ao interpretar essas reações que sentimos a emoção, como medo ou ansiedade.

As emoções podem ser classificadas de diferentes formas, dependendo da abordagem teórica adotada. Katherine Hevner (1936) propôs um modelo categórico de emoções associadas à música, organizando-as em oito categorias, como alegre, tenso, triste e sereno, permitindo a análise das perceções emocionais evocadas por estímulos musicais. Outros modelos, como o *Circumplex Model of Affect* de Russell (1980) organizam as emoções em dimensões de valência (positiva ou negativa) e ativação (alta ou baixa). Estes modelos são fundamentais para a análise computacional de emoções em música (áudio, letra), permitindo, no caso da letra, a categorização e predição de sentimentos expressos em letras musicais.

2.3 Emoção Expressa, Percebida e Sentida

Gabrielsson (2002), uma das principais pesquisadoras da relação entre música e emoção, desenvolveu um modelo que distingue entre três tipos de emoções associadas à experiência musical: emoção expressa, emoção percebida e emoção sentida. Esse modelo permite compreender como a música comunica emoções e como os ouvintes interagem emocionalmente com ela, o que é essencial para a análise da música como veículo de expressão emocional.

Emoção Expressa

A emoção expressa é aquela que o compositor ou intérprete "transmite" intencionalmente através da música. Gabrielsson argumenta que as características estruturais de uma peça musical — como a melodia, harmonia, ritmo e dinâmica — são escolhidas para evocar sentimentos específicos. Em uma sinfonia alegre, por exemplo, o compositor pode usar acordes maiores, tempos rápidos e uma dinâmica mais vibrante para expressar felicidade ou excitação. O objetivo do compositor ou intérprete é comunicar uma emoção particular, independentemente do estado emocional do público.

Gabrielsson observa que a emoção expressa é frequentemente o ponto de partida para a criação musical. Compositores e intérpretes usam a música como uma "linguagem" para codificar emoções, explorando aspetos formais como repetição, variação e transposição para manipular e modelar o conteúdo emocional da peça. Isso reflete uma intenção consciente por parte do criador da obra, que usa a estrutura musical para "expressar" emoções de forma a serem reconhecíveis. Essa emoção expressa é, assim, uma projeção de uma intenção emocional que nem sempre corresponde ao sentimento real do compositor no momento da criação.

Emoção Percebida

A emoção percebida é o tipo de emoção que os ouvintes identificam ou "percebem" na música, mesmo que não a sintam pessoalmente. Em outras palavras, é a interpretação emocional da peça por parte do público. Gabrielsson aponta que o ouvinte pode ouvir uma peça melancólica e reconhecer a tristeza na música sem necessariamente se sentir triste. A emoção percebida é, portanto, uma resposta cognitiva, em que o ouvinte analisa e interpreta a "linguagem emocional" da música, sem se envolver emocionalmente com ela.

Esse conceito é particularmente importante no estudo das respostas emocionais à música, pois permite que os pesquisadores, como Gabrielsson, identifiquem como certos elementos estruturais da música estão associados a emoções específicas. Por exemplo, mudanças de tom, variações de tempo e dinâmicas intensas podem ser percebidas como emocionantes ou agitadas, enquanto harmonias suaves e progressões lentas podem ser percebidas como tranquilas ou introspetivas. A perceção da emoção é, portanto, uma interpretação do ouvinte, e Gabrielsson explora como as diferenças culturais, a formação musical e as experiências pessoais dos ouvintes podem influenciar essa perceção.

Emoção Sentida

A emoção sentida é a resposta emocional genuína e subjetiva que o ouvinte experimenta ao ouvir uma peça de música. Ao contrário da emoção percebida, que é uma identificação cognitiva, a emoção sentida é uma reação emocional real que o ouvinte vive em resposta à música. Essa emoção pode ou não coincidir com a emoção expressa pelo compositor ou intérprete. Por exemplo, uma pessoa pode ouvir uma música alegre, mas sentir-se nostálgica devido a memórias pessoais associadas àquela melodia.

Gabrielsson destaca que a emoção sentida é influenciada por fatores contextuais e pessoais, como as experiências de vida do ouvinte, seu estado de humor e o ambiente em que está ouvindo a música. A emoção sentida é, portanto, uma interação entre a música e o contexto emocional individual do ouvinte, o que torna a experiência musical única para cada pessoa. Esse aspeto emocional subjetivo é complexo e dificilmente previsível, uma vez que está profundamente ligado à vida emocional interna do ouvinte.

2.4 Modelos de Emoções

As emoções são fenómenos complexos que influenciam a experiência humana em diferentes contextos, incluindo a perceção e apreciação musical. Na área da Psicologia das Emoções, os modelos de emoção desempenham um papel central, pois permitem compreender e organizar como diferentes estímulos evocam respostas emocionais específicas.

No contexto da MER, estes modelos são fundamentais porque fornecem a base teórica para a análise computacional de sentimentos em música, quer através do áudio, quer através da letra. Ao operacionalizar a forma como as emoções podem ser representadas, estes modelos tornam possível desenvolver algoritmos capazes de categorizar e prever emoções expressas ou percebidas em músicas.

De forma geral, os modelos de emoção podem ser classificados em duas grandes abordagens: modelos categóricos e modelos dimensionais (Ekman, 1992; Russel, 1980).

- Os modelos categóricos consideram que existem emoções discretas e universais, como alegria, tristeza ou raiva.

- Os modelos dimensionais, por sua vez, representam as emoções num espaço contínuo, avaliando dimensões como a valência (positiva ou negativa) e a ativação (alta ou baixa).

A seguir, são apresentados exemplos representativos de cada uma destas abordagens.

2.4.1 Modelos Categóricos

Os modelos categóricos assumem que as emoções podem ser identificadas como tipos específicos, distintos entre si. Esta perspetiva tem origem em estudos clássicos da psicologia, como os de Ekman (1992) e Hevner (1936).

Modelo de Hevner (Figura 2): desenvolveu uma taxonomia pioneira para a música, identificando 66 adjetivos emocionais agrupados em oito categorias. Este modelo demonstrou empiricamente que determinadas características musicais evocam estados emocionais específicos, tornando-se uma das primeiras referências estruturadas em MER.

Modelo de Ekman: identificou um conjunto de emoções básicas universais (alegria, tristeza, raiva, medo, surpresa e nojo), reconhecíveis em todas as culturas através das expressões faciais. Este modelo categórico é amplamente utilizado por demonstrar que certos estados emocionais são biologicamente inatos e transversais, o que também tem implicações para a música e a comunicação emocional.

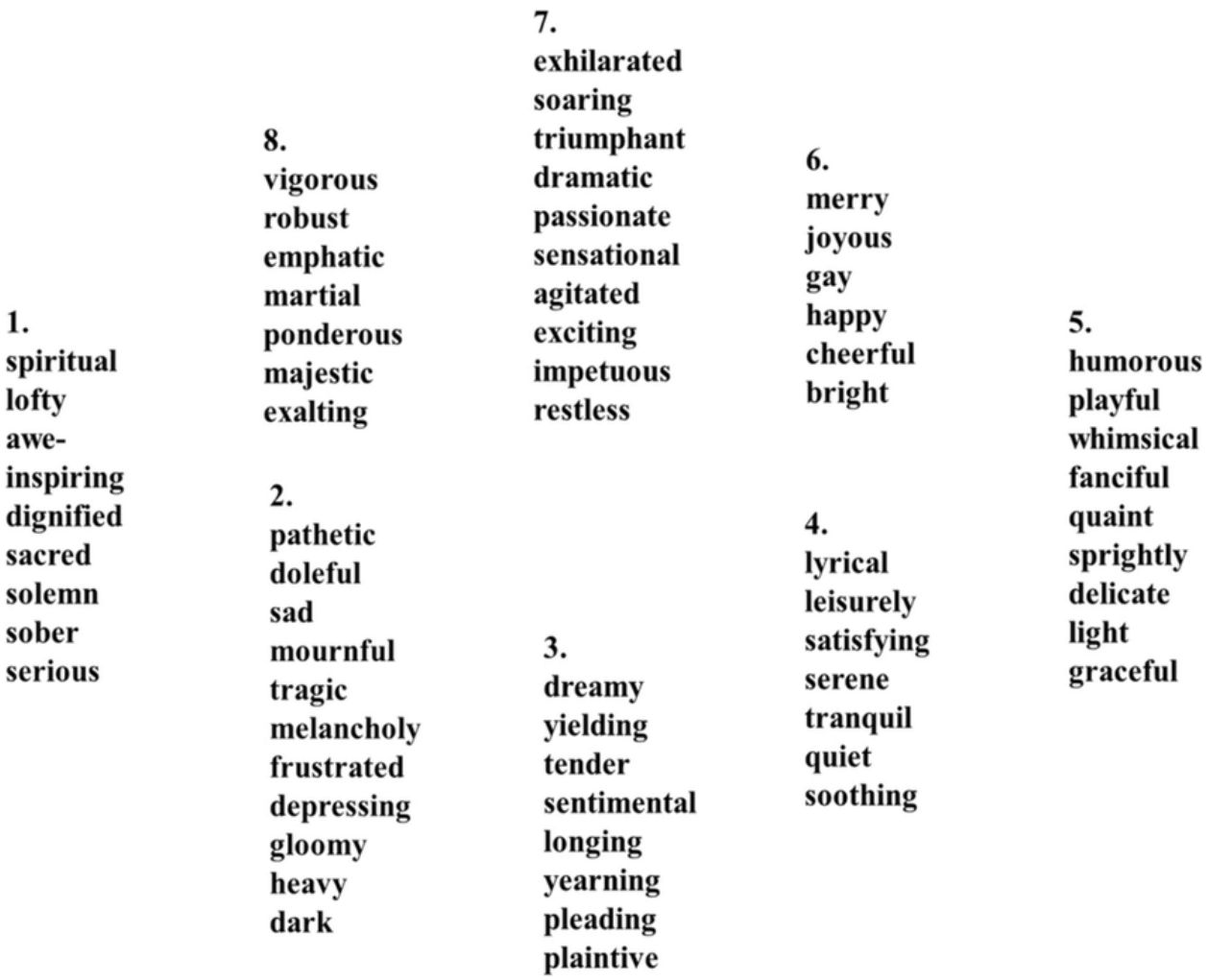

Figura 2- Modelo de Hevner [Fonte:https://lc.cx/RALhlk]

2.4.1 Modelos Dimensionais

Um dos modelos dimensionais mais conhecidos é o modelo circumplexo de Russell (1980), onde as emoções são posicionadas em um plano bidimensional composto por dois eixos, designados como valência e ativação. A valência refere-se à polaridade da emoção, que pode ser positiva (agradável) ou negativa (desagradável), enquanto a ativação indica o nível de excitação ou intensidade associado à emoção, variando de alta intensidade a baixa intensidade.

O modelo circumplexo (*Figura 3*) sugere que as emoções não são discretas, mas sim representadas como pontos em um contínuo, o que permite uma maior flexibilidade na compreensão das emoções humanas. Esse modelo baseia-se na ideia de que todas as emoções podem ser descritas em termos dessas duas dimensões, facilitando uma análise mais integrada e menos ambígua do estado emocional de um indivíduo. O modelo de Russell é amplamente utilizado em pesquisas sobre emoções, pois oferece uma forma prática e intuitiva de mapear a complexidade emocional em um espaço bidimensional.

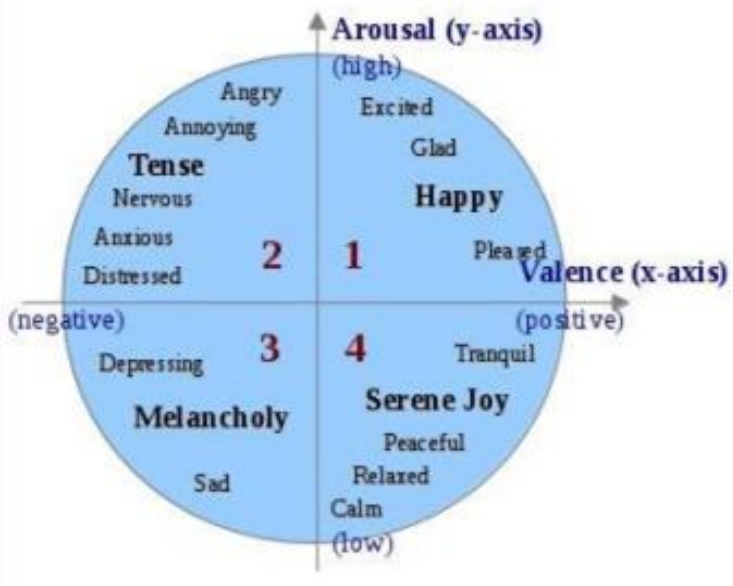

Figura 3- Modelo circumplexo de Russell [Fonte: Ricardo Manuel da Silva Malheiro, 2016 pag.45]

2.5 Sumarização Automática de Textos

A sumarização automática de textos visa extrair as partes mais importantes de um documento, produzindo um resumo conciso que mantém a essência do conteúdo original. No contexto das letras de música, a sumarização pode ajudar a identificar frases e versos-chave que capturam os temas e emoções principais.

2.5.1 Sumarização Extrativa

A sumarização extrativa consiste em selecionar frases ou sentenças diretamente do texto original, formando um resumo que preserva a estrutura linguística existente. Esta abordagem baseia-se em identificar as partes mais relevantes de um documento a partir de métricas como frequência de palavras, posição das frases ou centralidade em grafos de similaridade (Nenkova & McKeown, 2012). Embora não produza novas construções linguísticas, garante elevada fidelidade ao conteúdo original. Métodos clássicos como o algoritmo de Luhn, LSA (*LAtent Semantic Analysis*), LexRank e TextRank exemplificam esta categoria, sendo amplamente utilizados como baseline em estudos de sumarização. Podemos ver em seguida:

(a) Luhn Algorithm

O algoritmo de Luhn é considerado um dos primeiros métodos de sumarização automática. Propõe identificar palavras-chave com base na frequência e importância no texto, descartando palavras de função (como artigos e preposições). As frases que contêm maior densidade de palavras relevantes são selecionadas para compor o resumo (Luhn, 1958).

Vantagens: Simplicidade, baixo custo computacional e aplicável em textos curtos.

Limitações: Desconsidera relações semânticas profundas, podendo gerar resumos fragmentados ou incoerentes.

Aplicações: Embora obsoleto em comparação com métodos modernos, é usado como baseline em estudos de sumarização para demonstrar avanços sobre os métodos pioneiros.

(b) Latent Semantic Analysis (LSA)

O LSA aplica decomposição de matrizes (*Singular Value Decomposition* – SVD) sobre a matriz termo-documento, identificando padrões semânticos latentes. Em sumarização, frases com maior relevância em tópicos principais são selecionadas (Gong & Liu, 2001).

Vantagens: Capta relações semânticas além da simples frequência de palavras; eficaz em textos técnicos.

Limitações: Depende de uma representação vetorial simples (saco de palavras), perdendo contexto sintático; escalabilidade limitada em grandes corpora.

Aplicações: Usado em áreas técnicas e científicas para sumarizar relatórios ou artigos, especialmente quando a terminologia é repetitiva.

(c) LexRank

O LexRank (Erkan & Radev, 2004) é um algoritmo baseado em grafos, onde cada frase é um nó e as ligações representam a similaridade entre frases (geralmente medida pelo cosseno entre vetores TF-IDF). Inspirado no PageRank, atribui maior importância às frases mais centrais no grafo, isto é, as que mais se relacionam com outras.

Vantagens: Não supervisionado, robusto, adaptável a diferentes domínios.

Limitações: Pode introduzir redundância (frases semelhantes são escolhidas); depende fortemente de métricas de similaridade lexical.

Aplicações: Usado em competições como DUC (*Document Understanding Conference*) e TAC (*Text Analysis Conference*) como baseline de comparação com modelos modernos.

(d) TextRank

O TextRank (Mihalcea & Tarau, 2004) segue a mesma lógica de grafos do LexRank, mas não depende de representações vetoriais explícitas. Ele constrói um grafo de sentenças conectadas pela coocorrência de palavras em janelas deslizantes. Aplica o algoritmo PageRank para identificar as frases mais relevantes.

Vantagens: Simples, não supervisionado, não requer treino, aplicável a múltiplos idiomas.

Limitações: Resumos podem ser redundantes; não captam relações semânticas profundas.

Aplicações: Muito usado em textos jornalísticos e científicos; disponível em bibliotecas populares de NLP (*Natural Language Processing*) (ex.: sumy, gensim).

2.5.2 Sumarização Abstrativa

A sumarização abstrativa procura gerar resumos em linguagem natural, reformulando frases e criando expressões que não precisam estar presentes no texto original. Aproxima-se do modo como humanos sintetizam informação, mas exige maior capacidade computacional e modelos de linguagem mais avançados [Rush et al., 2015]. Ao contrário da abordagem extrativa, esta técnica pode introduzir generalizações, paráfrases ou até nova informação inferida. Modelos baseados em *transformers*, alcançando resultados superiores em *benchmarks* internacionais de sumarização automática. Em seguida podemos ver:

(a) BART

O BART (Lewis et al., 2020), desenvolvido pela Meta AI, é um modelo seq2seq baseado em transformers que combina pré-treino auto-regressivo e autoencoder. Treinado para reconstruir texto corrompido, torna-se especialmente eficaz em tarefas de geração de linguagem como a sumarização abstrativa.

Vantagens: Produz resumos naturais e gramaticalmente corretos; captura nuances semânticas.

Limitações: Requer grande capacidade computacional; pode introduzir informação não presente no texto.

Aplicações: Usado em *benchmarks* como CNN/DailyMail (Hermann et al., 2015), sendo referência no estado da arte em geração de resumos.

(b) DistilBART

O DistilBART é uma versão reduzida do BART, obtida via *knowledge distillation*. Mantém grande parte do desempenho, mas com menos parâmetros.

Vantagens: Menor custo computacional, adequado para aplicações em tempo real ou em dispositivos com recursos limitados.

Limitações: Ligeira perda de qualidade em relação ao BART.

Aplicações: Usado em contextos móveis e aplicações que exigem eficiência sem comprometer muito a fluência.

(c) LED (Longformer Encoder-Decoder)

O LED (Beltagy et al., 2020) foi projetado para lidar com textos muito longos (até 16k tokens). Utiliza atenção local combinada com atenção global para processar documentos extensos.

Vantagens: Permite resumir relatórios longos, artigos científicos ou documentos jurídicos.

Limitações: Exige otimização de memória e treino específico para domínios longos.

Aplicações: Ideal em áreas médicas, jurídicas ou científicas.

(d) PEGASUS

O PEGASUS (Zhang et al., 2019) introduziu a técnica de *gap-sentence generation* no pré-treino: o modelo aprende a prever frases omitidas de um texto, o que aproxima a tarefa de pré-treino da sumarização.

Vantagens: Resultados de ponta em *benchmarks* de sumarização; capta bem informação semântica global.

Limitações: Exigente em recursos; pode gerar alucinações textuais.

Aplicações: Sumarização de notícias (CNN/DailyMail, XSum) e relatórios técnicos.

(e) T5

O T5 (Raffel et al., 2020) unificou tarefas de PLN (*Processamento de Linguagem Natural*) num paradigma de texto-para-texto. Para sumarização, basta aplicar o prefixo “summarize:” ao texto de entrada.

Vantagens: Altamente versátil; pode ser adaptado a múltiplas tarefas além da sumarização.

Limitações: Modelos grandes exigem hardware especializado.

Aplicações: *Benchmarks* de *summarization*, tradução, QA (*Question Answering*).

2.5.3 Large Language Models (LLMs)

Os *Large Language Models* (LLMs) representam a evolução mais recente na área da sumarização, sendo modelos de redes neuronais profundas com mil milhões de parâmetros, treinados em grandes volumes de texto. Ao captarem dependências de longo alcance e compreenderem contexto de forma sofisticada, conseguem executar tanto sumarização extrativa como abstrativa (Bommasani et al., 2021). Modelos como BERT, especializado em extração de frases relevantes, e GPT-2, com capacidades generativas para resumos mais criativos, ilustram a versatilidade dos LLMs. Estes modelos aproximam a sumarização automática da produção humana de texto, embora ainda apresentem desafios como viés, alucinações e custos elevados de treino. Em seguida:

(a) BERT

O BERT (Devlin et al., 2018) é um modelo bidirecional de transformers. Em sumarização é aplicado em versões extrativas, onde embeddings contextuais servem para classificar quais frases devem ser mantidas no resumo.

Vantagens: Excelente compreensão contextual.

Limitações: Não gera texto novo; depende de *fine-tuning*.

Aplicações: Resumos extrativos em textos científicos e jornalísticos.

(b) GPT-2

O GPT-2 (Radford et al., 2019a), modelo generativo da OpenAI, pode ser usado para sumarização abstrativa via *prompt engineering*.

Vantagens: Capaz de produzir resumos fluidos e criativos.

Limitações: Não é especializado em sumarização; risco de gerar informação incorreta.

Aplicações: Contextos criativos e experimentais de sumarização.

Em síntese, a literatura evidencia a evolução das técnicas de sumarização desde abordagens extrativas clássicas, baseadas em estatísticas de frequência e relevância de frases, até métodos abstrativos, que exploram modelos de linguagem avançados para gerar resumos mais naturais e contextualmente adequados. No presente trabalho, estas diferentes famílias de algoritmos são retomadas (Capítulo 4), onde serão comparadas de forma sistemática com métodos mais simples (baseline extrativos), permitindo avaliar os ganhos proporcionados pelas abordagens mais recentes.

2.6 Extração de Features

A extração de features constitui uma etapa fundamental em diversas áreas como por exemplo no reconhecimento automático de emoções em letras musicais (*Lyric-based Music Emotion Recognition* — LMER). As features representam informações linguísticas, estilísticas ou semânticas que traduzem propriedades do texto em variáveis numéricas, permitindo o seu processamento por algoritmos de aprendizagem automática. A literatura identifica diferentes categorias de *features*, cada uma capturando aspetos específicos da linguagem, desde a frequência de palavras até à organização estrutural da letra (Hu et al., 2009).

Neste trabalho, são consideradas quatro grandes categorias de *features*: *Content-Based Features* (CBF), *Stylistic-Based Features* (StyBF), *Structural-Based Features* (StruBF) e *Semantic-Based Features* (SemBF).

2.6.1 Content-Based Features (CBF):

As features baseadas no conteúdo são as mais utilizadas em tarefas de análise de texto e incluem representações como o *Bag-of-Words* (BoW) (Sebastiani, 2002). O BoW representa o texto como um conjunto de palavras ou sequências de palavras (*n-grams*), desconsiderando a ordem sintática, mas preservando a frequência ou presença dos termos. Assim, as letras são transformadas em vetores numéricos que refletem padrões lexicais diretamente relacionados ao conteúdo.

O processo de BoW é normalmente acompanhado por operações de pré-processamento, como tokenização, remoção de stopwords e stemming. A tokenização divide o texto em unidades básicas (tokens), enquanto a remoção de stopwords elimina palavras muito frequentes (como "o", "a", "de") que pouco contribuem para a distinção emocional. Já o stemming reduz as palavras às suas raízes, agrupando variantes morfológicas (e.g., “cantar”, “cantava” → “cant”).

Além das palavras, também se utilizam n-grams (e.g., unigramas, bigramas, trigramas), que capturam contexto progressivamente mais amplo. Unigramas representam palavras isoladas, enquanto bigramas e trigramas conseguem refletir sequências que transportam significado emocional mais rico (e.g., “sem ti”, “tenho saudade”). Outro recurso importante são as etiquetas gramaticais (POS tags), que classificam as palavras em categorias gramaticais como nomes, verbos ou adjetivos. Estas permitem criar n-grams de POS, úteis para captar padrões linguísticos além do vocabulário explícito (Mayer et al., 2008).

O BoW, com ou sem estas transformações, continua a ser uma técnica fundamental em LMER, servindo muitas vezes como baseline em estudos experimentais pela sua simplicidade e eficácia.

2.6.2 Stylistic-Based Features (StyBF):

As features estilísticas captam aspetos relacionados ao estilo de escrita, refletindo escolhas linguísticas que podem transmitir emoções implícitas. Em letras musicais, o estilo pode estar associado ao género musical, ao perfil do compositor ou à intenção emocional subjacente.

Um exemplo comum é o uso das categorias gramaticais (POS), como adjetivos, advérbios e interjeições, que frequentemente carregam valor afetivo. A presença de advérbios de intensidade (e.g., “muito”, “tão”) ou de interjeições (e.g., “ah”, “oh”) pode indicar maior expressividade emocional. Também se analisam padrões gráficos, como o número de palavras em maiúsculas (ACL), que podem sugerir ênfase ou intensidade, e palavras iniciadas em maiúsculas fora do início de frases (FCL), que podem refletir recursos estilísticos específicos.

Outro exemplo relevante é o uso de gíria (slang), especialmente em géneros como hip-hop ou rap, onde a linguagem informal e popular está fortemente ligada à expressão de identidade e emoção (Hu et al., 2009). Assim, a presença de palavras de gíria pode atuar como indicador estilístico de determinadas emoções ou de contextos culturais específicos.

Estas features, ao captarem o estilo e não apenas o conteúdo, complementam as análises baseadas em BoW, tornando possível identificar padrões emocionais menos explícitos, mas relevantes.

2.6.3 Structural-Based Features (StruBF):

As features estruturais estão relacionadas com a organização formal das letras, ou seja, a forma como versos e refrões são dispostos. Apesar de pouco exploradas na literatura de LMER, representam uma dimensão importante, pois a estrutura pode transmitir intenções emocionais específicas (Malheiro et al., 2016).

Exemplos incluem o número de repetições do refrão (#CH), o número de vezes em que o título aparece na letra (#Title), ou a proporção entre versos e refrões. Também se analisam padrões estruturais mais complexos, como a alternância entre versos e refrões (VCVC, VCCVCC, etc.) e a presença de repetições prolongadas de refrões no final da música, frequentemente associadas a géneros mais dançáveis.

2.6.4 Semantic-Based Features (SemBF):

As features semânticas exploram diretamente o significado das palavras, indo além da forma ou frequência. Este tipo de abordagem recorre a dicionários, recursos linguísticos e frameworks que categorizam palavras em dimensões afetivas ou semânticas.

Entre os recursos mais utilizados destacam-se:

- LIWC (*Linguistic Inquiry and Word Count*) (Tausczik & Pennebaker, 2010), que classifica palavras em categorias emocionais, cognitivas e sociais, contabilizando cerca de 80 a 90 categorias (features) em versões mais recentes.
- General Inquirer (GI) (Stone, Philip J., Dunphi Dexter C., Smith S. Marshall, 1966), que organiza palavras em 182 categorias semânticas.
- ANEW (*Affective Norms for English Words*) (Bradley & Lang, 1999), que disponibiliza valores de valência, ativação e dominância para cerca de 1.034 palavras em inglês.
- DAL (*Dictionary of Affect in Language*) (Whissell, 2009), que contém cerca de 8.742 palavras anotadas em dimensões afetivas, permitindo calcular métricas psicológicas como energia, avaliação e potência.

Além disso, ferramentas como o *Synesketch* e a *ConceptNet* acrescentam informação semântica e relacional ao texto. Consiste na criação de *gazetteers* baseados no modelo de Russell (1980), onde listas de palavras são associadas a cada quadrante do espaço valência-ativação. Assim, é possível medir até que ponto uma letra contém palavras relacionadas a estados emocionais como alegria, tristeza, ansiedade ou serenidade.

Estas features são cruciais em LMER porque permitem ligar diretamente o vocabulário da letra às emoções humanas, representando de forma mais próxima o conteúdo semântico e afetivo do texto.

2.7 Abordagens de Classificação

A classificação é uma etapa central em tarefas de *Music Emotion Recognition* (MER), permitindo associar as features extraídas das letras ou do áudio a categorias emocionais ou quadrantes de modelos psicológicos. Neste trabalho, consideram-se tanto modelos clássicos de machine learning como modelos baseados em *deep learning* e linguagem natural, nomeadamente:

Support Vector Machine (SVM)

O *Support Vector Machine* (SVM) é um algoritmo supervisionado introduzido por Boser, Guyon e Vapnik (1992), amplamente utilizado em problemas de classificação de texto e análise de sentimentos. O SVM procura identificar o hiperplano ótimo que maximiza a margem entre diferentes classes num espaço vetorial.

Apesar de originalmente concebido para classificação binária, o método pode ser adaptado a problemas multiclasse através de estratégias como one-vs-rest ou one-vs-one. A utilização de funções kernel (linear, polinomial, radial basis function) permite ao SVM lidar com padrões lineares e não lineares, tornando-o versátil.

Na área de MER, o SVM tem sido uma escolha frequente devido à sua robustez em *datasets* de média dimensão e à capacidade de generalizar bem mesmo em cenários de alta dimensionalidade (como em representações TF-IDF de lyrics). Contudo, apresenta como limitação um custo computacional elevado em datasets muito grandes.

Random Forest

O *Random Forest*, proposto por Breiman (2001),é um algoritmo de ensemble learning baseado na construção de múltiplas árvores de decisão treinadas em subconjuntos aleatórios do dataset (técnica de bagging). A predição final resulta da agregação (votação) das saídas das várias árvores, o que garante maior robustez e menor risco de overfitting.

Este modelo é considerado um baseline sólido para tarefas de classificação multiclasse em lyrics, dado que lida bem com features de natureza heterogénea (lexicais, estilísticas, semânticas). No entanto, apresenta como desvantagem a menor interpretabilidade em comparação com modelos mais simples, já que o número elevado de árvores dificulta a compreensão das regras de decisão.

Multilayer Perceptron (MLP)

O *Multilayer Perceptron* (MLP) é um tipo de rede neuronal artificial *feedforward*, composta por uma camada de entrada, múltiplas camadas ocultas e uma camada de saída. Cada neurónio realiza uma transformação linear dos inputs seguida de uma função de ativação não linear, como ReLU ou tanh (DE Rumelhart et al., 1986; F Rosenblatt, 1958).

A capacidade do MLP em modelar relações não lineares complexas torna-o adequado para problemas de classificação em MER, sobretudo quando se pretende capturar padrões latentes em features semânticas ou embeddings. No entanto, exige maior quantidade de dados e cuidado com a regularização, devido à sua propensão ao overfitting.

H20 (MLP)

Nesta investigação, foi utilizado o H2O Deep Learning Estimator, disponibilizado pela framework H2O.ai. O modelo consiste num MLP configurado com duas camadas ocultas (100 e 50 neurónios), funções de ativação do tipo RectifierWithDropout e regularização L1 e L2. Adicionalmente, foram aplicados mecanismos de paragem antecipada (early stopping) para evitar overfitting e otimizar o treino.

Este tipo de abordagem permite explorar a flexibilidade de redes neurais profundas em contextos de classificação de emoções, conciliando performance elevada com controlo da complexidade computacional.

BERT (Bidirectional Encoder Representations from Transformers)

O BERT, desenvolvido pela Google (Jacob Devlin et al., 2019), é um modelo de linguagem baseado na arquitetura Transformer (Vaswani et al., 2017). A sua principal inovação é a atenção bidirecional, que permite compreender o contexto de uma palavra considerando simultaneamente as palavras à esquerda e à direita.

No âmbito da sumarização e da classificação de emoções em lyrics, o BERT é utilizado para extrair embeddings contextuais ricos, que alimentam classificadores *downstream*. A sua capacidade de capturar nuances semânticas torna-o especialmente eficaz para diferenciar emoções próximas, como tristeza vs. melancolia.

GPT2

O GPT-2, desenvolvido pela OpenAI (Radford et al., 2019), é um modelo de linguagem baseado em transformadores unidirecionais (causais). Originalmente concebido para geração de texto, o GPT-2 pode ser adaptado a tarefas de classificação utilizando os seus embeddings de saída como representações do texto.

Apesar de não ter sido desenhado especificamente para classificação de emoções, o GPT-2 mostra-se útil em contextos onde se valoriza criatividade e flexibilidade na interpretação semântica. No entanto, apresenta maior risco de alucinação textual e exige um ajustamento mais cuidadoso do que modelos como BERT.

2.8 Métricas para Comparação de Modelos de Classificação:

A comparação entre diferentes modelos de classificação é uma etapa essencial em projetos de Aprendizagem Automática, especialmente quando o objetivo é selecionar o modelo que oferece melhor desempenho, maior robustez e que melhor se adapta ao problema em estudo. Conforme sugerem Kuhn e Johnson (2013), a avaliação objetiva do desempenho dos modelos permite não apenas a escolha do melhor algoritmo, mas também uma compreensão mais aprofundada das limitações e potencialidades de cada abordagem.

Acurácia (Accuracy)

A acurácia é uma das métricas mais intuitivas e refere-se à proporção de previsões corretas face ao total de previsões. É uma métrica útil em contextos onde as classes estão aproximadamente equilibradas. No entanto, em situações com classes não balanceadas, a acurácia pode ser enganadora, já que um modelo pode alcançar um valor elevado simplesmente ao privilegiar a classe maioritária (Géron, 2022).

Precisão, Revocação e F1-Score

Para problemas com desequilíbrio entre classes, torna-se essencial recorrer a métricas mais sensíveis, como a precisão (precision), a revocação (recall) e o F1-score. A precisão mede a proporção de exemplos classificados como positivos que são de facto positivos, enquanto a revocação quantifica a proporção de exemplos positivos que foram corretamente identificados pelo modelo. O F1-score, introduzido por Sokolova & Lapalme (2009), é a média harmónica entre precisão e revocação, sendo particularmente útil quando é necessário um equilíbrio entre ambas (Sokolova & Lapalme, 2009).

Matriz de Confusão

A matriz de confusão permite visualizar a performance do modelo em relação a cada classe específica, fornecendo uma representação mais detalhada do tipo de erros cometidos. É particularmente valiosa quando se deseja compreender quais classes estão a ser confundidas entre si.

Curvas ROC e AUC

A curva ROC (*Receiver Operating Characteristic*) e a métrica AUC (*Area Under the Curve*) foram introduzidas originalmente em contextos de radar e adaptadas à estatística e à aprendizagem automática. A ROC avalia a taxa de verdadeiros positivos contra a taxa de falsos positivos, enquanto a AUC resume essa relação num único valor. Conforme Powers (2011) destaca, a AUC é especialmente eficaz em situações de não balanceamento entre classes e quando o objetivo é medir a capacidade discriminativa do modelo.

Validação Cruzada

Outra abordagem crucial para a comparação de modelos é a validação cruzada, mais especificamente a validação cruzada *k-fold*. Esta técnica, amplamente recomendada por Kuhn e Johnson (2013), divide os dados em k subconjuntos (ou "folds"), utilizando k-1 para treino e um para teste, de forma iterativa. Este processo reduz o viés associado a uma única divisão dos dados e proporciona uma estimativa mais fiável da generalização do modelo.

Fine-tuning de Modelos de Linguagem de Grande Escala (LLMs)

O *fine-tuning* consiste no processo de ajustar um modelo de linguagem pré-treinado (LLM – *Large Language Model*) a uma tarefa ou domínio específico, usando um conjunto de dados adicional. Enquanto o pré-treinamento fornece ao modelo uma compreensão ampla da linguagem, o fine-tuning permite que ele aprenda nuances mais específicas do contexto desejado, como linguagem técnica, estilo de escrita ou tipo de conteúdo.

Durante o fine-tuning, os pesos do modelo são atualizados com base num novo conjunto de dados rotulado, o que resulta numa melhoria do desempenho do modelo em tarefas direcionadas, como classificação de texto, sumarização, reconhecimento de emoções ou resposta a perguntas. Este processo é geralmente mais leve e eficiente do que treinar um modelo do zero, pois aproveita o conhecimento adquirido no pré-treinamento de grandes volumes de dados.

3. DESCRIÇÃO DO CORPUS

O *dataset* utilizado neste estudo foi fornecido no âmbito do projeto MERGE[1] (Music Emotion Recognition New Generation), desenvolvido pela equipa de investigação da qual faz parte o Professor Doutor Ricardo Malheiro (Centro de Informática e Sistemas da Universidade de Coimbra - CISUC)[2]. Este projeto teve como objetivo a construção de um corpus anotado de letras musicais, que permitisse explorar de forma sistemática a classificação e regressão de emoções musicais, recorrendo ao modelo dimensional de Russell (1980).

O corpus completo incluía diferentes modalidades:

- 162 instâncias com áudio
- 180 instâncias com letra
- 133 instâncias bimodais (áudio + letra)

Nesta tese optou-se por utilizar apenas a componente textual (180 letras musicais), uma vez que o foco do trabalho é avaliar o impacto da sumarização automática na análise de emoções a partir de letras.

Link: https://mir.dei.uc.pt/downloads.html (Lyrics emotion dataset (Russell's model) (2016)).

O dataset foi construído no âmbito do artigo “Emotionally-Relevant Features for Classification and Regression of Music Lyrics” (Malheiro et al., 2018), que aborda o reconhecimento de emoções em músicas, com foco nas letras, utilizando o modelo dimensional de emoções de Russell. O estudo procura melhorar a precisão na identificação de emoções em músicas, aproveitando características da letra de música específicas e avançando o estado da arte na área de MER.

O estudo teve como objetivos principais:

- Criar um dataset anotado manualmente que pudesse ser utilizado para treino e validação de modelos de reconhecimento de emoções.
- Desenvolver e explorar novas *features* de letra de música, complementando aquelas já amplamente utilizadas, como *Bag-of-Words* (BOW) e *Part-of-Speech* (POS).
- Implementar métodos de classificação e regressão para prever as dimensões de valência e ativação, bem como os quadrantes emocionais.
- Resumo do processo de construção do dataset:

[1] https://www.cisuc.uc.pt/en/projects/MERGE
[2] https://www.cisuc.uc.pt/en

Características do *dataset*:

- Diversidade de géneros musicais e épocas.
- Distribuição uniforme pelos 4 quadrantes do modelo emocional de Russell.
- Cada música pertence predominantemente a um dos 4 quadrantes.

A Figura 4 apresenta a distribuição das 180 músicas no espaço bidimensional definido pelas dimensões Valência (eixo horizontal) e Ativação (eixo vertical), de acordo com o modelo de Russell (1980). Como se observa, os pontos encontram-se repartidos de forma clara pelos quatro quadrantes emocionais, sem concentrações excessivas em apenas uma região.

Esta distribuição confirma que o corpus se encontra relativamente balanceado, assegurando representatividade em todas as combinações de valência e ativação. Esse equilíbrio é fundamental para a fiabilidade dos experimentos de classificação, pois reduz enviesamentos durante o treino e garante que os modelos tenham oportunidade de aprender padrões emocionais em diferentes contextos.

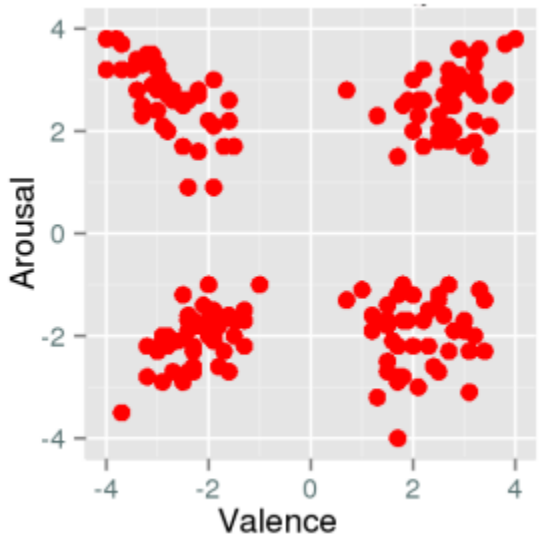

Figura 4: Distribuição das músicas por quadrantes [Fonte: Ricardo Manuel da Silva Malheiro, (2016) pag.54]

4. PROCESSO DE SUMARIZAÇÕES

Para a construção das sumarizações das letras musicais foram implementados, em ambiente Jupyter Lab, diferentes algoritmos de natureza extrativa e abstrativa, bem como modelos de *Large Language Models* (LLMs). O objetivo foi reduzir o tamanho das letras sem comprometer a informação relevante para a tarefa de classificação de emoções, preservando o conteúdo semântico e estilístico necessário à correta identificação do quadrante emocional.

4.1 Sumarização Extrativa

Na vertente extrativa foram utilizados quatro algoritmos clássicos: Luhn, Latent Semantic Analysis (LSA), LexRank e TextRank. Estes métodos foram implementados com recurso a bibliotecas como sumy e gensim, parametrizados para extrair uma percentagem fixa do texto (número de frases). O processo consistiu em:

1. Pré-processamento das letras (remoção de caracteres especiais e normalização textual).
2. Aplicação dos algoritmos, que selecionaram as frases mais relevantes com base em estatísticas de frequência e coesão semântica.
3. Geração de resumos de tamanho controlado, posteriormente usados nos classificadores de emoção.

Na definição do tamanho dos resumos extrativas (LexRank, LSA, Luhn, TextRank) foi usada a unidade natural de compressão que é a frase, uma vez que estas técnicas selecionam diretamente partes do texto original com base em medidas de relevância. Assim, foi adotado um valor fixo de 3 frases por resumo.

4.2 Sumarização Abstrativa

Na vertente abstrativa foram explorados modelos baseados em transformers, nomeadamente BART, DistilBART, Longformer Encoder-Decoder (LED), PEGASUS e T5, através da biblioteca *Hugging Face Transformers*. Estes modelos foram aplicados para gerar versões reformuladas das letras, sintetizando o conteúdo em linguagem natural.

Durante os testes verificaram-se limitações em alguns modelos, como o PEGASUS-XSum e o T5-Small, que introduziam repetições excessivas ou texto irrelevante (ex.: “Check out 888-492-0's full tracklisting below”). Devido a esses problemas, optou-se pela exclusão dessas sumarizações no processo de classificação.

A definição do tamanho dos resumos das abstrativas (T5, BART, PEGASUS, LED, DistilBART) foi controlado por parâmetros de geração (max_length e min_length), definidos em tokens. Para garantir consistência, definiu-se como limite principal 150 tokens (com

min_length de 30), permitindo gerar resumos compactos mas semanticamente informativos. No caso do T5, foram ainda testadas variantes com 100 e 200 tokens, de forma a avaliar o impacto do tamanho na qualidade do resumo e no desempenho da classificação.

4.3 Large Language Models (LLMs)

Adicionalmente, foram testados modelos de larga escala como BERT (para extração de segmentos relevantes) e GPT-2 (para geração de resumos abstrativos). Inicialmente, estes modelos foram utilizados sem qualquer limite de comprimento, situação em que os resumos produzidos apresentaram dimensões muito próximas do texto original, não cumprindo assim o objetivo de compressão. Por esse motivo, considerou-se uma segunda configuração com limite de 150 tokens, permitindo avaliar o impacto de uma restrição mais rígida no comprimento dos resumos, tanto na preservação da coerência como na retenção da informação emocional.

4.4 Pós-Processamento e Validação das Sumarizações

Durante a experimentação identificaram-se problemas recorrentes, tais como:

- inserção de caracteres especiais e espaços desnecessários,
- repetição de frases em loop,
- introdução de conteúdo genérico não relacionado com a letra.

Para mitigar estes problemas foi necessário aplicar um pós-processamento, que incluiu:

- limpeza de espaços e remoção de caracteres inválidos,
- eliminação de frases repetidas ou incoerentes,
- normalização textual para garantir consistência.

A qualidade das sumarizações foi avaliada de forma manual e exploratória, através da inspeção aleatória de resumos, verificando se o texto mantinha relação com o conteúdo emocional da música. Além disso, a validação final foi realizada de forma indireta, analisando o impacto das sumarizações no desempenho dos classificadores de emoções.

4.5 Estratégia de Escolha

A utilização de diferentes abordagens (extrativas, abstrativas e LLMs) teve um caráter comparativo e exploratório. O objetivo não foi determinar um modelo universalmente superior, mas sim compreender como diferentes famílias de algoritmos afetam a tarefa de *Music Emotion Recognition* (MER) baseada em letras.

- Métodos extrativos: adotados como baseline, pela sua simplicidade e interpretabilidade, servindo de ponto de referência inicial.
- Modelos abstrativos: permitem avaliar se a reformulação das letras em linguagem mais fluida preserva ou distorce os sinais emocionais relevantes.
- LLMs (*Large Language Models*): foram incluídos pela sua capacidade de modelar contexto em larga escala, de forma a avaliar o seu potencial em domínios específicos como a música.

Assim, o desenho experimental permitiu comparar diretamente o baseline extrativo com as técnicas mais avançadas do estado da arte, avaliando até que ponto a maior complexidade algorítmica se traduz em ganhos efetivos no reconhecimento de emoções musicais.

A Tabela 1 apresenta a quantidade total de palavras resultante de cada técnica de sumarização em comparação com as letras originais (41.999 palavras). Verifica-se que os métodos extrativos, como TextRank e Luhn, mantêm uma maior proporção do texto (entre 39% e 47%), ou seja, preservam uma parte substancial do comprimento original porque selecionam frases completas sem reescrita. Já os métodos abstrativos, como DistilBART ou PEGASUS, geram resumos bastante mais compactos (de 16% a 19%), sendo “compactos” aqui entendido como maior grau de compressão lexical, resultante da capacidade de reescrever e condensar informação. Por outro lado, o GPT-2, por ser mais generativo, produziu resumos significativamente mais extensos (77% do total), no sentido de apresentarem menor compressão e maior tendência a expandir ou reformular partes do texto. Estes resultados confirmam que o grau de compressão textual varia substancialmente entre técnicas, influenciando diretamente a densidade de informação preservada para a classificação emocional.

Tabela 1: Quantidade de Palavras por Total Sumarização

Tipo Sumarização:	Quantidade Total de palavras	% de número de palavras face ao número de palavras da letra original
Features_Original_letra	41999	100%
abstractive_distil_bart	6493	16%
abstractive_led	13079	31%
abstractive_pegasus_dailymail	8161	19%
abstractive_t5	12751	30%
abstrativa_distilbart	6530	16%
Features_abstractive_bart	6756	16%
lex_rank	16244	39%
lsa	16852	40%
luhn	19228	46%
text_rank	19909	47%
z.BERT	12998	31%
z.GPT2	32599	77%

5. PROCESSO DE CLASSIFICAÇÃO

O processo de classificação desenvolvido nesta investigação teve como objetivo identificar padrões relevantes nas letras musicais e nos seus resumos, de forma a treinar modelos capazes de distinguir diferentes classes. Para tal, foi seguida uma abordagem estruturada, que envolveu a extração de *features* a partir dos textos originais e das versões sumarizadas, a seleção das *features* mais representativas e, finalmente, a implementação de vários modelos de classificação para avaliação comparativa do desempenho.

5.1 Extração de Features

As *features* utilizadas para o treino dos modelos foram geradas por um extrator externo, responsável por automatizar o pré-processamento e a extração de características textuais. O extrator adotado neste projeto tem como base o trabalho desenvolvido por Malheiro (2016) na sua tese de doutoramento “Emotion-based analysis and classification of music lyrics”, realizada no âmbito do Programa Doutoral em Ciências e Tecnologias da Informação da Universidade de Coimbra.

Neste trabalho, o autor propôs novos métodos de análise de emoções em letras de músicas, incluindo o desenvolvimento de um extrator de features originalmente implementado em Java. Para o presente projeto, foi utilizada a versão atualizada em Python, disponibilizada no âmbito do projeto desenvolvido pela equipa do projeto MERGE *Feature Extraction System*, que implementa um sistema robusto para extração de diferentes tipos de atributos linguísticos.

Após a obtenção das letras originais e das versões sumarizadas, procedeu-se à extração de features em ambas as versões, de acordo com quatro categorias principais:

5.1.1 Content-Based Features (CBF)

As features baseadas em conteúdo foram construídas a partir de representações lexicais simples, tendo em conta diferentes níveis de contexto (palavras isoladas, pares de palavras e sequências de três). Para este fim, foram extraídos três conjuntos específicos:

- CBF_unig_nada_freq.csv: inclui unigramas (palavras isoladas) sem transformação adicional, representados pela sua frequência absoluta. Este tipo de representação captura a ocorrência de termos individuais relevantes para o reconhecimento emocional. Unigramas + nenhuma transformação + frequência absoluta (freq)
- CBF_big_st_bool.csv: contém bigramas (pares de palavras consecutivas), processados com stemming (redução à raiz) e representados de forma binária (presença/ausência).

Esta abordagem permite captar padrões linguísticos curtos e recorrentes, associados a contextos emocionais.

- CBF_trig_st_norm.csv: reúne trigramas (sequências de três palavras consecutivas), também com stemming, mas aplicando normalização para reduzir o impacto da frequência bruta. Estas combinações oferecem maior sensibilidade ao contexto e ao estilo de escrita nas letras.

A seleção destas três configurações visou garantir diversidade de representação lexical, permitindo explorar diferentes níveis de granularidade. Enquanto os unigramas fornecem uma visão ampla do vocabulário, bigramas e trigramas capturam padrões de coocorrência e nuances semânticas que emergem apenas em contextos mais longos.

5.1.2 Stylistic-Based Features (StyBF)

As features estilísticas foram extraídas a partir de métricas que captam a forma e estrutura das letras, independentemente do seu conteúdo semântico. Para este fim, foram utilizadas duas fontes principais:

CapitalLetters_M45.csv: contém a proporção de palavras em maiúsculas em cada letra. Esta métrica está relacionada com a expressividade textual, podendo indicar gritos, intensidade ou ênfase emocional.

Stylistic.csv: reúne múltiplos atributos estilísticos, incluindo:

- número total de palavras,
- comprimento médio das frases (em tokens),
- número médio de caracteres por palavra,
- frequência de sinais de pontuação (ex.: pontos de exclamação e interrogação, associados a intensidade emocional),
- diversidade lexical (type-token ratio), que mede a variedade de vocabulário.

Estas características permitem capturar marcas estilísticas do texto, úteis na identificação de emoções latentes, uma vez que a forma de escrever pode refletir estados afetivos, como agitação (uso repetido de “!”) ou serenidade (frases longas e estruturadas).

5.1.3 Semantic-Based Features (SemBF)

As features semânticas foram extraídas com base em dicionários e recursos linguísticos que associam palavras a dimensões emocionais ou psicológicas. Foram considerados os seguintes conjuntos:

- DAL ANEW: integra os recursos ANEW (Bradley & Lang, 1999) e DAL (Whissell, 2009), atribuindo valores de valência, ativação e dominância a palavras, permitindo mapear emoções em dimensões psicológicas.
- *General Inquire*: categorias semânticas do General Inquirer, que organiza palavras em 1180 dimensões (positivo/negativo, ativo/passivo, forte/fraco, etc.).
- *Gazeteers*: listas de palavras construídas com base no modelo de Russell (1980), associando termos específicos a cada quadrante emocional (alegria, tristeza, ansiedade, serenidade).
- *Synesketch*: anotações automáticas geradas pelo Synesketch, que mapeia palavras em 49 categorias emocionais pré-definidas, representando um espectro mais detalhado de estados afetivos.
- *Semantic*: consolida recursos linguísticos e semânticos adicionais, servindo como matriz geral de atributos emocionais derivados de dicionários.
- *Words Dictionary*: dicionário auxiliar de palavras, usado como suporte no mapeamento do vocabulário das letras para categorias emocionais.

Estas características permitem associar o vocabulário das letras não apenas a significados literais, mas também a valores afetivos e dimensões emocionais, criando um elo direto entre o texto e as emoções humanas.

5.2 Seleção de *Features*

Para a seleção de atributos recorreu-se ao *software* Weka (Hall et al., 2009), utilizando a métrica de avaliação ReliefF (I Kononenko, 1994), em combinação com o método de pesquisa Ranker. O algoritmo ReliefF avalia a relevância de cada atributo com base na sua capacidade de distinguir instâncias de diferentes classes, através da comparação entre vizinhos próximos da mesma e de diferentes classes. O método Ranker, por sua vez, ordena os atributos de acordo com o peso atribuído por ReliefF, produzindo assim um ranking de features em função da sua relevância individual para o target. Esta abordagem permitiu identificar, de forma sistemática, os atributos mais informativos do conjunto de dados.

Após o ranking inicial, foi conduzida uma análise exploratória para avaliar o impacto da redução progressiva do número de atributos no desempenho do classificador. Para tal, foram realizados testes em diferentes subconjuntos de features, aplicando uma estratégia de saltos no número de atributos avaliados.

Os resultados mostraram que, à medida que se reduzia a dimensionalidade (tabelas do capítulo 5.3), o desempenho do modelo melhorava gradualmente até estabilizar em torno de

algumas centenas de atributos. Esta tendência revelou que muitos dos atributos originais eram redundantes ou pouco relevantes, e que a utilização de um subconjunto otimizado permitia manter, ou mesmo aumentar, a eficácia dos classificadores.

Com base nesta análise, foi adotada uma estratégia de remoção iterativa de atributos, tendo como referência a métrica de *F-Measure* (ou *F1-score*) obtida pelo classificador SMO (*Sequential Minimal Optimization*, implementação de SVM no Weka). A seleção final correspondeu ao subconjunto que proporcionou o melhor equilíbrio entre precisão e recall.

Este conjunto otimizado foi então utilizado nas etapas subsequentes de treino e teste dos modelos de machine learning, servindo como base para as experiências comparativas apresentadas posteriormente.

Embora o processo de seleção final de atributos tenha sido conduzido com o classificador SMO (*Sequential Minimal Optimization*) que corresponde à implementação de máquinas de suporte de vetores (SVM) no Weka, foram realizadas verificações pontuais com outros algoritmos, como *Random Forest*, de forma a confirmar a robustez da seleção. Esses testes mostraram que os subconjuntos de atributos mais relevantes eram, na maioria dos casos, coincidentes (com pequenas variações de 1–2 atributos), o que reforça a estabilidade do processo de seleção. No entanto, dado que as diferenças observadas foram mínimas, optou-se por documentar apenas a configuração baseada no SMO, utilizada como modelo de referência ao longo do trabalho.

5.3 Resultados da seleção

A Tabela 2 apresenta os resultados do processo de seleção de atributos aplicados às letras originais. Tal como nas restantes experiências, foi utilizado o classificador SMO (*Sequential Minimal Optimization*) para avaliar subconjuntos de *features* com diferentes dimensões, aplicando saltos progressivos na redução da dimensionalidade.

Os resultados indicam que, com um número elevado de atributos (acima de 40.000), o desempenho do modelo é relativamente baixo (*F-Measure* ≈ 0.50). À medida que o número de atributos vai sendo reduzido, o valor da *F-Measure* aumenta de forma gradual e consistente, estabilizando a partir de aproximadamente 500 atributos. O melhor desempenho foi obtido com 353 features, alcançando um valor máximo de *F-Measure* = 0.694.

Este resultado demonstra que a eliminação de atributos redundantes contribuiu para melhorar o desempenho, permitindo reduzir drasticamente a dimensionalidade inicial (mais de 47.000 atributos) sem perda de eficácia. Pelo contrário, a simplificação do espaço de atributos revelou-se benéfica para o classificador.

Tabela 2 Features Originais

Número de Features	F-Measure		Número de Features	F-Measure
47186	0.506		41000	0.581
40000	0.578		43000	0.605
35000	0.5		44000	0.56
30000	0.403		550	0.667
25000	0.46		500	0.694
20000	0.485		450	0.694
15000	0.568		400	0.694
10000	0.617		380	0.694
8000	0.636		370	0.694
7000	0.661		360	0.694
6000	0.674		355	0.694
5000	0.661		353**	0.694
4000	0.64		352	0.684
3000	0.655		351	0.678
2000	0.654		350	0.69

A Tabela 3 apresenta os resultados obtidos para o processo de seleção de atributos aplicado às sumarizações abstrativas geradas pelo modelo DistilBART. Tal como na letra original, foi utilizado o classificador SMO (*Sequential Minimal Optimization*) para avaliar diferentes subconjuntos de *features*, recorrendo a saltos progressivos na redução da dimensionalidade.

Os resultados mostram que, com um número elevado de atributos (mais de 10.000), o desempenho é relativamente baixo (*F-Measure* ≈ 0.49). No entanto, à medida que o número de atributos é reduzido, o valor da *F-Measure* aumenta gradualmente, alcançando o seu ponto máximo em torno de 976 features (*F-Measure* = 0.607). Este resultado indica que a remoção de atributos redundantes e pouco informativos foi essencial para melhorar a eficácia da classificação. Embora a performance global seja inferior à obtida com as letras originais, a tendência de melhoria com a redução da dimensionalidade mantém-se, confirmando que a seleção criteriosa de atributos desempenha um papel fundamental no processamento de letras sumarizadas.

Tabela 3: Features abstractive_distil_bart

Features	Valor SMO (f-Measure)		Features	Valor SMO (f-Measure)
11470	0.49		976**	0.607
10000	0.482		975	0.597
8000	0.534		974	0.597
6000	0.538		970	0.581
5000	0.536		950	0.581
4000	0.525		800	0.558
3000	0.521		710	0.578
2000	0.555		700	0.578
1100	0.563		690	0.578
1000	0.591		600	0.553
990	0.591		500	0.558
980	0.591		250	0.547
977	0.602		100	0.565

A Tabela 4 apresenta os resultados da seleção de atributos aplicada às sumarizações produzidas pelo modelo LED (*Longformer Encoder-Decoder*). Os valores de *F-Measure* obtidos pelo classificador SMO mostram que o desempenho inicial, com mais de 15.000 atributos, é baixo (≈ 0.41), mas melhora gradualmente com a redução da dimensionalidade. O melhor valor surge em torno de 191 atributos (*F-Measure* = 0.537). Este comportamento reforça que a simplificação do espaço de atributos foi fundamental para extrair informação útil das sumarizações LED, embora o desempenho global se mantenha inferior ao das letras originais.

Tabela 4: Features abstractive_led

Número de Features	F-Measure		Número de Features	F-Measure
15256	0.411		210	0.53
12000	0.294		200	0.53
10000	0.255		192	0.53
8000	0.302		191**	0.537
6000	0.316		190	0.532
4000	0.375		180	0.506
2000	0.389		100	0.48
1000	0.388		55	0.498
800	0.4		54	0.499
700	0.414		53	0.493
600	0.414		50	0.492
500	0.413		30	0.432
300	0.52			

A Tabela 5 mostra os resultados para as sumarizações geradas pelo modelo PEGASUS (CNN/DailyMail). Observa-se que, com muitos atributos (acima de 10.000), os valores de *F-Measure* são baixos (≈ 0.43–0.47). A performance melhora progressivamente com a redução, estabilizando em torno dos 167 atributos (*F-Measure* = 0.537). Apesar de não atingir a eficácia das letras originais, a tendência confirma que a seleção de atributos é essencial para extrair padrões relevantes nas sumarizações produzidas pelo PEGASUS.

Tabela 5: Feature abstractive_pegasus_dailymail

Número de Features	F-Measure		Número de Features	F-Measure
13188	0.437		210	0.509
12000	0.462		200	0.525
10000	0.47		190	0.525
8000	0.428		180	0.525
6000	0.477		172	0.526
4000	0.498		170	0.537
2000	0.498		168	0.537
1000	0.438		167**	0.537
800	0.464		166	0.521
600	0.434		165	0.51
500	0.433		160	0.522
300	0.471		100	0.454

A Tabela 6 apresenta os resultados do modelo T5, aplicados às letras sumarizadas. O desempenho inicial é fraco com mais de 10.000 atributos (*F-Measure* = 0.40), mas melhora com a redução. O melhor resultado foi obtido com cerca de 400 atributos (*F-Measure* = 0.557). Este valor é competitivo em relação aos demais modelos abstrativos, mostrando que o T5 conseguiu capturar padrões mais consistentes após a seleção de atributos.

Tabela 6: Feature abstractive_t5

Número de Features	F-Measure		Número de Features	F-Measure
16982	0.45		410	0.557
12000	0.376		405	0.557
10000	0.404		400**	0.557
8000	0.428		399	0.538
6000	0.475		398	0.526
4000	0.486		395	0.533
2000	0.516		390	0.527
1000	0.531		300	0.529
800	0.542		200	0.484
600	0.534		100	0.506
500	0.539		80	0.506
460	0.521		25	0.459

A Tabela 7 mostra os resultados para o modelo DistilBART, evidenciando um comportamento semelhante ao de outros modelos abstrativos. Com um número elevado de atributos (≈ 11.000), o desempenho é reduzido (0.49), mas melhora progressivamente até estabilizar em torno de 1999 atributos (*F-Measure* = 0.612). Este valor é próximo ao das letras originais, sugerindo que o DistilBART preservou informação emocional relevante mesmo após a sumarização.

Tabela 7: Feature abstrativa_distilbart

Número de Features	F-Measure		Número de Features	F-Measure
11506	0.497		1800	0.603
10000	0.483		1500	0.601
8000	0.529		1000	0.551
6000	0.535		800	0.571
4000	0.538		600	0.558
2500	0.511		500	0.558
2000	0.612		400	0.525
1999**	0.612		300	0.549
1990	0.584		200	0.551
1980	0.599		100	0.543
1950	0.599		80	0.478
			25	0.511

A Tabela 8 apresenta os resultados obtidos a partir do modelo BART. O desempenho mantém-se estável na faixa de 0.52–0.58 de *F-Measure* à medida que o número de atributos é reduzido. O melhor resultado foi alcançado com 1959 atributos (*F-Measure* = 0.583). Embora inferior às letras originais, este valor mostra que o BART conseguiu manter alguma consistência nos padrões emocionais após a sumarização.

Tabela 8: Features F_abstractive_bart

Número de Features	F-Measure		Número de Features	F-Measure
12567	0.526		1958	0.577
10000	0.523		1950	0.577
8000	0.516		1000	0.541
6000	0.549		800	0.562
4000	0.54		600	0.525
2000	0.583		500	0.521
1999	0.583		400	0.526
1990	0.583		300	0.53
1980	0.583		200	0.549
1970	0.583		100	0.535
1960	0.583			
1959**	0.583			

A Tabela 9 mostra os resultados para a abordagem extrativa LexRank. O LexRank obteve um desempenho onde os valores de F-Measure foram aumentando até atingir o máximo de 0.597 com apenas 925 atributos.

Tabela 9: Features lex_rank

Número de Features	F-Measure		Número de Features	F-Measure
24230	0.404		924	0.583
20000	0.503		920	0.587
16000	0.402		900	0.575
12000	0.454		800	0.567
10000	0.44		600	0.578
8000	0.472		500	0.579
6000	0.493		430	0.578
4000	0.554		410	0.582
2000	0.562		400	0.582
1010	0.594		390	0.582
1000	0.594		300	0.558
950	0.594		200	0.548
930	0.592		100	0.561
925**	0.597		50	0.569

A Tabela 10 apresenta os resultados da seleção de atributos com base na sumarização LSA (*Latent Semantic Analysis*). O desempenho inicial com muitos atributos é baixo (≈ 0.41), mas melhora de forma consistente até estabilizar em torno de 200 atributos (*F-Measure* = 0.569). Embora inferior às letras originais, este resultado mostra que o LSA, aliado à seleção de atributos, conseguiu captar parte da informação semântica relevante para a classificação emocional.

Tabela 10: Features lsa

Número de Features	F-Measure		Número de Features	F-Measure
24735	0.415		600	0.516
20000	0.446		500	0.512
16000	0.368		400	0.526
12000	0.425		300	0.507
10000	0.432		205	0.562
8000	0.47		201	0.569
6000	0.489		200**	0.569
4000	0.513		198	0.552
2000	0.514		180	0.523
1000	0.506		100	0.498
800	0.535			

A Tabela 11 apresenta os resultados para o método extrativo Luhn. O desempenho inicial é reduzido (≈ 0.35 com 14.000 atributos), mas melhora progressivamente até atingir o máximo de 0.556 com 2000 atributos. Tal como no LexRank, a redução da dimensionalidade foi decisiva para melhorar a performance, confirmando que métodos extrativos conseguem preservar conteúdo informativo útil para o reconhecimento emocional.

Tabela 11: Features luhn

Número de Features	F-Measure		Número de Features	F-Measure
21404	0.46		1999	0.553
18000	0.437		1990	0.553
14000	0.355		1950	0.548
12000	0.366		1000	0.541
10000	0.435		800	0.536
8000	0.423		600	0.502
6000	0.458		500	0.501
4000	0.49		400	0.478
3000	0.516		300	0.487
2050	0.553		200	0.501
2001	0.548		100	0.5
2000**	0.556		50	0.499

A Tabela 12 mostra os resultados do método TextRank. Observa-se que, embora os valores iniciais sejam baixos (≈ 0.34–0.41), o desempenho aumenta significativamente com a redução de atributos, alcançando o máximo de 0.629 com 369 atributos. Este é um dos melhores resultados entre todas as abordagens, demonstrando a eficácia do TextRank como técnica extrativa aliada à seleção de atributos.

Tabela 12: Features text_rank

Número de Features	F-Measure		Número de Features	F-Measure
23915	0.441		500	0.586
20000	0.426		400	0.598
16000	0.341		398	0.598
12000	0.396		390	0.62
10000	0.412		380	0.625
8000	0.404		370	0.629
6000	0.436		369**	0.629
4000	0.517		368	0.623
2000	0.491		360	0.622
1000	0.524		300	0.558
800	0.563		100	0.587
600	0.58		50	0.565

A Tabela 13 apresenta os resultados da seleção de atributos aplicada às representações geradas pelo modelo BERT. Diferente de alguns modelos abstrativos, o BERT obteve já de início um desempenho elevado (≈ 0.60 de *F-Measure* com mais de 17.000 atributos). À medida que a dimensionalidade foi reduzida, os valores mantiveram-se estáveis, atingindo o melhor resultado em torno de 783 atributos, com *F-Measure* = 0.616.

Estes resultados confirmam que o BERT, enquanto modelo pré-treinado com forte capacidade de representação semântica, conseguiu preservar informação relevante mesmo após a compressão do espaço de atributos. Comparativamente a outros modelos testados, o BERT destaca-se por apresentar um desempenho consistente, sem grandes perdas associadas à redução de dimensionalidade.

Tabela 13: Features BERT

Número de Features	F-Measure		Número de Features	F-Measure
17649	0.603		785	0.612
17600	0.598		783**	0.616
14000	0.566		782	0.605
12000	0.554		600	0.561
10000	0.542		500	0.595
8000	0.529		400	0.599
6000	0.543		390	0.599
4000	0.575		350	0.599
2000	0.599		320	0.571
1000	0.584		300	0.571
800	0.6		200	0.589
795	0.6		100	0.559

A Tabela 14 apresenta os resultados obtidos com as sumarizações geradas pelo modelo GPT-2. O desempenho inicial, com mais de 20.000 atributos, é baixo (≈ 0.30–0.37), mas melhora progressivamente com a redução da dimensionalidade. O melhor resultado foi alcançado com 299 atributos, atingindo *F-Measure* = 0.518. Embora inferior a outros modelos abstrativos como DistilBART ou T5, este valor mostra que o GPT-2 manteve alguma capacidade de representar informação emocional após a sumarização.

Tabela 14: Features GPT2

Número de Features	F-Measure		Número de Features	F-Measure
25236	0.478		600	0.457
22000	0.38		500	0.474
18000	0.308		400	0.492
14000	0.317		305	0.506
12000	0.366		301	0.508
10000	0.37		300	0.518
8000	0.362		299**	0.518
6000	0.433		298	0.513
4000	0.443		297	0.513
2000	0.487		295	0.507
1000	0.442		200	0.501
800	0.474		100	0.478

A Tabela 15 apresenta um resumo consolidado dos resultados obtidos no processo de seleção de atributos para todas as abordagens testadas. Nela estão reunidas a quantidade inicial e final de *features*, bem como os valores correspondentes de *F-Measure* antes e depois da seleção.

Observa-se que, em todos os casos, a redução da dimensionalidade resultou em melhorias de desempenho, confirmando a relevância da seleção de atributos para eliminar redundâncias e reforçar a eficácia dos classificadores. Entre os resultados, destacam-se:

- Letras Originais: obtiveram o melhor desempenho global, com 353 atributos e *F-Measure* de 0.694, mostrando que a informação completa da letra é a representação mais rica.
- Métodos Extrativos (LexRank, Luhn, TextRank, LSA): apresentaram resultados consistentes, sendo o TextRank o mais eficaz, alcançando 0.629 de F-Measure com apenas 369 atributos.
- Modelos Abstrativos (DistilBART, PEGASUS, LED, T5, BART): registaram valores mais baixos, com F-Measure entre 0.537 e 0.612, o que demonstra perda de informação relevante durante a sumarização. Ainda assim, o DistilBART destacou-se com 1999 atributos e 0.612 de F-Measure, aproximando-se do desempenho dos originais.
- Modelos baseados em LLMs (BERT e GPT-2): apresentaram comportamentos distintos; o BERT manteve uma performance estável e competitiva (0.616), enquanto o GPT-2 apresentou resultados inferiores (0.518), sugerindo menor robustez na preservação de informação emocional.

Os resultados confirmam que, embora as sumarizações abstrativas reduzam a dimensão textual, há uma perda de informação emocional relevante. Já as abordagens extrativas, em particular o TextRank, mostraram-se mais eficazes, conseguindo reduzir drasticamente o número de atributos e ainda assim manter bons níveis de desempenho.

Tabela 15: Resultados resumidos da seleção de features

Tipo Features:	Quantidade features Inicial	F-Measure	Quantidade features Final	F-Measure
Features_Original_letra	47187	0.506	353	0.694
abstractive_distil_bart	11470	0.49	976	0.607
abstractive_led	15256	0.411	191	0.537
abstractive_pegasus_dailymail	13188	0.437	167	0.537
abstractive_t5	16982	0.45	400	0.557
abstrativa_distilbart	11506	0.497	1999	0.612
Features_abstractive_bart	12567	0.526	1959	0.583
lex_rank	24230	0.404	925	0.597
lsa	24735	0.415	200	0.569
luhn	21404	0.46	2000	0.556
text_rank	23915	0.441	369	0.629
z.BERT	17649	0.603	783	0.616
z.GPT2	25236	0.478	299	0.518

5.4 Classificação

Nesta fase do trabalho, o objetivo foi aplicar técnicas de classificação para prever [emoções/quadrantes] a partir das letras musicais previamente processadas. A escolha de diferentes algoritmos permitiu avaliar qual o modelo mais adequado para lidar com a complexidade e variabilidade dos dados, assegurando não apenas um bom desempenho, mas também uma maior capacidade de generalização.

Para garantir a robustez dos resultados, foi adotada uma estratégia de validação cruzada (*Cross-Validation*) com 10 folds, prática comum que permite avaliar a estabilidade do desempenho dos modelos ao longo de diferentes divisões dos dados.

No pré-processamento, para lidar com valores em falta (NaN) nas variáveis numéricas, optou-se por uma substituição simples por zero. Esta abordagem evita a exclusão de amostras e mantém a integridade do conjunto de treino, embora se reconheça que possa introduzir viés, sendo necessário avaliar o seu impacto nos resultados.

Além disso, recorreu-se ao método de g*rid search* para a otimização de hiperparâmetros, avaliando sistematicamente várias combinações de parâmetros com base em métricas como acurácia e F1-score.

Após a validação cruzada e a afinação dos parâmetros, os modelos foram treinados novamente com todos os dados disponíveis e avaliados na sua configuração final.

Foram considerados e comparados os seguintes algoritmos e abordagens:

- Support Vector Machine (SVM)
- Random Forest
- Multilayer Perceptron (MLP)
- H2O MLP (framework H2O.ai)
- BERT (modelo de linguagem baseado em transformadores)
- GPT-2 (modelo autoregressivo pré-treinado para tarefas de NLP)

Todos os modelos foram implementados e testados em Jupyter Lab, com análise e registo sistemático de cada execução, permitindo uma avaliação detalhada de cada configuração e respetivo desempenho.

5.4.1 Métricas para Comparação de Modelos de Classificação:

A comparação entre diferentes modelos de classificação é uma etapa essencial no desenvolvimento de sistemas de aprendizagem automática. Esta análise permite selecionar o modelo mais eficaz para um determinado problema, com base em critérios quantitativos e qualitativos (Raschka & Mirjalili, 2019).

Para a análise comparativa, foram calculadas várias métricas de desempenho, de forma a obter uma avaliação abrangente da eficácia de cada modelo. Embora a acurácia tenha sido incluída como métrica complementar, não foi considerada isoladamente como critério principal, uma vez que pode ser enganadora em diferentes contextos. Ao avaliar deu-se maior relevância ao *F1-score*. Para além disso, foi analisadas separadamente a precisão, de modo a identificar a capacidade dos modelos em evitar falsos positivos e em reconhecer corretamente os verdadeiros positivos.

Complementarmente, foi analisada a AUC (*Area Under the Curve*), que fornece uma medida da capacidade discriminativa do modelo, independente do limiar de decisão, sendo especialmente útil para compreender o trade-off entre verdadeiros positivos e falsos positivos. A matriz de confusão foi ainda registada, permitindo observar diretamente a distribuição de acertos e erros por classe.

O valor da coluna matriz utilizada nas próximas tabelas é o número total de classificações corretas obtidas por cada modelo em relação ao conjunto de 180 letras musicais. Este valor resulta da soma dos acertos na matriz de confusão de cada execução, funcionando como um indicador adicional da performance geral. Embora não forneça o detalhe por classe, esta métrica permite uma perceção imediata do número de previsões bem-sucedidas face ao total, complementando as restantes métricas (*Accuracy, F1-score, Precision* e AUC).

Por fim, foi considerado também o tempo médio de execução, uma vez que, em cenários práticos, a eficiência computacional pode ser um fator determinante na escolha do modelo. Esta métrica garante uma perspetiva mais realista sobre o custo de utilização dos algoritmos em sistemas de classificação aplicados a grandes volumes de dados.

Assim, a avaliação baseou-se num conjunto diversificado de métricas, assegurando uma comparação justa e completa entre os diferentes modelos testados.

5.5 Resultados da Classificação

Nesta secção são apresentados os resultados obtidos com os diferentes algoritmos de classificação aplicados às letras musicais, tanto na versão original como nas versões sumarizadas. O objetivo é avaliar de forma sistemática o desempenho de cada modelo, destacando os pontos fortes e limitações de cada abordagem.

5.5.1 SVM

O classificador SVM (*Support Vector Machine*) foi avaliado com validação cruzada de 10 *folds*, após afinação dos hiperparâmetros através de *grid search*. A Tabela 16 resume os resultados obtidos para as diferentes representações das letras (originais e sumarizadas), considerando as principais métricas de avaliação: *accuracy*, *F1-score*, *precision*, AUC, matriz de confusão e tempo médio de execução.

Entre as técnicas de sumarização, destacam-se os métodos extrativos, em particular o TextRank (*F1-score* = 0.5775, AUC = 0.85) e o Luhn (*F1-score* = 0.5875), que preservaram melhor a informação relevante para a classificação. Já os métodos abstrativos apresentaram desempenhos mais modestos, situando-se entre 0.53 e 0.61 de AUC, refletindo a perda de informação durante a reescrita das letras.

Os modelos baseados em LLMs tiveram desempenhos distintos: o BERT alcançou resultados competitivos (*F1-score* = 0.65, AUC = 0.855), próximos das letras originais, enquanto o GPT-2 registou um desempenho inferior (*F1-score* = 0.5775, AUC = 0.7525).

Para além dos resultados de desempenho, foi registada também a configuração final de hiperparâmetros obtida pelo processo de *grid search* para cada representação das letras. A Tabela 17 apresenta os valores de C, gamma e o tipo de kernel selecionados pelo SVM em cada cenário.

Verifica-se que o kernel RBF foi, na maioria dos casos, a melhor escolha, refletindo a sua capacidade de modelar fronteiras de decisão não lineares, adequadas à complexidade dos dados. Em contrapartida, em alguns casos específicos (por exemplo, abstractive_bart e luhn), o kernel linear foi suficiente, indicando que os atributos selecionados apresentavam uma separabilidade mais simples.

Os valores do parâmetro C oscilaram entre 0.1, 1 e 10, ajustando o grau de regularização em função da representação textual. Da mesma forma, o parâmetro gamma variou entre auto e scale, influenciando a flexibilidade do modelo. Esta diversidade confirma que a otimização de hiperparâmetros desempenhou um papel central para maximizar o desempenho do SVM em cada tipo de sumarização.

Tabela 16: Classificação SVM

Tipo Sumarização:	Accuracy	F1-score	Precision	AUC	Matriz	Tempo
Features_Original_letra	0,72	0,7125	0,7175	0,915	127/180	3s
abstractive_distil_bart	0,58	0,575	0,575	0,8225	104/180	3s
abstractive_led	0,56	0,5425	0,5925	0,7925	101/180	4s
abstractive_pegasus_dailymail	0,52	0,53	0,535	0,7525	97/180	6s
abstractive_t5	0,57	0,5675	0,5725	0,82	102/180	5s
abstrativa_distilbart	0,56	0,5575	0,57	0,8175	101/180	3s
Features_abstractive_bart	0,56	0,555	0,56	0,8125	101/180	3s
lex_rank	0,57	0,565	0,57	0,8225	102/180	4s
lsa	0,56	0,5625	0,6125	0,8075	100/180	2s
luhn	0,59	0,5875	0,6025	0,8075	107/180	3s
text_rank	0,58	0,5775	0,5825	0,85	104/180	4s
z.BERT	0,65	0,65	0,6725	0,855	117/180	5s
z.GPT2	0,56	0,5775	0,5775	0,7525	100/180	3s

Tabela 17: Parâmetros do SVM

Tipo Sumarização:	C	Gamma	Kernel
Features_Original_letra	10	Scale	rbf
abstractive_distil_bart	10	Scale	rbf
abstractive_led	1	Auto	rbf
abstractive_pegasus_dailymail	1	Scale	rbf
abstractive_t5	10	Auto	rbf
abstrativa_distilbart	0.1	Scale	Linear
Features_abstractive_bart	0.1	Scale	Linear
lex_rank	10	Scale	rbf
lsa	1	Auto	rbf
luhn	0.1	Scale	Linear
text_rank	10	Auto	rbf
z.BERT	1	Scale	rbf
z.GPT2	10	Scale	rbf

5.5.2 Random Forest

O classificador Random Forest também foi avaliado com validação cruzada de 10 *folds*, tendo os seus hiperparâmetros otimizados através de *grid search*. A Tabela 18 resume os resultados de desempenho obtidos para cada tipo de representação das letras.

De forma geral, o desempenho do *random forest* foi inferior ao do SVM, com valores de *F1-score* a variar entre 0.46 e 0.65. O melhor resultado foi alcançado com as letras originais (F1-score = 0.645, AUC = 0.83), confirmando novamente que os textos completos preservam mais informação relevante para a tarefa de classificação emocional.

Entre as técnicas de sumarização, destacaram-se o TextRank (F1-score = 0.6175, AUC = 0.8175) e o T5 (F1-score = 0.59, AUC = 0.8075), que superaram outros métodos, ainda que sem atingir os valores das letras originais. Por outro lado, modelos como o PEGASUS apresentaram os resultados mais baixos (*F1-score* = 0.4625, AUC = 0.7125), evidenciando dificuldades em manter informação discriminativa após a sumarização.

Nos modelos baseados em LLMs, o BERT voltou a demonstrar resultados competitivos (*F1-score* = 0.59, AUC = 0.81), enquanto o GPT-2 registou desempenhos mais modestos (*F1-score* = 0.51, AUC = 0.715).

A Tabela 19 apresenta os parâmetros finais selecionados no processo de otimização.

Verifica-se que, na maioria dos casos, a profundidade máxima das árvores (max_depth) não foi restringida, permitindo ao modelo explorar toda a complexidade dos dados. O parâmetro max_features variou entre sqrt e log2, adaptando-se ao equilíbrio entre diversidade e robustez das árvores. Já os parâmetros min_samples_leaf e min_samples_split oscilaram entre valores reduzidos (1–2), refletindo a tentativa do modelo de ajustar-se a conjuntos relativamente pequenos de atributos.

O número de estimadores (n_estimators) variou entre 100 e 300, dependendo do tipo de sumarização, influenciando diretamente o tempo de execução, que foi em média superior ao observado no SVM (cerca de 70–100 segundos).

Em síntese, embora o *random forest* apresente boa capacidade de generalização, o seu desempenho foi inferior ao do SVM neste problema específico. No entanto, demonstrou-se particularmente eficaz em representações sumarizadas como TextRank e T5, evidenciando que pode ser competitivo quando aplicado a representações mais compactas.

Tabela 18: Classificação Random Forest

Tipo Features:	Accuracy	F1-score	Precision	AUC	Matriz	Tempo
Features_Original_letra	0,65	0,645	0,66	0,83	117/180	75s
abstractive_distil_bart	0,56	0,5425	0,5775	0,76	100/180	84s
abstractive_led	0,53	0,5425	0,53	0,7525	95/180	86s
abstractive_pegasus_dailymail	0,47	0,4625	0,4825	0,7125	84/180	74s
abstractive_t5	0,60	0,59	0,6075	0,8075	108/180	90s
abstrativa_distilbart	0,58	0,58	0,5975	0,7875	105/180	70s
Features_abstractive_bart	0,59	0,5775	0,59	0,7825	106/180	73s
lex_rank	0,53	0,535	0,575	0,7825	96/180	101s
lsa	0,53	0,535	0,545	0,75	95/180	92s
luhn	0,53	0,5275	0,545	0,745	95/180	86s
text_rank	0,62	0,6175	0,625	0,8175	111/180	73s
z.BERT	0.59	0,59	0,615	0,81	107/180	79s
z.GPT2	0,52	0,515	0,53	0,715	93/180	81s

Tabela 19: Parâmetros do Random Forest

Tipo Sumarização:	max_depth	max_features	min_samples_leaf	min_samples_split	n_estimators
Features_Original_letra	None	sqrt	1	5	100
abstractive_distil_bart	10	Log2	2	10	100
abstractive_led	None	Log2	2	10	100
abstractive_pegasus_dailymail	10	sqrt	2	5	300
abstractive_t5	None	Log2	1	5	300
abstrativa_distilbart	None	sqrt	1	2	300
Features_abstractive_bart	10	sqrt	2	2	100
lex_rank	None	sqrt	2	2	300
lsa	None	sqrt	1	2	100
luhn	None	sqrt	1	10	100
text_rank	10	sqrt	1	10	300
z.BERT	10	Log2	1	2	100
z.GPT2	None	sqrt	2	5	300

5.5.3 Neuronal Multilayer Perceptron

A rede neuronal *multilayer perceptron* (MLP) foi avaliada seguindo o mesmo padrão de validação cruzada de 10 folds, após afinação de hiperparâmetros através de *grid search*. A Tabela 20 apresenta os resultados de desempenho obtidos para cada tipo de representação textual.

Os resultados evidenciam que a MLP alcançou o melhor desempenho global entre os modelos tradicionais, superando tanto o SVM como o *random forest* em várias métricas. Nas letras originais, a rede obteve o valor mais elevado de *F1-score* (0.7625) e de AUC (0.9225), confirmando a capacidade das redes neuronais de capturar padrões complexos nos dados.

Entre as técnicas de sumarização, o destaque vai para o TextRank (*F1-score* = 0.6925, AUC = 0.88) e para o BERT (*F1-score* = 0.70, AUC = 0.8775), que se aproximaram bastante dos resultados das letras originais. Estes valores sugerem que tanto os métodos extrativos quanto os modelos de linguagem baseados em transformadores preservaram de forma eficaz a informação relevante para a classificação emocional.

Por outro lado, métodos abstrativos como PEGASUS e LED registaram desempenhos mais baixos (*F1-score* ≈ 0.51–0.52), revelando limitações na manutenção de características discriminativas após a compressão textual.

A Tabela 21 apresenta a configuração final dos hiperparâmetros da MLP.

Verifica-se que a função de ativação oscilou entre Tanh e Relu, adaptando-se ao tipo de representação. A dimensão das camadas escondidas variou entre 50 e 100 neurónios, com arquiteturas simples (1–2 camadas), demonstrando que redes relativamente compactas foram suficientes para capturar padrões relevantes. O parâmetro de regularização (alpha) foi mantido baixo (0.0001), enquanto a taxa de aprendizagem oscilou entre 0.001 e 0.01. O otimizador Adam foi o mais frequente, embora em alguns casos tenha sido utilizado o SGD, refletindo ajustes específicos de convergência.

O tempo médio de execução manteve-se estável (≈ 36–42 segundos), mostrando-se eficiente em comparação com o Random Forest e competitivo face ao SVM.

Os resultados confirmam a robustez da MLP neste contexto, destacando-se como o classificador com melhor desempenho global, especialmente quando aplicado às letras originais e a representações sumarizadas por métodos extrativos e modelos pré-treinados como o BERT.

Tabela 20: Classificação Rede Neuronal MLP

Tipo Features:	Accuracy	F1-score	Precision	AUC	Matriz	Tempo
Features_Original_letra	0,77	0,7625	0,7675	0,9225	135/180	37s
abstractive_distil_bart	0,64	0,645	0,645	0,825	116/180	38s
abstractive_led	0,51	0,5125	0,5175	0,755	92/180	38s
abstractive_pegasus_dailymail	0,52	0,51	0,51	0,7275	93/180	40s
abstractive_t5	0,61	0,6125	0,6125	0,8	110/180	36s
abstrativa_distilbart	0,58	0,5725	0,585	0,8125	105/180	37s
Features_abstractive_bart	0,61	0,5975	0,6	0,81	109/180	37s
lex_rank	0,60	0,6025	0,605	0,83	108/180	42s
lsa	0,57	0,58	0,5875	0,79	103/180	36s
luhn	0,58	0,5775	0,58	0,7975	105/180	37s
text_rank	0,69	0,6925	0,7025	0,88	124/180	40s
z.BERT	0,70	0,7	0,7	0,8775	126/180	38s
z.GPT2	0,58	0,5825	0,5825	0,7875	105/180	37s

Tabela 21: Parâmetros da Rede Neuronal MLP

Tipo Sumarização:	activation	alpha	hidden_layer_sizes	learning_rate_init	solver
Features_Original_letra	Tanh	0.0001	100,0	0.01	Adam
abstractive_distil_bart	Tanh	0.0001	50,0	0.01	Adam
abstractive_led	Relu	0.001	100,0	0.01	Adam
abstractive_pegasus_dailymail	Relu	0.0001	50,0	0.001	Sgd
abstractive_t5	Relu	0.0001	50,0	0.01	Adam
abstrativa_distilbart	Tanh	0.0001	50,50	0.01	Adam
Features_abstractive_bart	Tanh	0.0001	50,50	0.01	Adam
lex_rank	Relu	0.0001	50,50	0.01	Adam
lsa	Relu	0.0001	50,0	0.001	Adam
luhn	Tanh	0.0001	50,0	0.01	Sgd
text_rank	Relu	0.0001	50,0	0.01	Adam
z.BERT	Relu	0.0001	50,50	0.01	Adam
z.GPT2	Tanh	0.0001	50,0	0.001	Sgd

5.5.4 Neuronal na framework H2O.ai

A rede neuronal implementada na framework H2O.ai foi avaliada com validação cruzada de 10 folds. Diferentemente das outras abordagens, nesta implementação os parâmetros de rede são pré-definidos pelo sistema, pelo que não foi necessário realizar *grid search* para otimização.

Na Tabela 22 os resultados mostram que o modelo obteve desempenhos competitivos, embora de forma geral inferiores aos da MLP tradicional apresentada na secção anterior. Nas letras originais, foi atingido um *F1-score* de 0.725 e uma AUC de 0.9075, valores que, apesar de sólidos, ficaram abaixo da rede neuronal configurada manualmente, que ultrapassou os 0.76 de F1-score.

Entre as representações sumarizadas, destacaram-se novamente o TextRank (*F1-score* = 0.6875, AUC = 0.8765) e o BERT (*F1-score* = 0.7025, AUC = 0.8725), confirmando a robustez destes métodos na preservação de informação discriminativa. Em contraste, modelos abstrativos como PEGASUS e LED registaram desempenhos mais modestos (F1-score ≈ 0.52–0.58), alinhados com as limitações observadas em classificadores anteriores.

Um aspeto relevante desta implementação foi o tempo de execução, significativamente mais elevado em comparação com as outras abordagens. Os tempos médios variaram entre 450 e 512 segundos, tornando este modelo menos eficiente em cenários práticos que exijam rapidez de treino ou reclassificação.

Em síntese, a rede neuronal em H2O.ai demonstrou desempenho estável e competitivo, sobretudo nas letras originais e em representações sumarizadas por métodos extrativos ou modelos baseados em transformadores. Contudo, o custo computacional elevado limita a sua aplicabilidade prática quando comparada com a MLP tradicional ou mesmo com o SVM.

Tabela 22: Classificação Rede Neuronal em H20.AI

Tipo Features:	Accuracy	F1-score	Precision	AUC	Matriz	Tempo
Features_Original_letra	0,73	0,725	0,73	0,9075	128/180	451s
abstractive_distil_bart	0,62	0,6275	0,6375	0,805	112/180	492s
abstractive_led	0,58	0,5775	0,5925	0,8025	104/180	472s
abstractive_pegasus_dailymail	0,53	0,5225	0,5275	0,7575	96/180	483s
abstractive_t5	0,62	0,615	0,625	0,8425	111/180	476s
abstrativa_distilbart	0,63	0,6275	0,6375	0,8275	114/180	459s
Features_abstractive_bart	0,58	0,57	0,5875	0,8275	104/180	461s
lex_rank	0,61	0,61	0,615	0,835	109/180	512s
lsa	0,53	0,535	0,5425	0,785	96/180	492s
luhn	0,56	0,56	0,5675	0,7825	101/180	481s
text_rank	0,68	0,685	0,6975	0,875	123/180	456s
z.BERT	0,65	0,65	0,6525	0,8725	117/180	450s
z.GPT2	0,57	0,5725	0,5825	0,79	103/180	446s

5.5.5 BERT

O modelo BERT foi submetido a um processo de *fine-tuning* gradual. Inicialmente, apenas a camada de classificação adicionada foi treinada, mantendo os pesos do modelo base congelados. A partir de determinada época, as camadas do encoder foram desbloqueadas, permitindo a atualização progressiva dos parâmetros do modelo pré-treinado.

A Tabela 23 apresenta os resultados obtidos, incluindo as métricas principais e o tempo médio de execução.

Os resultados demonstram que o BERT alcançou desempenhos competitivos, sobretudo em termos de AUC, onde se destacou com valores elevados em quase todas as representações (entre 0.84 e 0.93). Nas letras originais, o modelo atingiu um F1-score de 0.675 e uma AUC de 0.927, confirmando a sua forte capacidade discriminativa e robustez na classificação.

Entre as técnicas de sumarização, destacaram-se novamente os métodos extrativos (como LexRank, TextRank e LSA), que mantiveram *F1-scores* entre 0.585 e 0.66 e AUC consistentemente acima de 0.90. Estes resultados mostram que o BERT conseguiu extrair informação relevante mesmo de representações mais condensadas, beneficiando da sua arquitetura contextual bidirecional.

Os métodos abstrativos (DistilBART, T5, BART e PEGASUS) apresentaram desempenhos ligeiramente inferiores, com *F1-scores* na faixa dos 0.52–0.61, mas ainda com AUC elevada (≈0.87–0.90), o que sugere que o modelo preserva a capacidade de distinguir classes mesmo quando a informação lexical é reduzida.

Um ponto a destacar é o tempo de execução, consideravelmente mais elevado do que os modelos tradicionais. Os tempos médios variaram entre 498 e 568 segundos, refletindo o elevado custo computacional associado ao fine-tuning de modelos de larga escala como o BERT.

Em síntese, o BERT apresentou-se como um dos modelos mais robustos em termos de AUC e consistência de resultados, mostrando desempenho sólido tanto nas letras originais quanto em representações sumarizadas. No entanto, o seu custo computacional torna-o menos eficiente em cenários que exigem rapidez ou recursos limitados, quando comparado a modelos mais leves como SVM ou MLP.

Tabela 23: Classificação LLM BERT

Tipo Features:	Accuracy	F1-score	Precision	AUC	Matriz	Tempo
Features_Original_letra	0,68	0,675	0,765	0,9270	123/180	551s
abstractive_distil_bart	0,56	0,5325	0,6025	0,8422	100/180	503s
abstractive_led	0,42	0,32	0,68	0,8035	75/180	523s
abstractive_pegasus_dailymail	0,55	0,5175	0,7425	0,8912	99/180	535s
abstractive_t5	0,56	0,52	0,7175	0,8696	101/180	492s
abstrativa_distilbart	0,53	0,475	0,7075	0,8935	96/180	546s
Features_abstractive_bart	0,63	0,615	0,775	0,9026	114/180	503s
lex_rank	0,61	0,5875	0,74	0,9086	109/180	568s
lsa	0,58	0,565	0,7275	0,9068	105/180	563s
luhn	0,54	0,475	0,715	0,8734	98/180	549s
text_rank	0,67	0,66	0,7475	0,9053	121/180	546s
z.BERT	0,46	0,385	0,75	0,8652	83/180	498s
z.GPT2	0,67	0,67	0,735	0,8869	121/180	503s

5.5.6 GPT-2

No caso do GPT-2, optou-se por um processo de *fine-tuning* completo, em que todos os parâmetros do modelo pré-treinado foram atualizados durante o treino no corpus de letras musicais. Esta abordagem permitiu que o modelo ajustasse integralmente as suas representações internas ao domínio em estudo, explorando de forma mais profunda as especificidades linguísticas associadas à expressão de emoções em letras de música.

Na Tabela 24 os resultados mostram que o GPT-2 apresentou um desempenho consideravelmente inferior em comparação com todos os outros classificadores. Os valores de *F1-score* situaram-se entre 0.11 e 0.26, enquanto as AUCs oscilaram em torno de 0.46–0.55, revelando uma capacidade discriminativa muito limitada.

Mesmo nas letras originais, onde seria expectável um desempenho superior, o modelo alcançou apenas um *F1-score* de 0.11 e uma AUC de 0.4875, resultados que se aproximam de um comportamento aleatório. Entre as representações sumarizadas, o melhor desempenho ocorreu no método Luhn (F1-score = 0.26, AUC = 0.515), embora ainda bastante aquém dos restantes modelos.

Outro aspeto relevante é o tempo de execução, que foi elevado (entre 523 e 622 segundos), tornando o custo computacional desproporcional face ao baixo desempenho alcançado.

Uma possível explicação para estes resultados prende-se com a natureza do GPT-2 como modelo autoregressivo. Enquanto modelos como o BERT foram concebidos para tarefas de compreensão textual (e, portanto, mais adequados para classificação), o GPT-2 foi originalmente projetado para geração de texto, não sendo naturalmente otimizado para tarefas de classificação supervisionada. Assim, mesmo após *fine-tuning*, a sua capacidade de capturar padrões discriminativos neste domínio revelou-se limitada.

Em síntese, o GPT-2 demonstrou ser ineficiente para a tarefa de classificação de emoções em letras musicais, apresentando baixo desempenho em todas as métricas e elevados tempos de execução. A comparação direta com o BERT reforça esta conclusão, destacando o impacto das diferenças arquiteturais entre modelos de linguagem de larga escala.

Tabela 24: Classificação LLM GPT2

Tipo Features:	Accuracy	F1-score	Precision	AUC	Matriz	Tempo
Features_Original_letra	0,28	0,11	0,07	0,4875	51/180	523s
abstractive_distil_bart	0,24	0,1675	0,16	0,4975	43/180	549s
abstractive_led	0,26	0,16	0,2275	0,5	47/180	579s
abstractive_pegasus_dailymail	0,23	0,1475	0,1825	0,48	41/180	529s
abstractive_t5	0,29	0,175	0,375	0,505	52/180	531s
abstrativa_distilbart	0,28	0,17	0,1475	0,5475	50/180	579s
Features_abstractive_bart	0,28	0,11	0,07	0,46	51/180	573s
lex_rank	0,21	0,1125	0,1775	0,5075	38/180	562s
lsa	0,28	0,175	0,1375	0,505	51/180	622s
luhn	0,31	0,26	0,25	0,515	56/180	559s
text_rank	0,25	0,13	0,1575	0,535	45/180	565s
z.BERT	0,26	0,13	0,1575	0,555	46/180	523s
z.GPT2	0,28	0,165	0,1425	0,5075	50/180	537s

6. COMPARAÇÃO DE MODELOS

A Tabela 25 resume os resultados obtidos por todos os classificadores quando avaliados com as *features* originais extraídas das letras musicais, permitindo uma comparação direta do seu desempenho.

Tabela 25: Resultados da classificação com letras Originais

Modelo	Accuracy	F1-score	Precision	AUC	Matriz
SVM	0,72	0,7125	0,7175	0,915	127/180
Random Forest	0,65	0,645	0,66	0,83	117/180
Neuronal MLP	0,77	0,7625	0,7675	0,9225	135/180
Neuronal H20	0,73	0,725	0,73	0,9075	128/180
LLM BERT	0,68	0,675	0,765	0,9270	123/180
LLM GPT2	0,28	0,11	0,07	0,4875	51/180

Entre os modelos testados, o *multilayer perceptron* (MLP) destacou-se como o de melhor desempenho global, alcançando uma accuracy de 0,77, *F1-score* de 0,7625, precision de 0,7675 e a maior AUC (0,9225). Estes resultados evidenciam a capacidade das redes neuronais de capturar relações não lineares complexas nas features extraídas das letras, algo particularmente relevante dada a subjetividade associada às emoções.

O modelo SVM apresentou resultados sólidos (*F1-score* = 0,7125; AUC = 0,915), confirmando a sua robustez como classificador em cenários com alta dimensionalidade, embora sem atingir o nível de desempenho do MLP. O *random forest*, apesar de ser competitivo, obteve métricas mais modestas (*F1-score* = 0,645; AUC = 0,83), sugerindo que a sua abordagem baseada que combina várias árvores de decisão não conseguiu capturar com a mesma eficácia as nuances emocionais presentes nos textos.

No caso das redes neuronais com implementação em H2O.ai, os resultados foram consistentes (*F1-score* = 0,725; AUC = 0,9075), mas acompanhados de tempos de execução significativamente mais elevados, o que limita a sua aplicabilidade em cenários práticos.

Entre os modelos de linguagem de larga escala, o BERT apresentou métricas competitivas (*F1-score* = 0,675; AUC = 0,927), mostrando-se especialmente eficaz na métrica de

AUC, embora o custo computacional tenha sido elevado. Em contrapartida, o GPT-2 registou um desempenho muito baixo (*F1-score* = 0,11; AUC = 0,4875), próximo de um comportamento aleatório. Este resultado confirma as limitações da arquitetura autoregressiva do GPT-2 para tarefas de classificação, em contraste com a arquitetura bidirecional do BERT, mais adequada para compreensão textual.

Os resultados permitem concluir que:

- O MLP tradicional foi o modelo com melhor equilíbrio entre desempenho e eficiência, sendo adotado como modelo de referência nesta fase da análise.
- O SVM e o BERT apresentaram desempenhos robustos, posicionando-se como alternativas competitivas.
- O *Random Forest* e a rede em H2O.ai obtiveram resultados medianos, úteis em determinados contextos, mas menos eficientes.
- O GPT-2 demonstrou ser inadequado para a tarefa em estudo.

A tabela 26 apresenta os melhores resultados obtidos quando a classificação foi realizada sobre as letras sumarizadas em vez das letras originais. Este cenário permite avaliar qual foi a melhor técnica de classificação referente a sua respetiva sumarização e até que ponto os diferentes métodos de sumarização preservam informação relevante para a tarefa de classificação emocional.

Tabela 26: Resultados da classificação com features Sumarizadas

Sumarização	Melhor modelo	Accuracy	F1-score	Precision	AUC	Matriz
abstractive_distil_bart	Neuronal MLP	0,64	0,645	0,645	0,825	116/180
abstractive_led	SVM	0,56	0,5425	0,5925	0,7925	101/180
abstractiv_pegasus_dailymail	BERT	0,55	0,5175	0,7425	0,8912	99/180
abstractive_t5	Neuronal H20	0,62	0,615	0,625	0,8425	111/180
abstrativa_distilbart	Neuronal H20	0,63	0,6275	0,6375	0,8275	114/180
Features_abstractive_bart	BERT	0,63	0,615	0,775	0,9026	114/180
lex_rank	BERT	0,61	0,5875	0,74	0,9086	109/180
lsa	BERT	0,58	0,565	0,7275	0,9068	105/180
luhn	SVM	0,59	0,5875	0,6025	0,8075	107/180
text_rank	Neuronal MLP	0,69	0,6925	0,7025	0,88	124/180
z.BERT	Neuronal MLP	0,70	0,70	0,70	0,8775	126/180
z.GPT2	Neuronal MLP	0,58	0,5825	0,5825	0,7875	105/180

De entre todas as combinações testadas, o melhor resultado foi alcançado com a sumarização z.BERT, utilizando o modelo *multilayer perceptron* (MLP), que obteve uma *accuracy* de 0,70, *F1-score* de 0,70, *precision* de 0,70 e uma AUC de 0,8775. Estes valores mostram um desempenho bastante sólido e muito próximo daquele observado com as letras originais completas, reforçando a viabilidade de uso da sumarização como uma forma eficaz de compressão da informação sem perda significativa de desempenho.

No conjunto dos métodos extrativos, destacaram-se o TextRank (*F1-score* = 0,6925, AUC = 0,88) e o LexRank (*F1-score* = 0,5875, AUC = 0,9082), demonstrando capacidade de preservar a informação essencial mesmo em versões reduzidas dos textos. Já entre os métodos abstrativos, verificaram-se desempenhos mais modestos (ex.: PEGASUS, LED e T5 com F1-score entre 0,51 e 0,62), embora ainda aceitáveis para contextos de redução extrema de dimensionalidade.

Um aspeto importante é que, em alguns cenários, os modelos BERT e H2O apresentaram desempenhos competitivos, mas o MLP voltou a destacar-se como o modelo mais consistente. Isto confirma a sua robustez e capacidade de adaptação, tanto em representações originais quanto em versões sumarizadas.

Os resultados demonstram que a aplicação de técnicas de sumarização, em particular baseadas em LLMs como o BERT, pode reduzir substancialmente o volume de informação mantendo níveis de desempenho muito próximos dos observados com o texto completo. Este achado é relevante para contextos em que a eficiência computacional e a redução de dimensionalidade são fatores críticos.

7. RESULTADOS

A análise comparativa realizada permite destacar três eixos principais: o impacto do uso da letra completa versus letras sumarizadas, a diferença de desempenho entre modelos clássicos e modelos de *deep learning*, e a influência do tipo de técnica de sumarização aplicada.

7.1 Modelos com sumarização vs. letras completas

Os modelos treinados com as letras completas apresentaram, de forma consistente, o melhor desempenho global, sobretudo em métricas como o *F1-score* e a AUC. Contudo, as melhores técnicas de sumarização — em particular o z.BERT e o TextRank — alcançaram resultados bastante próximos dos textos originais.

Este resultado confirma que a utilização de resumos bem construídos não compromete significativamente a precisão dos modelos, permitindo reduzir a dimensionalidade e, consequentemente, o custo computacional. Assim, a sumarização revela-se uma alternativa válida em contextos onde a eficiência é um requisito crítico.

7.2 *Deep learning* vs. modelos clássicos

Os modelos baseados em redes neuronais (MLP e H2O MLP) superaram consistentemente os modelos clássicos (SVM e *Random Forest*), tanto nas letras originais como nas versões sumarizadas.

Este comportamento reforça a conclusão de que os modelos de *deep learning* são mais eficazes para a previsão de emoções em letras musicais, pela sua capacidade de capturar relações não lineares complexas entre variáveis, que os modelos clássicos não conseguem modelar com igual eficácia.

Apesar disso, modelos como o SVM mantiveram resultados competitivos, confirmando a sua relevância como alternativa em cenários de menor disponibilidade computacional.

7.3 Impacto do tipo de sumarização

Ao analisar comparativamente os diferentes métodos, verificou-se que as técnicas de sumarização extrativa (como z.BERT e TextRank) foram mais eficazes do que várias das técnicas abstrativas, como PEGASUS e LED, cujos F1-scores ficaram abaixo de 0,55.

Isto sugere que a sumarização extrativa preserva melhor os elementos emocionais essenciais da letra, mantendo palavras-chave e expressões diretamente relacionadas com os sentimentos transmitidos. Já os métodos abstrativos, ao reformular os textos, podem introduzir perda de nuances semânticas relevantes para a classificação de emoções.

8. CONCLUSÕES E TRABALHO FUTURO

No presente capítulo apresentam-se as conclusões finais do trabalho desenvolvido, organizadas de forma a revisitar o problema de investigação, sintetizar os principais resultados obtidos e discutir em que medida os objetivos propostos foram atingidos. Além disso, são identificados os contributos da investigação e apontadas possíveis linhas de trabalho futuro, capazes de dar continuidade e aprofundar as questões aqui exploradas.

8.1 Revisão do problema e objetivos

Esta investigação partiu da seguinte questão central:

Será que a utilização de letras sumarizadas (por métodos extrativos e abstrativos) permite treinar modelos de classificação de emoções que mantenham ou até melhorem o desempenho obtido com letras completas, sem comprometer a integridade da informação emocional?

Para responder a esta questão, definiu-se como objetivo principal avaliar o impacto da sumarização automática de letras musicais no reconhecimento automático de emoções, explorando tanto técnicas extrativas como abstrativas e aplicando diferentes modelos de machine learning e deep learning.

Entre os objetivos secundários, destacaram-se:

- Implementar e comparar métodos de sumarização.
- Extrair e selecionar features a partir de letras originais e resumidas.
- Treinar e avaliar modelos clássicos e modelos de deep learning.

8.2 Síntese dos resultados

Os resultados obtidos permitem tirar várias conclusões relevantes:

1. Letras completas vs. sumarizadas – O uso das letras completas assegurou os melhores resultados globais. No entanto, algumas técnicas de sumarização, sobretudo z.BERT e TextRank, atingiram desempenhos próximos, evidenciando que é possível reduzir a dimensionalidade textual sem perda expressiva de desempenho.
2. Modelos de *machine learning* – O *multilayer perceptron* (MLP) destacou-se como o modelo mais robusto, alcançando os melhores valores de *F1-score* e AUC em praticamente todos os cenários. Modelos clássicos como o SVM mantiveram resultados competitivos, enquanto o *random forest* foi menos eficaz.
3. Modelos de larga escala (LLMs) – O BERT apresentou resultados consistentes, especialmente em termos de AUC, confirmando a sua adequação para tarefas de

classificação de texto. Já o GPT-2 obteve métricas muito baixas, reforçando que a sua arquitetura não é a mais apropriada para este tipo de tarefa.

4. Extrativo vs. Abstrativo – As técnicas de sumarização extrativa preservaram melhor os elementos emocionais essenciais, levando a desempenhos superiores. As técnicas abstrativas, por vezes, introduziram perdas de informação ou ruído, resultando em métricas mais baixas (*F1-score* < 0,55 em alguns casos).

8.3 Cumprimento dos objetivos

Os objetivos inicialmente definidos foram cumpridos. Demonstrou-se a viabilidade da aplicação de técnicas de sumarização automática em *Music Emotion Recognition* (MER), explorando tanto métodos extrativos como abstrativos, bem como a utilização de LLMs. Os resultados confirmaram que a redução do texto através de sumarização pode, em muitos casos, preservar a informação emocional necessária para a classificação, sendo que técnicas extrativas como o z.BERT se destacou pela sua capacidade de gerar resumos consistentes.

Outro objetivo importante consistia em identificar os modelos de classificação mais eficazes. A análise revelou que os modelos de deep learning, nomeadamente o *multilayer perceptron* (MLP), apresentaram o melhor desempenho global, superando modelos clássicos como o SVM e o *random forest*. O H2O MLP e os modelos com *fine-tuning* (DistilBERT e GPT-2) também mostraram resultados competitivos, reforçando a relevância da aprendizagem profunda no contexto do MER.

Foi ainda cumprido o objetivo de comparar de forma sistemática métodos de sumarização e algoritmos de classificação, permitindo observar não apenas os ganhos e perdas no desempenho, mas também como diferentes técnicas de compressão textual influenciam a preservação de sinais emocionais. Este cruzamento entre estratégias contribui para um melhor entendimento da interação entre pré-processamento textual e desempenho em tarefas de classificação de emoções.

Apesar destes avanços, algumas limitações foram identificadas: a dimensão e diversidade reduzida do corpus (180 letras), que limita a generalização dos resultados, o elevado custo computacional de alguns modelos, como o GPT-2, que exigiram recursos significativos para treino e a instabilidade de alguns modelos abstrativos, como PEGASUS e LED, que frequentemente geraram resumos incoerentes ou de tamanho próximo ao original, comprometendo o objetivo da compressão.

8.4 Contributos

O presente estudo oferece vários contributos relevantes para a comunidade científica e para a prática no campo do MER. Em primeiro lugar, demonstra a integração de técnicas de sumarização automática com classificação de emoções musicais, evidenciando que é possível reduzir substancialmente o tamanho das letras sem perdas significativas de desempenho. Esta constatação abre caminho para sistemas mais eficientes do ponto de vista computacional, especialmente úteis em contextos de processamento em larga escala.

Em segundo lugar, o trabalho fornece uma análise comparativa detalhada entre diferentes famílias de algoritmos. Métodos clássicos de *machine learning*, redes neuronais tradicionais (MLP), *frameworks* otimizadas (H2O MLP) e modelos de linguagem de larga escala com *fine-tuning* (BERT e GPT-2). Esta análise reforça a perceção de que os modelos baseados em *deep learning* apresentam maior robustez e capacidade de generalização na tarefa de MER.

Por fim, outro contributo científico prende-se com a avaliação crítica do papel da sumarização extrativa e abstrativa em tarefas emocionais. Enquanto os métodos extrativos demonstraram maior estabilidade na preservação de conteúdo emocional, os métodos abstrativos revelaram as suas fragilidades, sobretudo na consistência e relevância dos resumos. Esta comparação fornece *insights* valiosos para futuras investigações, ajudando a orientar escolhas metodológicas em trabalhos semelhantes.

8.5 Trabalho futuro

Com base nos resultados obtidos, identificam-se várias linhas de investigação que podem dar continuidade a este estudo e superar algumas das suas limitações:

Expansão e diversificação do corpus

Uma das limitações mais evidentes foi a dimensão relativamente reduzida e homogénea do dataset (180 letras). Como trabalho futuro, propõe-se a expansão do corpus, não apenas em termos de volume, mas também de diversidade linguística e cultural. Incluir letras de diferentes idiomas, géneros musicais e períodos históricos permitirá testar a generalização dos resultados e verificar se os modelos continuam robustos em contextos mais variados.

Novas representações textuais e LLMs

Apesar de terem sido explorados modelos como BERT e GPT-2, a rápida evolução dos LLMs abre espaço para testar embeddings mais recentes, como variantes otimizadas de BERT (RoBERTa, DeBERTa), ou modelos de última geração como GPT-4 e LLaMA-2. A utilização destes modelos poderá fornecer representações semânticas mais ricas, captando nuances emocionais que escapam a modelos anteriores.

***Transfer learning* entre géneros e contextos**

Outra linha de investigação relevante passa por aplicar *transfer learning*, testando até que ponto modelos treinados em determinados géneros ou idiomas conseguem ser adaptados a novos contextos musicais com menos dados anotados. Esta abordagem permitiria aumentar a escalabilidade dos sistemas de MER e reduzir a necessidade de construção de datasets extensos.

Exploração de novas *features* e estratégias de extração

Por fim, será importante explorar novos tipos de *features* para complementar as atuais, incluindo atributos linguísticos mais sofisticados (como sintaxe e semântica profunda), ou até indicadores estilísticos e retóricos próprios de letras musicais (ex.: repetição de refrões, intensidade emocional de palavras-chave).

9. REFERÊNCIAS

Beltagy, I., Peters, M. E., & Cohan, A. (2020). *Longformer: The Long-Document Transformer*. https://arxiv.org/pdf/2004.05150

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., … Liang, P. (2021). *On the Opportunities and Risks of Foundation Models*. https://arxiv.org/pdf/2108.07258

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory (pp. 144–152). ACM. https://doi.org/10.1145/130385.130401

Bradley, M. M., & Lang, P. J. (1999). *Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings*.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 4171–4186. https://arxiv.org/pdf/1810.04805

Ekman, P. (1992). Are there basic emotions? In P. Ekman & R. J. Davidson (Eds.), The nature of emotion: Fundamental questions (pp. 15–19). Oxford University Press. https://psycnet.apa.org/record/1992-41830-001

Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479. https://doi.org/10.1613/JAIR.1523

Gabrielsson, Ann-Marie (2002). Emotion perceived and emotion felt: Same or different? *Journals.Sagepub.ComA GabrielssonMusicae Scientiae, 2001•journals.Sagepub.Com*, *5*(1_suppl), 123–147. https://doi.org/10.1177/10298649020050S105

Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media. https://books.google.com/books?id=X5ySEAAAQBAJ

Gong, Y., & Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 19–25. https://doi.org/10.1145/383952.383955

Gross, J. J., & Munoz, R. F. (1998). Emotion regulation and mental health. Clinical Psychology: Science and Practice, 5(3), 151–164. https://doi.org/10.1111/j.1468-2850.1998.tb00119.x

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter, 11(1), 10–18. https://doi.org/10.1145/1656274.1656278

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching Machines to Read and Comprehend. *Advances in Neural Information Processing Systems*, *2015-January*, 1693–1701. https://arxiv.org/pdf/1506.03340

Hevner, K. (1936). Experimental studies of the elements of expression in music. The American Journal of Psychology, 47(2), 162–183. https://www.jstor.org/stable/1415746

Hu, X., Downie, J. S., & Ehmann, A. F. (2009). Lyric text mining in music mood classification. In Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009) (pp. 411–416). https://experts.illinois.edu/en/publications/lyric-text-mining-in-music-mood-classification

Hu, X., Downie, J. S., & Ehmann, A. F. (2010). When lyrics outperform audio for music mood classification: A feature analysis. Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010), 106–111. https://ismir2010.ismir.net/proceedings/ismir2010-106.pdf

James, W. (2000). The principles of psychology (Vol. 1). Dover Publications. (Original work published 1890)

Juslin, P. N., & Laukka, P. (2004). Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening. Journal of New Music Research, 33(3), 217–238. https://doi.org/10.1080/0929821042000317813

Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., Speck, J. A., & Turnbull, D. (2010). Music emotion recognition: A state of the art review. Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010), 255–260. https://archives.ismir.net/ismir2010/paper/000045.pdf

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. In European Conference on Machine Learning (Vol. 784, pp. 171–182). Springer. https://doi.org/10.1007/3-540-57868-4_57

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. Springer. https://doi.org/10.1007/978-1-4614-6849-3

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural

Language Generation, Translation, and Comprehension. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 7871–7880. https://doi.org/10.18653/V1/2020.ACL-MAIN.703

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, *2*(2), 159–165. https://doi.org/10.1147/RD.22.0159

Malheiro, R. M. da S. (2016). Emotion-based analysis and classification of music lyrics (Tese de doutoramento). ProQuest Dissertations Publishing. https://search.proquest.com/openview/c3de8f1216bd6c1e19cc4c996bab5cc2/1?pq-origsite=gscholar&cbl=2026366&diss=y

Malheiro, R., Panda, R., Gomes, P., & Paiva, R. P. (2018). Emotionally-Relevant Features for Classification and Regression of Music Lyrics. IEEE Transactions on Affective Computing, 9(2), 240–254. https://doi.org/10.1109/TAFFC.2016.2598569

Mayer, R., Neumayer, R., & Rauber, A. (2008). Rhyme and style features for musical genre classification by song lyrics

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004) (pp. 404–411). Association for Computational Linguistics. https://aclanthology.org/W04-3252/

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In C. C. Aggarwal & C. Zhai (Eds.), Mining text data (pp. 43–76). Springer. https://doi.org/10.1007/978-1-4614-3223-4_3

Powers, D. M. W. (2011). *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. https://arxiv.org/pdf/2010.16061

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*(140), 1–67. http://jmlr.org/papers/v21/20-074.html

Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2 (3rd ed.). Packt Publishing

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408. https://psycnet.apa.org/journals/rev/65/6/386/

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. https://www.nature.com/articles/323533a0

Russell, J. A. (1980). A circumplex model of affect. Journal of Personality and Social Psychology, 39(6), 1161–1178. https://doi.org/10.1037/h0077714

Russell, J. A. (2003). Core affect and the psychological construction of emotion. Psychological Review, 110(1), 145–172. https://doi.org/10.1037/0033-295X.110.1.145

Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys, 34(1), 1–47. https://doi.org/10.1145/505282.505283

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/J.IPM.2009.03.002

Stone, Philip J., Dunphi Dexter C., Smith S. Marshall, O. D. M. (1966). The General Inquirer. *Journal of Regional Science*, *8*(1), 113–116.

Tausczik, Y. R., & Pennebaker, J. W. (2010). The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, *29*(1), 24–54. https://doi.org/10.1177/0261927X09351676

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS). https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Whissell, C. (2009). Using the Revised Dictionary of Affect in Language to Quantify the Emotional Undertones of Samples of Natural Language. *Psychological Reports*, *105*(2), 509–521. https://doi.org/10.2466/PR0.105.2.509-521

Yang, Y. H., Lin, Y. C., Su, Y. F., & Chen, H. H. (2012). A regression approach to music emotion recognition. IEEE Transactions on Audio, Speech, and Language Processing, 16(2), 448–457. https://doi.org/10.1109/TASL.2007.912645

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2019). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *37th International Conference on Machine Learning, ICML 2020*, *PartF168147-15*, 11265–11276. https://arxiv.org/pdf/1912.08777