DESENVOLVIMENTO DE MODELOS PREDITIVOS DE EMOÇÕES MUSICAIS ATRAVÉS DE TEXT MINING

Marta Ferreira Rodrigues

Projeto inserido no Mestrado de Ciência de Dados

Leiria, abril 2026

DESENVOLVIMENTO DE MODELOS PREDITIVOS DE EMOÇÕES MUSICAIS ATRAVÉS DE TEXT MINING

Marta Ferreira Rodrigues

Projeto inserido no Mestrado de Ciência de Dados, orientado pelo professor Ricardo Malheiro

Leiria, abril 2026

Agradecimentos

Em primeiro lugar, agradeço ao meu orientador, pela disponibilidade e apoio ao longo de todo o desenvolvimento desta tese. A sua experiência e aconselhamento foram fundamentais para a concretização deste projeto.

Aos meus familiares, pelo apoio incondicional, compreensão e incentivo ao longo deste percurso académico.

Às minhas amigas, pela motivação, apoio e momentos de descontração, que foram essenciais para manter o equilíbrio ao longo deste processo.

Por fim, um agradecimento especial ao meu namorado, pelo apoio constante, paciência e encorajamento em todos os momentos.

Abstract

Music Information Retrieval (MIR) is a research field focused on the organization and analysis of information extracted from music. Within this domain, Music Emotion Recognition (MER) aims to identify the emotions expressed, particularly through song lyrics.

The main objective of this work is to explore the contribution of stylistic features, specifically metaphors, to the automatic detection of emotions in song lyrics, using Text Mining and Machine Learning techniques. To this end, several classification models were developed for metaphor detection, including both traditional approaches and Deep Learning models such as BERT and RoBERTa. The best-performing model was RoBERTa with *fine-tuning*, achieving approximately 92% accuracy in sentence classification.

Based on this model, features related to the presence and distribution of metaphors in lyrics were extracted and subsequently integrated with previously defined features. These were then used in different classification models to predict the emotional quadrant of each song.

The results show a slight improvement in the performance of traditional models with the inclusion of metaphor-based features. However, in Deep Learning models, a decrease in performance was observed in the emotional classification task.

Keywords

Music Emotion Recognition, Music Information Retrieval, Machine Learning, Metaphor Detection, Natural Language Processing

Resumo

A Recuperação da Informação Musical (*Music Information Retrieval* - MIR) é uma área de investigação focada na organização e análise de informação extraída de música. Dentro deste domínio, o Reconhecimento de Emoções em Música (*Music Emotion Recognition* - MER) procura identificar as emoções expressas, nomeadamente através das letras.

Este trabalho tem como principal objetivo explorar o contributo de características estilísticas, em particular metáforas, na deteção automática de emoções em letras de músicas, recorrendo a técnicas de *Text Mining* e *Machine Learning*. Para tal, foram desenvolvidos vários modelos de classificação para a deteção de metáforas, incluindo abordagens tradicionais e modelos de *Deep Learning*, como BERT e RoBERTa. O modelo que apresentou melhor desempenho foi o RoBERTa com *fine-tuning*, alcançando cerca de 92% de precisão na classificação de frases.

A partir deste modelo, foram extraídas *features* relacionadas com a presença e distribuição de metáforas nas letras, que foram posteriormente integradas com outras características previamente definidas. Estas *features* foram utilizadas em diferentes modelos de classificação para prever o quadrante emocional das músicas.

Os resultados demonstram uma ligeira melhoria no desempenho dos modelos tradicionais com a inclusão das *features* de metáforas. No entanto, nos modelos baseados em *Deep Learning*, verificou-se uma diminuição do desempenho na tarefa de classificação emocional.

Palavras Chave

Recuperação da Informação Musical, Reconhecimento de Emoções em Música, *Machine Learning,* Deteção de Metáforas, Processamento de Linguagem Natural

Acrónimos

BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
DNN	Deep Neural Network
MER	Music Emotion Recognition
MIR	Music Information Retrieval
MLP	Multi-Layer Perceptron
PLN	Natural Language Processing
POS	Part Of Speech
RFC	Random Forest Classifier
RL	Logistic Regression
RNN	Recurrent Neural Network
RoBERTa	Robustly Optimized BERT Pretraining Approach
SGD	Stochastic Gradient Descent
SMER	Static Music Emotion Recognition
SVC	Support Vector Classifier
SVM	Support Vector Machine
TF-IDF	Term Frequency–Inverse Document Frequency
XLNet	eXtra Large Neural Networks
XGBoost	Extreme Gradient Boosting

Conteúdo

Lista de Figuras

Lista de Tabelas

Esta página foi intencionalmente deixada em branco.

1. Introdução

Vários autores escreveram diversos textos que nos relembram da importância da música no que toca à experiência humana. Escolher um que resuma a complexidade da música é praticamente impossível.

No que toca à escrita da letra de uma música, o livro *Writing Down the Bones* (Goldberg, 1986) lembra-nos que "Nós escrevemos no momento e refletimos as emoções daquele momento".

Portanto, podemos assumir que uma música, constituída por letra e áudio, tenta recriar ou dar enfase a um momento, emoções ou pensamentos do compositor, o que faz com que o ouvinte se sinta conectado com a música em si, uma vez que reflete experiências que podem ou não ser pessoais. Por exemplo, a banda sonora de um filme, tem como principal objetivo envolver ainda mais o espectador, fazendo-o sentir ao máximo a história do mesmo. Uma discoteca, tende a ter músicas com uma letra e áudio mais alegres (com mais batida e ritmo) e que façam com que as pessoas que a ouvem se sintam mais felizes e até com vontade de dançar.

A recuperação da informação musical (*Music Information Retrieval*, MIR) pretende, entre outras coisas, recuperar e organizar toda a informação musical, como por exemplo, o género e o artista; contextual, tal como a data de lançamento e o nome do álbum e ainda sobre o ouvinte como o histórico de audição e as suas preferências. O reconhecimento de emoções nas músicas (*Music Emotion Recognition*, MER) é uma área do MIR, mas mais focado nas emoções que as músicas tendem a transmitir.

Ao longo dos anos o MIR tornou-se cada vez mais importante nomeadamente para as plataformas de música como *Spotify* e *Apple Music*[1], que criam playlists personalizadas consoante o género, tipo e emoções das músicas que contêm. Estas plataformas necessitam de catalogar centenas de músicas diariamente e como é impraticável fazê-lo manualmente, precisam de extrair informação que permita alimentar modelos automáticos de previsão, daí ser tão importante o seu constante estudo e aperfeiçoamento dos métodos que utiliza.

[1] *Spotify* e *Apple Music* utilizam sistemas de recomendação baseados em algoritmos de análise de dados e aprendizagem automática para sugerir conteúdos personalizados aos utilizadores (Schedl et al., 2018)

Este projeto desenvolve-se parcialmente no âmbito do projeto MERGE (*Music Emotion Recognition – Next Generation*, PTDC/CCI-COM/3171/2021) financiado pela Fundação para a Ciência e Tecnologia (FCT).

1.1. Descrição do problema, motivação e escopo

A identificação automática de emoções em música tem sido amplamente estudada no contexto de MER. Esta tarefa é particularmente relevante para aplicações como sistemas de recomendação musical, organização automática de bibliotecas musicais e análise de conteúdo multimédia.

De acordo com (Juslin & Laukka, 2004) 29% das pessoas indicam que a letra é um fator importante na transmissão de emoções da música.

Estudos anteriores indicam que abordagens bimodais, que analisam simultaneamente o áudio e as letras das músicas, tendem a apresentar melhores resultados na deteção de emoções do que abordagens baseadas apenas numa destas fontes de informação. Ao longo dos anos, vários trabalhos têm procurado identificar *features* relevantes que permitam melhorar o desempenho destes modelos.

No entanto, apesar dos avanços alcançados, a identificação de emoções a partir das letras continua a ser um desafio, principalmente devido à natureza figurativa e ambígua da linguagem utilizada nas músicas. Muitas letras recorrem frequentemente a figuras de estilo, como metáforas, para expressar estados emocionais, o que dificulta a sua interpretação automática por sistemas computacionais.

De acordo com (Malheiro, 2016) as *features* utilizadas na classificação de emoções em letras de músicas podem ser agrupadas em quatro categorias principais:

1. **Baseadas no conteúdo**: com e sem as transformações típicas de Processamento de Linguagem Natural (PLN). As letras incluem, por exemplo, representações textuais como *n-grams*, com ou sem transformações típicas de PLN, tais como *stemming* e *Part-of-Speech tagging* (POS tags);
2. **Estilísticas**: como o número de ocorrências de sinais de pontuação, classes gramaticais e, dentro destas, as figuras de estilo onde se inserem as metáforas;
3. **Estrutura**: como o número de ocorrências de versos e do refrão.

4. **Semânticas:** baseadas em léxicos conhecidos extraídos de várias frameworks, como por exemplo GI e LIWC [2]

Embora muitas abordagens explorem características baseadas no conteúdo ou em léxicos semânticos, as *features* estilísticas relacionadas com o uso de linguagem figurativa têm sido menos exploradas, apesar do seu potencial para capturar informação emocional implícita nas letras.

Neste contexto, o presente projeto foca-se especificamente na deteção de metáforas em letras de músicas, com o objetivo de avaliar se esta figura de estilo pode contribuir como uma feature relevante para a identificação automática de emoções.

(Malheiro, 2016) obteve resultados quase idênticos para ativação (0,61) e resultados muito melhores para valência (0,64). Como referência de trabalhos anteriores, reuniu diversas *features* utilizadas nesta área, alcançando um *F1-score* de 76,4% e uma acurácia de 69,8% na identificação de emoções.

Assim, este trabalho pretende investigar se a introdução de *features* baseadas em metáforas pode contribuir para melhorar o desempenho dos modelos de classificação de emoções em letras de músicas.

1.2. Objetivos e abordagens

Este projeto tem como principal objetivo perceber como é que as características estilísticas podem facilitar a deteção de emoção baseando-se apenas na letra da música. Assim, pretendemos:

- Desenvolver métodos para identificar metáforas nas letras das músicas, através da criação de modelos preditivos treinados com bases de dados previamente anotadas;
- Extrair *features* associadas às metáforas identificadas, de forma a representar informação estilística potencialmente relevante do ponto de vista emocional;
- Extrair essas características para comparar o desempenho dos modelos com e sem as *features* de metáforas, de modo a analisar o contributo destas características para a tarefa de MER;

[2] *GI (General Inquirer)* e *LIWC (Linguistic Inquiry and Word Count)* são recursos léxicos e semânticos amplamente utilizados em PLN, contendo anotações psicológicas, emocionais e semânticas que permitem analisar o conteúdo afetivo e cognitivo do texto

- Identificar as *features* mais relevantes, recorrendo a técnicas de seleção de características, com vista à construção de modelos mais eficazes e interpretáveis.

1.3. Planeamento

De modo a organizar todo o processo de elaboração deste projeto, foi essencial realizar um diagrama de *Gantt* com todas as tarefas e prazos para a realização das mesmas. A Figura 1 contém esse mesmo diagrama com a duração expectável e real.

Diagrama de Gantt

DESENVOLVIMENTO DE MODELOS PREDITIVOS DE EMOÇÕES MUSICAIS ATRAVÉS DE TEXT MINING

Legenda:
Planeado
Real

Figura 1 - Diagrama de Gantt

1.4. Estrutura do trabalho

Este relatório está organizado em nove capítulos. Começando com a introdução que identifica o problema do projeto e qual a sua motivação, identificação de objetivos e abordagens utilizadas na realização do mesmo e qual o seu planeamento.

O segundo capítulo contém a descrição e identificação de alguns conceitos que são importantes para o entendimento de todo o projeto. O terceiro capítulo, designado de "Estado de Arte" serve como resumo de tudo o que já foi elaborado sobre este tema até à data do projeto. Contém ainda a descrição de algumas técnicas que serão relevantes na parte prática do trabalho. O quarto capítulo descreve as bases de dados e corpus que foram utilizados para a elaboração do projeto.

Entrando na parte mais prática de todo o trabalho, o quinto capítulo descreve todo o processo de identificação de metáforas (preparação dos dados e desenvolvimento dos modelos) e revela ainda qual o melhor modelo a utilizar para a parte final do projeto.

O sexto e o sétimo capítulo resumem os resultados obtidos, identificam quais as conclusões retiradas e exploram o trabalho e melhorias que podem ser realizadas nos próximos tempos.

Os últimos dois capítulos são destinados às referências e anexos do projeto.

2. Conceitos Teóricos

Este capítulo tem como objetivo apresentar os principais conceitos teóricos necessários para a compreensão e enquadramento do presente trabalho. São analisadas diferentes abordagens à representação das emoções, bem como modelos utilizados na sua caracterização. Adicionalmente, são introduzidos conceitos relacionados com figuras de estilo, nomeadamente as metáforas, e outros elementos fundamentais para o desenvolvimento do projeto.

2.1. Emoção

Neste capítulo iremos abordar noções essenciais para a compreensão deste projeto, acerca das emoções.

2.1.1. Definição de emoção

De acordo com o dicionário da língua portuguesa (Priberam), emoção é o "conjunto de reações, variáveis na duração e na intensidade, que ocorrem no corpo e no cérebro". De acordo com (Martinazzo, 2010) emoções são estados mentais e psicológicos associados a vários sentimentos, pensamentos e comportamentos.

Apesar de o ser humano sentir emoções diariamente, nem sempre é fácil descrever algo tão complexo e, por vezes, é mais simples defini-las através de uma experiência comum.

Talvez seja por isso, que letras de músicas sejam escritas depois de vivências do autor, que incapaz de se expressar de outra forma, escreve e compõe para que os outros o entendam e, quem sabe, percebam a sua forma de comunicar. Como por exemplo a música "*How do I Say Goodbye*" de *Dean Lewis* que escreveu a letra da música quando o seu pai foi diagnosticado com cancro, em 2019.

2.1.2. Tipos de emoção

Um dos principais objetivos do projeto é a deteção de emoções em letras de músicas. Por isso, é de extrema importância conhecermos quais os tipos de emoção. Existem três tipos de emoções numa música: expressa, percebida e sentida (Gabrielsson, 2002).

- A **emoção expressa** é aquela que o intérprete ou compositor pretende comunicar através da maneira como a interpreta ou compõe.
- A **emoção percebida** é a emoção que se percebe como sendo expressa numa canção quando se olha para ela objetivamente.
- A **emoção sentida** é a que está diretamente ligada ao ouvinte, tendo em conta eventos passados ou experiências que o conectam à música.

Resumindo, uma canção pode ter sido escrita a pensar numa emoção, mas no fim, vários ouvintes podem ter opiniões diferentes tendo em conta tudo o que essa música significa para cada pessoa. Isto deve-se à distinção entre emoção expressa, emoção percebida e emoção sentida, que, embora relacionadas, operam em níveis distintos do processo musical.

A emoção expressa reflete a intenção comunicativa do compositor ou intérprete, sendo moldada por escolhas estruturais e interpretativas; a emoção percebida corresponde à leitura que o ouvinte faz dessas mesmas características sonoras, podendo ou não coincidir com a intenção original; por fim, a emoção sentida emerge da experiência subjetiva individual, profundamente influenciada por fatores extramusicais, como memórias, contexto sociocultural e estado emocional do ouvinte no momento da escuta.

Assim, a experiência musical revela-se como um fenómeno complexo, no qual o significado emocional não é fixo, mas construído dinamicamente na interação entre obra, intérprete e ouvinte.

2.1.3. Modelos de emoções na música

Existem dois tipos de modelos que organizam as emoções e as representam de forma diferente (Malheiro, 2016).

Os **modelos categóricos** dividem as emoções em categorias ou clusters, formando grupos de emoções que são semelhantes entre si. Por exemplo, o modelo de (Hevner, 1936) (Figura 2) divide 67 adjetivos em 8 categorias num modelo circular. Isto faz com que em grupos vizinhos (por exemplo os grupos 1 e 2) se encontrem adjetivos emocionalmente próximos, enquanto em grupos opostos, os adjetivos são contrários (por exemplo os grupos 2 e 6).

Figura 2 - Modelo de Hevner - (Hevner, 1936) (adaptado de (Yang & Chen, 2012))

Os **modelos dimensionais** de representação de emoções têm como principal objetivo agrupar e organizar as emoções tendo em conta um espaço dimensional. O modelo de *James Russell*, apresentado na Figura 3, divide as emoções pela sua valência (eixo dos x) e a sua intensidade (eixo dos y).

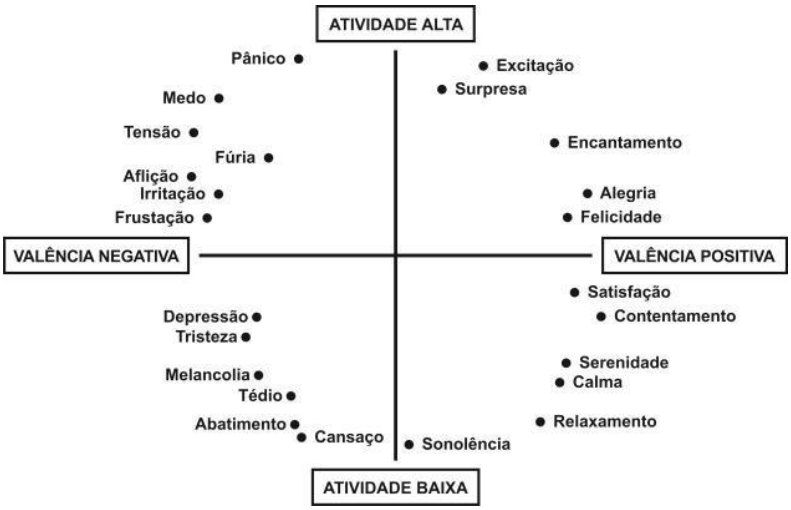

Figura 3 - Modelo dimensional de James Russell - (Russell, 1980)

Muitas vezes este modelo é também visto como categórico por quadrantes. Assim, o modelo (representado na Figura 3) está dividido em quatro quadrantes que representam o seguinte (Tabela 1):

Tabela 1 - Quadrantes do Modelo de James Russell

Quadrante		Descrição	Exemplos de emoções
1	Valência e intensidade positivas	Emoções alegres e energéticas	Entusiasmo e excitação
2	Valencia negativa e intensidade positiva	Emoções negativas muito fortes	Raiva, medo
3	Valência e intensidade negativas	Emoções negativas pouco energéticas	Tristeza, cansaço
4	Valencia positiva e intensidade negativa	Emoções positivas com pouca excitação	Serenidade e satisfação

2.2. Music Emotion Recognition (MER)

Todos nós, ouvintes de música, já ficamos insatisfeitos com a música que a estação de rádio escolheu. O tipo de música que queremos ouvir, varia consoante o nosso estado de espírito. Este problema levou à criação de várias plataformas de música, onde cada utilizador consegue escolher o que pretende ouvir, quando e onde o fazer. Para tal, foi necessária a recolha das informações das músicas (MIR), para que a sua busca na plataforma fosse possível. Informações como o nome do autor, nome do álbum, data de lançamento e a emoção que transmite são importantes para a pesquisa de uma música.

Este campo de estudo tem vindo a crescer e a ganhar cada vez mais reconhecimento. Neste projeto, vamos dar mais destaque a uma área do MIR que pretende reconhecer quais a emoções que uma música nos transmite (MER). Esta tarefa não é algo simples, pois como já vimos anteriormente, nem sempre é fácil expressar uma emoção de forma linear e percetível por todos. Para tal, existem algumas abordagens onde podemos utilizar características das letras e do áudio individualmente (unimodais), ou em união (bimodais).

2.2.1. Tipos de MER

Há dois tipos de MER (Yang & Chen, 2012):

1. SMER – *Static Music Emotion Recognition* – que tem como principal objetivo identificar a emoção predominante na música. Este tipo de reconhecimento é aplicado em plataformas de música, por exemplo, na criação de playlists próprias para várias emoções, como podemos ver na Figura 4.

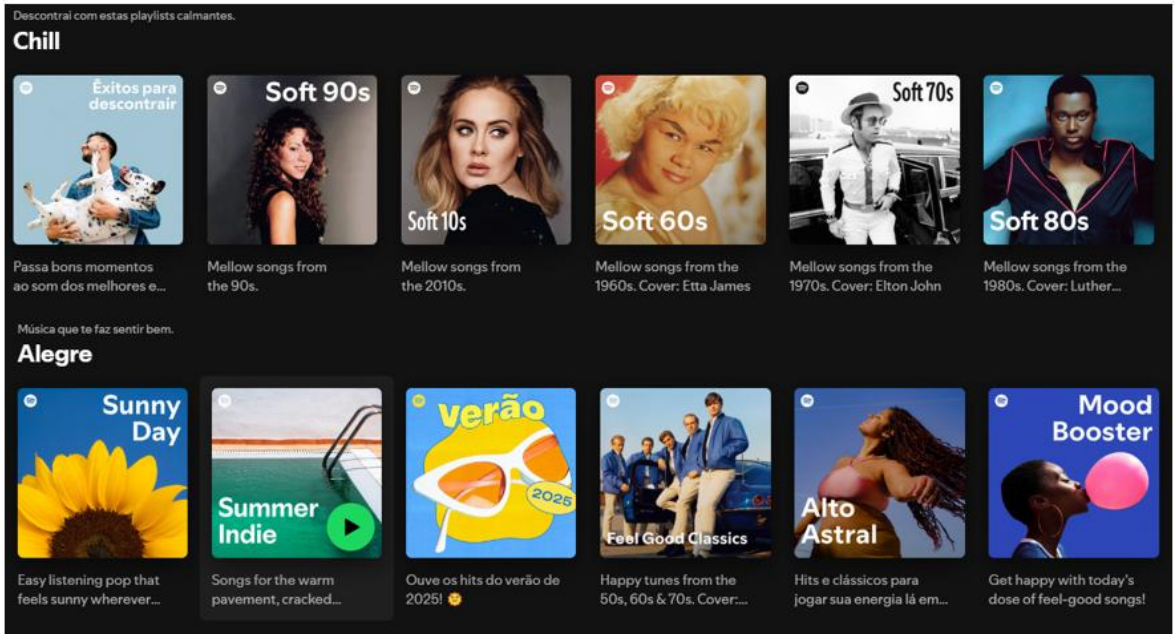

Figura 4 - Playlists criadas pelo Spotify

2. MEVD – *Music Emotion Variation Detection* – que analisa as alterações das emoções ao longo da música, e oferece uma análise mais precisa e contextualizada. Pode ser aplicado, por exemplo, em bandas sonoras de filmes onde a mesma música pode apresentar várias emoções que vão de acordo com a imagem e a ação do momento. Por exemplo a música "*Let it Go*" do filme *Frozen* de 2013, começa por revelar medo e insegurança, mas termina com orgulho e aceitação do destino da personagem.

2.2.2. Representação de Texto e Features para MER

A representação de texto desempenha um papel fundamental nas tarefas de MER, uma vez que a forma como a informação textual é codificada influencia diretamente a capacidade dos modelos em identificar padrões emocionais. Ao longo dos anos, diversas abordagens têm sido propostas, desde representações mais simples baseadas na frequência de termos até modelos mais complexos capazes de capturar relações semânticas e contextuais profundas.

Para além disso, a seleção de *features* relevantes – nomeadamente as associadas ao conteúdo linguístico e estilístico das letras – constitui um fator determinante no desempenho dos modelos de classificação emocional.

Neste contexto, esta secção apresenta as principais técnicas de representação de texto utilizadas em MER, bem como a importância da incorporação de *features* baseadas em figuras de estilo, como as metáforas, na modelação da dimensão emocional das letras de músicas.

Representações básicas

TF-IDF (*Term Frequency–Inverse Document Frequency*): É uma representação escalar que pesa a importância dos termos com base na sua frequência no documento e na raridade no corpus. É simples, eficiente e eficaz em muitos contextos de classificação de emoções a partir de texto. Contudo, trata cada palavra de forma independente, ignorando relações semânticas e contexto global do texto – limitando a sua capacidade de captar nuances emocionais subtis.

Word2Vec: Produz *embeddings* densos, aprendidos a partir de grandes corpora. Utiliza arquiteturas para capturar similaridades estatísticas entre palavras. Este método supera TF-IDF no que toca à captura contextual, mas gera representações estáticas: cada palavra tem um único vetor, sem distinção entre significados diferentes em contextos diversos – um obstáculo no caso de letras com ambiguidades ou metáforas.

GloVe: Cria *embeddings* densos com base em estatísticas globais de ocorrência de pares de palavras no corpus. Através da fatorização de uma matriz de co-ocorrências, obtém vetores que refletem tanto contexto local como relações globais entre palavras. Contudo, tal como *Word2Vec*, *GloVe* gera representações estáticas e sofre igualmente com problemas de polissemia.

Representações baseadas em contexto

Os modelos baseados em *Transformer* – como BERT e versões GPT – geram representações contextualizadas, ou seja, o vetor de uma palavra depende do seu uso na frase. Isto permite capturar polissemia e relações contextuais complexas, e leva a representações muito mais ricas e adequadas à análise emocional.

Especificamente no domínio de letras de música, abordagens baseadas em *Transformers* já mostram resultados promissores. Por exemplo, o estudo de (Agrawal et al., 2021), que utiliza o *Extra Large Neural Networks* (XLNet) como arquitetura base, obteve desempenho superior a métodos anteriores baseados em Redes Neuronais Recorrentes (RNN), *Bidirectional Encoder Representations from Transformers* (BERT) ou técnicas de extração de características estruturais, estabelecendo novos patamares de acurácia e F1-score para reconhecimento de emoções em músicas a partir de letras.

Features de letras baseadas em figuras de estilo (metáforas)

As metáforas desempenham um papel crucial na expressão emocional nas letras de música – evocam sensações, emoções e imagens que vão além do significado literal. Incorporar *features* estilísticas como metáforas permite ao modelo aceder a ligações emocionais mais profundas.

As metáforas podem criar pontes entre o emocional e o racional, através da transformação de conceitos abstratos. Deste forma, permitem a ativação de sentimentos pela forma como são interpretadas pelo ouvinte.

(Seyeditabari et al., 2019) Mostram que a adição de informação emocional em *embeddings* pré treinados melhora as métricas de similaridade emocional. Utilizando dois conjuntos de restrições: pares palavra-emoção positivos e pares palavra-emoção com a emoção oposta. Os resultados mostram melhorias nas métricas de similaridade emocional. Isto vem provar que a incorporação de restrições emocionais melhora significativamente a capacidade dos *embeddings* de refletir relações emocionais entre palavras, abrindo caminho para aplicações mais precisas em deteção de emoções.

Este tipo de abordagem pode levar a que a representação das metáforas – e outras figuras de estilo – fique mais alinhada com a emoção que expressam, o que é altamente relevante para MER e justifica a inclusão dessas *features* no projeto.

2.3. Machine Learning

O *Machine Learning* é uma área da inteligência artificial que permite a sistemas computacionais aprender padrões a partir de dados e melhorar o seu desempenho em tarefas específicas sem necessidade de programação explícita para cada caso. Estes sistemas constroem modelos matemáticos capazes de generalizar conhecimento a partir de exemplos previamente observados.

De forma geral, o funcionamento de um sistema de *Machine Learning* pode ser dividido em várias etapas fundamentais:

1. Preparação e análise de dados fornecidos

Esta etapa consiste na recolha, organização e pré-processamento dos dados que serão utilizados para treinar os modelos. No contexto da deteção de emoções em letras de músicas, este processo envolve a construção de um dataset com letras de músicas associadas a categorias emocionais, podendo estas ser definidas, por exemplo, com base em modelos como o Valência - Ativação de Russell.

A qualidade dos dados é um fator crítico para o desempenho dos modelos, sendo importante garantir que o dataset é representativo e, sempre que possível, balanceado entre as diferentes classes, de forma a evitar desvios no processo de aprendizagem.

2. Extração de características (*feature extraction*)

Após a preparação dos dados, procede-se à extração de características relevantes (*features*) que permitam representar a informação de forma adequada para os modelos. No caso de dados textuais, estas características podem incluir representações como *n-grams*, *Term Frequency–Inverse Document Frequency* (TF-IDF), ou *embeddings* semânticos como Word2Vec e GloVe.

Para além destas abordagens tradicionais, podem também ser consideradas características de natureza linguística e estilística, como POS *tags* ou a presença de figuras de estilo, como metáforas, que podem conter informação emocional implícita nas letras.

3. Treino do modelo

Nesta fase, os algoritmos de *Machine Learning* utilizam as *features* extraídas para aprender padrões que relacionam os dados de entrada com as respetivas classes. Dependendo da

abordagem utilizada, podem ser aplicados modelos tradicionais, como por exemplo, *Logistic Regression, Support Vector Machines* ou *Random Forest*.

O objetivo do treino é ajustar os parâmetros do modelo de forma a minimizar o erro nas previsões.

4. Validação e avaliação

Após o treino, é necessário avaliar o desempenho do modelo em dados não vistos, de forma a garantir a sua capacidade de generalização. Para isso, são utilizadas técnicas de comparação entre os dados reais e os previstos pelo modelo, como por exemplo matrizes de confusão e relatórios de classificação com métricas definidas.

5. Predição

Uma vez treinado e validado, o modelo pode ser utilizado para realizar previsões sobre novos dados. No contexto deste trabalho, isso corresponde à classificação automática de emoções em letras de músicas que não foram previamente analisadas.

No presente trabalho, estas etapas são aplicadas à tarefa de reconhecimento de emoções em músicas com base nas letras, com especial foco na exploração de características estilísticas relacionadas com a presença de metáforas.

2.4. Deep Learning

O *Deep Learning* é um subcampo da inteligência artificial que utiliza redes neuronais artificiais com múltiplas camadas para aprender representações complexas dos dados. Ao contrário de técnicas tradicionais de *Machine Learning*, o *Deep Learning* é capaz de extrair automaticamente características relevantes, reduzindo a necessidade de engenharia manual de *features*.

O seu funcionamento baseia-se nas seguintes etapas:

1. Preparação dos dados

Nesta fase, são recolhidos e organizados os dados que serão utilizados para treinar o modelo. No caso da identificação de metáforas em letras de músicas, este processo envolve a construção de um dataset composto por frases previamente rotuladas quanto à presença ou ausência de metáforas.

Posteriormente, o dataset é dividido em conjuntos de treino, validação e teste, permitindo avaliar a capacidade de generalização do modelo. Tal como nas abordagens tradicionais, é importante garantir que os dados estão bem distribuídos entre as diferentes classes, evitando desvios no processo de aprendizagem.

2. Construção e treino da rede neuronal

Nesta etapa, define-se a arquitetura do modelo, incluindo o número de camadas, os neurónios em cada camada, as funções de ativação (como *ReLU* ou *sigmoid*) e outros hiperparâmetros relevantes.

Durante o treino, o modelo ajusta os seus pesos com base no erro cometido nas previsões, recorrendo a algoritmos como *backpropagation* e a otimizadores como Adam ou *Stochastic Gradient Descent* (SGD). O objetivo é minimizar a função de perda e melhorar progressivamente o desempenho do modelo.

3. Avaliação e ajustes

Após o treino, o modelo é avaliado utilizando métricas como *accuracy, precision, recall, F1-score* e *F2-score*. A análise da matriz de confusão permite ainda compreender os tipos de erros cometidos.

Caso o desempenho não seja satisfatório, podem ser realizados ajustes na arquitetura, nos hiperparâmetros ou no próprio conjunto de dados, num processo iterativo de melhoria.

Modelos baseados em Transformers

Nos últimos anos, os modelos baseados na arquitetura de *Transformers* tornaram-se o estado da arte em tarefas de NPL. Estes modelos utilizam mecanismos de atenção (*attention*), que permitem analisar o contexto global de uma frase, captando relações entre palavras mesmo quando estão distantes no texto.

Uma das principais vantagens destes modelos é a possibilidade de utilização de versões pré-treinadas em grandes volumes de dados, que podem posteriormente ser adaptadas a tarefas específicas através de *fine-tuning*. Entre alguns dos modelos mais utilizados temos por exemplo o *Bidirectional Encoder Representations from Transformers,* BERT e o *Robustly Optimized* BERT *Pretraining Approach,* RoBERTa.

O BERT introduz uma abordagem bidirecional na análise do texto, enquanto o RoBERTa constitui uma versão otimizada deste modelo, apresentando melhorias no processo de treino e, frequentemente, melhores resultados.

No contexto deste trabalho, estes modelos são utilizados para a deteção automática de metáforas em letras de músicas, sendo exploradas abordagens com e sem *fine-tuning*. A utilização de *Transformers* permite obter representações mais ricas e contextualizadas do texto, contribuindo para um melhor desempenho na identificação de padrões linguísticos complexos associados às emoções.

2.5. Métricas de avaliação

De modo a compreender melhor quais as métricas de avaliação dos modelos que iremos utilizar, este subcapítulo irá descrevê-las. Desta forma, a escolha do melhor modelo para deteção de metáforas será facilitada.

Acurácia: É a divisão da quantidade de previsões corretas feitas, divididas pela quantidade de previsões totais feitas.

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN}$$

Precisão: É a relação entre os positivos corretamente identificados (verdadeiros positivos) e todos os positivos identificados.

$$Precisão = \frac{TP}{TP + FP}$$

Recall: Mede a capacidade do modelo de prever classes positivas reais. É a relação entre os verdadeiros positivos previstos e o que foi realmente marcado. A métrica de *recall* revela quantas das classes previstas estão corretas.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: Representa a média harmónica entre as métricas de Precisão e de Recall. A média harmónica é um tipo de média utilizada quando se necessita de representar valores que se comportam de maneira inversamente proporcional, como no caso do Precisão e Recall.

$$F1\ Score = 2 \times \frac{Precisão \times Recall}{Precisão + Recall}$$

F2-Score: Semelhante ao F1-Score, mas dá mais peso ao Recall do que à precisão

$$F2\ Score = (1 + 2^2) \times \frac{Precisão \times Recall}{(2^2 \times Precisão) + Recall}$$

No nosso caso, o objetivo seria diminuir as falsas previsões do modelo, tendo especial atenção aos falsos negativos, uma vez que é do nosso interesse que o modelo detete o máximo de metáforas possível. No entanto, não podemos esquecer que os falsos positivos também não podem ter um valor elevado, pois não queremos que o modelo classifique todas as frases como metáforas só para não falhar as que efetivamente são metáforas. Assim, dando mais peso à métrica *recall* que mede a capacidade do modelo de prever classes positivas reais, o nosso foco será escolher o modelo que tenha maior valor na métrica de F2-Score que pondera maioritariamente o *recall* mas também dá algum peso à precisão.

2.6. Figuras de estilo e metáforas

Uma figura de estilo é um tipo de recurso linguístico utilizado para tornar a frase mais expressiva com o propósito de aumentar a probabilidade de o leitor / ouvinte ser captado pela mensagem e tornar o texto mais apelativo à sua leitura. Daí serem tão importantes e comuns de utilizar na escrita de um poema ou de uma música.

Existem três tipos de figuras de estilo: **as figuras de sintaxe ou de construção**, como a anáfora (1), que alteram a construção gramatical; **as figuras de pensamento**, como por exemplo o eufemismo (2) que estão relacionadas com a ideia que a expressão transmite; e as **figuras de palavras**, que alteram o significado das palavras para que esta se adeque à mensagem que se pretende transmitir. É neste último tipo que se inserem as metáforas (3). Vejamos os seguintes exemplos para cada um dos tipos de figuras de estilo:

(1) "É urgente o amor. É urgente um barco no mar. É urgente destruir certas palavras." (Andrade, 1951)

(2) "Algum dali tomou perpétuo sono" (Camões, 1572)

(3) "Meu pensamento é um rio subterrâneo" (Pessoa, 1986)

Uma metáfora é uma figura de estilo onde se distorce o verdadeiro significado de uma palavra para que esta faça mais sentido na frase onde está a ser utilizada (Shutova et al., 2013). As metáforas são indispensáveis em letras de música que tendam a ter vertentes mais poéticas, na medida em que mostram a criatividade do autor, e aumentam o impacto e o carácter que a escrita tem no seu leitor, funcionando assim como ponte de linguagem e emoção transmitida (Kesarwani et al., 2017).

De acordo com o livro (Lakoff & Mark Johnson, 1980) existem três tipos de metáforas:

1. **Metáforas estruturais**: onde uma ideia é estruturada nos termos de outra
    a. Exemplo: *Time is money*
    b. Tradução: Tempo é dinheiro
2. **Metáforas orientacionais**: baseadas na orientação espacial das palavras
    a. Exemplo: *I'm feeling down.*
    b. Tradução: estou em baixo = deprimido
3. **Metáforas ontológicas**: atribui entidades a algo abstrato
    a. Exemplo: *The mind is a container*
    b. Tradução: A mente é um recipiente – de ideias

Existem ainda outros artigos e estudos que definem outros tipos de metáforas, como por exemplo (Nordquist, 2025):

- Metáforas conceptuais – nas quais um domínio é entendido nos termos de outro;
- Metáforas criativas – comparações originais que chamam à atenção tal como se fossem uma figura de linguagem
    - Exemplo: *I'm a satellite, I'm out of control*
        - letra da música "*Don't stop me now*" dos Queen;
    - Tradução: Sou um satélite, estou fora de controlo
- Metáforas mortas – Frases que perderam o seu efeito dado à sua utilização excessiva no discurso
    - Exemplos: "Cabeça do prego" e "Perna da mesa";
- Metáfora sinestésica (Basbaum, 2003) – quando juntam sensações de diferentes sentidos
    - Exemplos: "*harsh sound*" e "*sweet voice*"

Em relação à estrutura sintática (POS), podemos ainda dividir as metáforas em cinco tipos de acordo com (Kesarwani et al., 2017)

Para cada tipo de metáfora identificado, apresentamos um exemplo e a sua possível tradução para português de Portugal mantendo a sua construção frásica e sentido da mesma.

Tipo I: nome – verbo copulativo – (determinante –) nome

- Exemplo: "*All the world's a stage*" (Shakespeare, As You Like It, 1623)
- Tradução: O mundo é um palco

Tipo II: nome – verbo regular – nome

- Exemplo: "*Jealousy kills trust*"
- Tradução: A inveja mata a verdade

Tipo III: adjetivo – nome

- Exemplo: "*Sweet sorrow*" (Shakespeare, Romeo and Juliet, 1597)
- Tradução: Doce Tristeza

Tipo IV: nome – verbo

- Exemplo: "*My heart bleeds*"
- Tradução: O meu coração sangra

Tipo V: verbo – verbo

- Exemplo: "*To die – To sleep*" (Shakespeare, Hamlet, 1602)
- Tradução: Morrer – Dormir

3. Estado de Arte

Neste capítulo, apresentamos o panorama atual em relação aos métodos e técnicas utilizados na identificação de emoções em músicas a partir da análise das letras (*text mining*). Começamos por apresentar os resultados anteriores de deteção de emoções através das letras, áudio e bimodal, de seguida apresentamos os resultados de algumas técnicas de processamento de texto para deteção de metáforas, e, por fim, expomos algumas estratégias de segmentação de texto e quais os problemas que isso pode trazer para o nosso projeto.

3.1. Music Emotion Recognition

Diversos estudos na área de MER têm evidenciado a importância da utilização de *features* adequadas e de *datasets* bem estruturados para a melhoria do desempenho dos modelos de classificação emocional.

Ao nível do **áudio**, (Panda et al., 2020) propõem a identificação de novas *features* emocionalmente relevantes, tendo para esse efeito desenvolvido um *dataset* composto por 900 excertos de áudio anotados segundo o modelo de valência-ativação de Russell. Os resultados obtidos demonstram que a inclusão destas novas *features* contribui positivamente para o desempenho do modelo, atingindo um F1-score de 76,4% na tarefa de deteção de emoções.

No domínio específico das **letras**, (Malheiro, 2017) apresenta uma análise detalhada de diferentes tipos de *features* linguísticas, incluindo *content-based*, *stylistic* e *semantic features*, demonstrando que a combinação destas abordagens permite alcançar melhores resultados na classificação emocional. Neste estudo, foi obtido um F1-score de 82,7% com um conjunto selecionado de 388 *features*.

Adicionalmente, no contexto de abordagens **bimodais**, que integram simultaneamente informação proveniente do áudio e das letras, o mesmo autor reporta uma melhoria significativa no desempenho, atingindo um F1-score de 88,4% com 1057 *features* selecionadas. Estes resultados evidenciam o potencial da combinação de diferentes fontes de informação na modelação de emoções em música.

3.2. Deteção de Metáforas

Existem vários estudos sobre quais as figuras de estilo mais aplicadas em letras de músicas, por exemplo, em (Tambun & Sianipar, 2014) onde foram identificadas quais as figuras de estilo mais utilizadas nas músicas dos Westlife, sendo os resultados os seguintes: comparação (27,71%), metáfora (21,70%), personificação (1,20%), sinédoque (1,20%), metonímia (7,23%), símbolo (4,81%), hipérbole (33,75%) e paradoxo (2,40%). São imensas as figuras de estilo que um escritor pode optar por utilizar e, por essa mesma razão, é difícil enumerar apenas uma que seja a mais utilizada em letras de música. No entanto, vejamos alguns exemplos onde a metáfora foi a figura de estilo predominante.

Dhea Oktavia realizou um estudo (Oktavia, 2023) sobre as letras das músicas contidas em quatro álbuns da banda "Tame Impala". Os resultados comprovam que as figuras de estilo mais encontradas foram metáforas (38%), comparação (21%) e personificação (19%).

Outros exemplos de músicas que são escritas em torno de uma metáfora são: "Firework" da Katy Perry onde o fogo de artifício é comparado com o poder e a capacidade de cada um de nós para brilhar e dar o nosso melhor na nossa vida; "Stairway to Heaven" dos Led Zeppelin onde esta escada até ao céu é uma metáfora para a jornada da vida e dos seus mistérios após a morte (Hughes, 2024).

Para a deteção de metáforas, identificamos duas técnicas que podemos e iremos utilizar neste projeto: técnicas clássicas e de certa forma manuais, e técnicas modernas e mais autónomas de *Machine Learning*.

3.2.1. Técnicas clássicas

Dentro dos métodos clássicos que iremos utilizar para deteção de metáforas, existem dois tipos:

Léxicos de metáforas

Os léxicos de metáforas são recursos linguísticos que relacionam expressões metafóricas com os seus significados literais e figurados. Estes recursos permitem estruturar e sistematizar o conhecimento sobre metáforas, facilitando a sua identificação automática. Trabalhos como o de (Shutova et al., 2010) demonstram a utilidade de abordagens baseadas em conhecimento linguístico e bases lexicais para identificar metáforas através de

similaridade semântica. No entanto, estas abordagens apresentam limitações ao dependerem fortemente de recursos pré-definidos e da cobertura dos léxicos utilizados.

Métodos baseados em regras

As técnicas baseadas em regras utilizam padrões linguísticos previamente definidos, como estruturas sintáticas ou relações gramaticais, para identificar figuras de estilo, como as metáforas. Por exemplo, o método proposto por (Fass, 1991) utiliza restrições semânticas e relações estruturais para distinguir entre uso literal e figurado da linguagem.

A literalidade é reconhecida quando todas as preferências contextuais são satisfeitas, ou seja, quando não ocorre qualquer violação das restrições semânticas associadas aos verbos e seus argumentos. A metonímia é identificada através da aplicação de um conjunto de regras de inferência que estabelecem uma relação semântica conhecida entre uma fonte e um alvo. Uma inferência bem-sucedida permite reinterpretar a expressão e prosseguir a análise.

A metáfora, por sua vez, é detetada quando existe simultaneamente uma violação de preferência e uma analogia relevante entre os conceitos envolvidos. Esta analogia baseia-se numa correspondência estrutural entre elementos dos *frames* semânticos (designados por "células irmãs"), permitindo estabelecer uma relação de similaridade subjacente ao uso figurado.

Apesar da sua relevância histórica, este tipo de abordagem apresenta algumas limitações importantes. Em particular, depende fortemente de conhecimento linguístico previamente codificado e de regras manuais, o que dificulta a sua escalabilidade e adaptação a diferentes domínios ou línguas. Além disso, a capacidade de generalização é limitada, especialmente em contextos onde o significado figurado depende de nuances pragmáticas ou culturais mais complexas. Ainda assim, o trabalho de (Fass, 1991) estabelece as bases para o desenvolvimento de abordagens posteriores na deteção automática de metáforas, influenciando tanto métodos baseados em regras como técnicas mais modernas de aprendizagem automática.

Mais recentemente, (Krishnakumaran & Zhu, 2007) aplicaram padrões baseados em regras e recursos lexicais para identificar metáforas, obtendo resultados promissores em textos simples.

Para frases do tipo I, o algoritmo verifica se o sujeito e o objeto têm uma relação de hipónimo no *WordNet*. Caso contrário, a relação é classificada como metafórica. Esta abordagem

atinge 70% de precisão e 61% de *recall* num conjunto de teste. Para relações tipo II e III, o algoritmo utiliza hipónimos/hiperónimos do *WordNet* juntamente com estatísticas de coocorrência de *bigramas* provenientes de um grande corpus da web. Isto permite detetar usos metafóricos novos que não estão cobertos apenas pelo *WordNet*.

Contudo, estes métodos são geralmente limitados pela sua rigidez, apresentando dificuldades em capturar metáforas implícitas ou contextualmente complexas.

3.2.2. Técnicas modernas

As técnicas modernas de deteção de metáforas baseiam-se essencialmente em abordagens de *Machine Learning* e *Deep Learning*, que permitem automatizar o processo de identificação de linguagem figurada a partir de dados anotados. Estas metodologias exploram diferentes tipos de *features* linguísticas e representações semânticas, evoluindo desde modelos clássicos, que dependem de engenharia manual de características, até modelos mais recentes baseados em *Transformers*, capazes de capturar relações contextuais complexas. Nesta secção são apresentadas as principais abordagens modernas utilizadas na literatura para a deteção de metáforas.

Machine Learning

Modelos clássicos de *Machine Learning*, como *Support Vector Machines* (SVM), *Random Forest* ou *Naïve Bayes*, são utilizados na deteção de metáforas quando se dispõe de dados anotados. Recorrendo a *features* linguísticas como etiquetas gramaticais (POS tags), dependências sintáticas ou recursos lexicais, estes algoritmos aprendem a detetar metáforas mais simples. Embora úteis, requerem engenharia manual de *features* e não captam facilmente o uso figurado implícito nas expressões metafóricas.

Com a disponibilidade de dados anotados, modelos clássicos de *Machine Learning* passaram a ser utilizados na deteção de metáforas. Por exemplo, (Tsvetkov et al., 2014) propõem um modelo baseado em *Random Forest* que combina *features* linguísticas e semânticas, alcançando melhorias significativas face a abordagens baseadas apenas em regras. O modelo utiliza características conceptuais, como abstractividade e representações vetoriais, e demonstra que é possível distinguir eficazmente usos literais e metafóricos. Em termos de resultados, o sistema atinge desempenho de estado da arte em inglês (até cerca de 82% de

acurácia em relações sujeito-verbo-objeto e 86% em adjetivo-substantivo), além de apresentar boa generalização para novos domínios (F1-Score ≈ 76%). Um dos contributos mais relevantes é a capacidade de transferência entre línguas: treinado apenas em inglês, o modelo obtém resultados consistentes em espanhol, russo e farsi, com F1-scores entre aproximadamente 72% e 84% . Estes resultados sustentam a ideia de que as metáforas têm uma natureza conceptual e podem ser captadas por representações semânticas independentes da língua. No entanto, estes métodos continuam dependentes de engenharia manual de *features* e têm dificuldade em capturar relações semânticas profundas.

Deep Learning **e modelos baseados em** ***Transformers***

As técnicas de *Deep Learning* têm-se revelado particularmente eficazes na deteção de metáforas, principalmente devido à sua capacidade de capturar relações semânticas complexas e dependências de longo alcance no texto.

Modelos baseados em *transformers*, como BERT e *XLNet*, geram *embeddings* contextualizados para cada palavra, ou seja, representações vetoriais que incorporam não apenas o significado isolado da palavra, mas também o contexto em que ocorre. Esta contextualização é fundamental para a deteção de metáforas, uma vez que o significado metafórico depende fortemente da relação entre palavras vizinhas.

No artigo de (Wallace et al., 2020) as representações extraídas destes modelos são combinadas e, por vezes, enriquecidas com informação gramatical, como POS tags. Esses vetores são depois processados por redes neurais recorrentes, como *Bidirectional Long Short-Term Memory* (Bi-LSTM), capazes de analisar o texto em ambas as direções (do início para o fim e vice-versa) e capturar padrões semânticos e sintáticos relevantes para distinguir linguagem literal de linguagem figurada.

De forma complementar, (Tanasescu et al., 2018) demonstram que abordagens de *Deep Learning* baseadas em redes neuronais convolucionais (CNN) também apresentam melhorias significativas na deteção de metáforas, especialmente em contexto poético. Os resultados mostram que o modelo CNN atinge um *F1-score* de 83,3% para a classe metafórica e 74,4% para a classe literal, superando modelos tradicionais como SVM, que obtêm cerca de 78,1%.

A combinação de *embeddings* de diferentes modelos pré-treinados, aliada à capacidade das redes neuronais para explorar dependências contextuais, tem demonstrado superar abordagens tradicionais, na tarefa de deteção de metáforas.

3.3. Segmentação de Texto em Letras de Música

A segmentação de texto é uma etapa importante no processamento de letras de música, uma vez que influencia a forma como a informação é analisada posteriormente. No entanto, este processo apresenta desafios, sobretudo devido à ausência de pontuação e à estrutura não convencional das letras. Assim, nesta secção são apresentados os principais problemas associados à segmentação e algumas estratégias utilizadas para os ultrapassar.

3.3.1. Problema da ausência de pontuação

As letras de música frequentemente são disponibilizadas sem pontuação, nomeadamente em websites de letras ou plataformas de partilha de texto. Esta falta de sinais de pontuação prejudica as técnicas automáticas de análise textual, como a segmentação em frases ou orações, e pode desviar métricas linguísticas e quantitativas (como as de coerência, lexicais e sintáticas).

A ausência de pontuação em letras de música tem sido abordada na literatura através de diferentes estratégias complementares no âmbito do Processamento de Linguagem Natural. Por um lado, abordagens de restauração automática de pontuação, baseadas em modelos de linguagem como BERT *e* RoBERTa, têm demonstrado elevada eficácia na inserção de sinais de pontuação em texto contínuo, melhorando tarefas subsequentes como a segmentação e análise sintática, conforme evidenciado por estudos como (Avram, Păiş, & Tufiş, 2021). Por outro lado, têm emergido métodos de segmentação independentes de pontuação, que recorrem à informação semântica e aprendizagem automática para identificar fronteiras discursivas, como proposto por (Minixhofer et al., 2023), revelando maior robustez em textos não estruturados.

3.3.2. Estratégias para segmentação:

Devido à estrutura não convencional das letras de música, a segmentação do texto pode ser realizada através de diferentes estratégias, cada uma com vantagens e limitações. Estas

abordagens variam desde métodos simples, baseados na estrutura das letras, até técnicas mais avançadas que recorrem à informação semântica e a modelos de aprendizagem automática. Nesta secção são apresentadas as principais estratégias utilizadas para segmentar letras de música de forma eficaz.

1. Por linha ou por estrofe

A segmentação mais básica recorre à estrutura intrínseca das letras em linhas ou estrofes: cada linha é tratada como uma unidade independente, ou cada estrofe representa um segmento coeso. Embora simples de implementar, esta abordagem não garante que cada fragmento corresponda a uma unidade linguística significativa (como frases completas), podendo misturar ideias distintas ou interromper frases no meio.

2. Baseada em *embeddings* e similaridade semântica

Uma abordagem mais completa utiliza técnicas de representação semântica: cada linha ou segmento é convertido num vetor de *embedding* (por exemplo, usando modelos *Transformers*), e segmentos semanticamente semelhantes são agrupados ou delimitados com base na proximidade desses vetores em espaço vetorial.

3. Modelos de pontuação automática

Para facilitar na segmentação, poderíamos utilizar alguns modelos de pontuação automática. Existem alguns modelos de inteligência artificial como o *Trinka AI* que analisa os textos e corrige a sua gramática e linguagem. Um outro programa que também analisa a pontuação é o Word. No entanto, para letras de música, ele assume a sua estrutura como sendo a correta, portanto não iria apresentar melhorias de pontuação. O melhor método a utilizar seria modelos como o BERT que analisa o contexto da frase e, assim, realiza correções mais precisas.

Esta página foi intencionalmente deixada em branco.

4. Bases de Dados

Para a elaboração deste projeto, foram utilizados três bases de dados, que serão descritas neste capítulo.

4.1. Treino do Modelo para deteção de metáforas

Para a deteção de metáforas, foi necessário criar modelos de *Machine Learning*, que fossem treinados com um conjunto de dados de frases categorizadas com e sem metáforas, de modo a ser balanceado. Depois de pesquisar em artigos, passamos a enumerar alguns dos datasets encontrados.

4.1.1. VUA Metaphor Corpus

(Krennmayr & Steen, 2017) Contém frases anotadas manualmente com metáforas de diversos domínios. As frases foram retiradas de textos académicos, ficção e jornalismo. O objetivo deste dataset era perceber quais metáforas são utilizadas em que contexto de discurso e qual o seu propósito.

Exemplos:

a) "*he's like a ferret*"
    a. Tradução: "Ele é como um furão"
b) "*Naturally, to embark on such as step is not necessarily to succeed in realizing it*"
    a. Tradução: "Naturalmente, embarcar num processo não significa necessariamente ter sucesso ao realizá-lo"

4.1.2. TroFi Dataset

O corpus criado por (Birke, 2005) contém frases que incluem 50 palavras alvo (maioritariamente verbos). Sob cada palavra-alvo, é apresentado primeiro o agrupamento não literal e depois o agrupamento literal. A primeira coluna da base de dados contém um

identificador que liga a frases ao ficheiro fonte. A segunda coluna contém um de três rótulos referentes ao estado da anotação humana: L (Literal), N (Não literal) ou U (Não anotado). A

Figura 5 apresenta um exemplo para a palavra "*absorb*".

```
***absorb***
*nonliteral cluster*
wsj02:2251  U   Another option will be to try to curb the growth in education and other local assistance , which absorbs 66 %
of the state 's budget ./.
wsj03:2839  N   But in the short-term it will absorb a lot of top management 's energy and attention , '' says Philippe Haspe
slagh , a business professor at the European management school , Insead , in Paris ./.
*literal cluster*
wsj11:1363  L   An Energy Department spokesman says the sulfur dioxide might be simultaneously recoverable through the use of
powdered limestone , which tends to absorb the sulfur ./.
```

Figura 5 - Exemplo para a palavra "absorb" do corpus TroFi Dataset

4.1.3. MOH-X Dataset

Na base de dados elaborada por (Babieno et al., 2022), foram recolhidas definições de dicionários de forma totalmente automática, para depois recuperarem os sentidos não literais dos verbos, facilitando na deteção de metáforas. Todos os verbos neste Dataset têm vários significados, onde pelo menos um deles é metafórico.

Este método pode ser utilizado para deteção de metáforas, mas apenas ao nível do token.

Exemplos de alguns verbos contidos no Dataset:

a) Grasp
    a. Literal → agarrar fisicamente
    b. metafórico → compreender
        i. She grasped the concept → metafórico
b) Attack
    a. literal → atacar fisicamente
    b. metafórico → criticar
        i. He attacked the argument → metafórico

4.1.4. LCC Metaphor Datasets

O corpus criado por (Mohler et al., 2016) apresenta conjuntos de dados de metáforas anotadas pela LCC (*Language Computer Corporation*). Disponibiliza classificações de metaforicidade para pares de palavras dentro da frase, numa escala de quatro pontos e classificações da polaridade afetiva e da intensidade para cada par de palavras (alvo-fonte). Para além das 188741 anotações em inglês, contém um vasto conjunto de metáforas prováveis, extraídas de forma independente pelos sistemas de deteção de metáforas, mas que não foram analisadas pelos anotadores.

4.1.5. Poetry anotation corpus

Os autores (Kesarwani et al., 2017) construíram um corpus de poesia personalizado, uma vez que não existia nenhum conjunto de dados de poesia anotado para metáforas disponível publicamente. Recolheram 12 830 poemas da *Poetry Foundation* e selecionaram automaticamente frases que correspondiam a cinco padrões de etiquetas gramaticais (POS) associados a metáforas (tipo I: Nome–Verbo–Nome com verbo copulativo, estendido para Nome–Verbo–Det–Nome e tipos II–V com outros padrões)

A partir das frases extraídas, foram identificadas cerca de 1 500 potenciais instâncias do tipo I, das quais 720 foram anotadas manualmente por dois anotadores independentes:

- "y" para metáfora
    - Exemplo: "*where books were trees"*
    - Tradução: Onde os livros eram árvores
- "n" para não metáfora
    - Exemplo: *"in the field is a house"*
    - Tradução: No campo está uma casa
- "s" para ambíguo
    - Exemplo: *"not only underground are the brains of men"*
    - Tradução: Nem só debaixo da terra estão os cérebros dos homens

Este foi o escolhido para utilizar neste projeto.

4.2. Identificação de emoções em letras de músicas

Para a segunda parte do projeto (identificação de emoções), foram utilizados dois conjuntos de dados distintos: um para a tarefa de deteção de metáforas ao nível da frase e outro para a classificação de emoções ao nível da música.

4.2.1. Dataset com metáforas por quadrante

Esta base de dados foi elaborada com base no *Poetry corpus* mencionado no capítulo anterior. Porém, o dataset continha maioritariamente metáforas do tipo I, por isso, teve de ser alterado ligeiramente para vencer o problema do balanceamento.

Para cada tipo de estrutura de metáfora esta base de dados tem 50 frases anotadas com metáfora e 50 frases literais – total de 100 frases por tipo de estrutura de metáfora. Para modelos que exigem *fine-tuning* (outra camada de treino), adicionamos outras frases que foram anotadas manualmente, através da pesquisa em livros, frases do dia a dia e também com ajuda da inteligência artificial. O corpus final para o *fine-tuning* é constituído por 644 frases: 74 do tipo I, 63 do tipo II, 64 do tipo III, 60 do tipo IV e 60 do tipo V. Contém ainda 323 frases literais.

4.2.2. Extração de features com letras de músicas

O *dataset* utilizado é composto por 180 letras de músicas, selecionadas de forma a representar diferentes emoções e estilos musicais. Cada instância corresponde a uma música, sendo caracterizada por um conjunto de atributos que incluem informação descritiva, emocional e textual.

Em particular, o *dataset* integra variáveis como o título da música, o artista, um identificador único e a fonte da letra. Para além disso, inclui métricas emocionais previamente calculadas, nomeadamente a valência e ativação média, bem como os respetivos desvios padrão. Com base nestes valores, é atribuído a cada música um quadrante emocional, de acordo com o modelo de Russell. A componente textual do *dataset* é representada pela coluna *Lyrics*, que contém a letra completa de cada música.

Este conjunto de dados foi utilizado como base para a extração de *features* relacionadas com a presença de metáforas nas letras, nomeadamente a sua ocorrência, frequência e distribuição ao longo do texto. O principal objetivo da utilização deste *dataset* consiste em analisar de que forma as metáforas contribuem para a expressão emocional nas músicas. Para tal, as *features* extraídas foram posteriormente integradas com outras variáveis relevantes, previamente definidas no âmbito do projeto e explicadas no capítulo seguinte.

4.2.3. Features para identificação de emoções

Para a identificação do quadrante emocional das músicas, encontra-se previamente definido um conjunto de *features* estabelecido pela equipa do projeto MERGE. Estas incluem características relacionadas com o conteúdo, estilo, estrutura e semântica das letras, com o objetivo de capturar diferentes dimensões linguísticas relevantes para a deteção de emoções. O conjunto de *features* encontra-se organizado num *dataset* associado às 180 letras referidas no subcapítulo anterior.

O principal objetivo deste trabalho consiste na integração de *features* adicionais relacionadas com metáforas, de forma a avaliar o seu contributo e relevância na tarefa de deteção de emoções a partir das letras das músicas. Estas novas *features* incluem indicadores como a presença de metáforas na letra, a sua frequência e a sua densidade em relação ao número de segmentos textuais.

Após a integração destas *features*, foi construído um conjunto final que combina informação linguística tradicional com características estilísticas associadas às metáforas. Este conjunto foi utilizado como entrada em diferentes modelos de classificação, com o objetivo de identificar o quadrante emocional ao qual cada música pertence.

Desta forma, pretende-se avaliar o impacto das metáforas na tarefa de reconhecimento de emoções em música, comparando o desempenho dos modelos em dois cenários distintos: com e sem a inclusão das *features* associadas às metáforas. Esta comparação permite analisar se a informação adicional introduzida contribui para melhorar a capacidade dos modelos em capturar a dimensão emocional das letras.

Esta página foi intencionalmente deixada em branco.

5. Identificação de metáforas

Neste capítulo iremos descrever todo o processo de preparação e construção de modelos de *Machine* e *Deep Learning* para deteção de metáforas. Iremos descrever cada um dos modelos, a sua construção e que parâmetros foram utilizados.

5.1. Preparação dos dados

A preparação dos dados constitui uma etapa fundamental no desenvolvimento de modelos de *Machine Learning*, uma vez que a qualidade e organização do conjunto de dados influenciam diretamente o desempenho dos modelos.

O *dataset* utilizado neste trabalho foi carregado a partir de um ficheiro em formato Excel, contendo frases extraídas de letras de músicas, previamente anotadas quanto à presença de metáforas. Cada instância é composta por uma frase (*phrase*) e um rótulo associado (*label*), indicando se a frase contém ou não uma metáfora. Adicionalmente, o *dataset* inclui uma coluna com o tipo de metáfora (*Type*).

Numa fase inicial, os rótulos categóricos foram convertidos para valores numéricos, de forma a serem utilizados pelos algoritmos de classificação. Especificamente, o valor "y" (presença de metáfora) foi mapeado para 1, enquanto o valor "n" (ausência de metáfora) foi mapeado para 0.

Posteriormente, foi realizada a transformação dos dados textuais em representações numéricas, adequadas para a aplicação de modelos de *Machine Learning*. Para tal, foi utilizada a técnica TF-IDF, que permite representar cada frase com base na importância relativa das palavras no conjunto de dados. Foram considerados *n-grams* de tamanho 1 e 2, permitindo capturar não apenas palavras isoladas, mas também combinações de palavras relevantes para a deteção de metáforas.

Adicionalmente, foi analisado o impacto da remoção de *stopwords* no desempenho dos modelos. As *stopwords* correspondem a palavras muito frequentes na língua, como artigos e preposições, que nem sempre contribuem para a distinção entre classes. Assim, foram

testadas duas abordagens: uma com remoção de *stopwords* e outra mantendo o texto original, de forma a avaliar empiricamente qual das opções produzia melhores resultados.

Por fim, os dados foram organizados em matrizes de características (*features*) e rótulos (*labels*), constituindo o conjunto de entrada para o treino e avaliação dos modelos de classificação.

5.2. Modelos Tradicionais

Os modelos tradicionais desenvolvidos foram: Regressão Logística, *Random Forest Classifier, Support Vector Classifier e XGBClassifier*. Para cada um deles, analisamos a importância das stopwords. No entanto, como a metáfora depende também do contexto, aqui essa regra pode não fazer sentido.

5.2.1. Regressão Logística

A Regressão Logística é um algoritmo de classificação linear amplamente utilizado em problemas de classificação binária. Este modelo estima a probabilidade de pertença a uma determinada classe com base numa combinação linear das *features*, sendo particularmente adequado para dados representados em espaços de elevada dimensionalidade, como é o caso de representações TF-IDF.

Neste trabalho, a Regressão Logística foi aplicada ao conjunto de dados previamente descrito, utilizando como entrada as *features* textuais extraídas das frases, nomeadamente a representação TF-IDF com *n-grams* de tamanho 1 e 2.

Relativamente à configuração do modelo, foi utilizado o parâmetro max_iter=100, que define o número máximo de iterações do algoritmo de otimização. Este valor foi considerado suficiente para garantir a convergência do modelo, tendo em conta a dimensão do *dataset*.

Para a avaliação do desempenho, foi adotada uma estratégia de validação cruzada estratificada (*Stratified K-Fold*) com 5 partições (*n_splits=5*). Esta abordagem permite garantir que a proporção das classes é preservada em cada subconjunto, assegurando uma

avaliação mais robusta e reduzindo o risco de *overfitting*. Adicionalmente, foi utilizado o parâmetro shuffle=True, de forma a "baralhar" os dados antes da divisão, e random_state=42, garantindo a reprodutibilidade dos resultados.

5.2.2. Random Forest Classifier

O *Random Forest* é um algoritmo de aprendizagem supervisionada baseado em conjuntos (*ensemble learning*), que combina múltiplas árvores de decisão para realizar tarefas de classificação. O seu funcionamento assenta na construção de várias árvores treinadas sobre diferentes subconjuntos dos dados, sendo a decisão final obtida por votação maioritária. Esta abordagem tende a reduzir o risco de *overfitting* associado a árvores de decisão isoladas e permite capturar relações não lineares entre as *features*.

No presente trabalho, o modelo *Random Forest* foi aplicado ao conjunto de dados de deteção de metáforas, utilizando como entrada as *features* textuais obtidas através da vetorização TF-IDF das frases. Esta representação permitiu converter o texto em variáveis numéricas adequadas ao treino do modelo.

Relativamente à configuração, foi utilizado um modelo com n_estimators=100, o que significa que a "floresta" é composta por 100 árvores de decisão. Este valor foi escolhido por constituir uma configuração equilibrada entre desempenho e custo computacional. Adicionalmente, foi definido random_state=42, com o objetivo de garantir a reprodutibilidade dos resultados.

Tal como nos restantes modelos, a avaliação foi realizada através de validação cruzada estratificada com 5 partições. Esta estratégia permite dividir o conjunto de dados em subconjuntos de treino e teste, preservando a proporção entre classes em cada partição. Para além disso, foi utilizado shuffle=True, de forma a baralhar os dados antes da divisão, reduzindo possíveis efeitos relacionados com a sua ordem original.

A aplicação da função de validação cruzada permitiu obter previsões para todas as instâncias do dataset, sempre com base em subconjuntos de teste não observados durante o treino. A partir dessas previsões, foram posteriormente calculadas métricas de avaliação como precision, recall, F1-score e F2-score, bem como a matriz de confusão.

5.2.3. Support Vector Classifier (SVC)

O *Support Vector Classifier* (SVC) é um algoritmo de classificação supervisionada que procura encontrar o hiperplano ótimo que separa as diferentes classes no espaço das *features*, maximizando a margem entre os exemplos mais próximos de cada classe. Este modelo é particularmente eficaz em espaços de elevada dimensionalidade, como é o caso de representações textuais baseadas em TF-IDF.

No presente trabalho, foi utilizada uma versão do SVC com *kernel* linear, assumindo que as classes são aproximadamente separáveis de forma linear no espaço das variáveis. Esta escolha é adequada para dados textuais, onde a combinação de *features* pode frequentemente ser discriminada através de fronteiras lineares.

Relativamente à configuração do modelo, foi utilizado o parâmetro *kernel*="linear", que define o tipo de função de decisão, e *probability*=False, indicando que o modelo não calcula probabilidades associadas às previsões, mas apenas as classes. Esta opção permite reduzir o custo computacional, tornando o modelo mais eficiente. Tal como nos restantes modelos, foi definido *random_state*=42, garantindo a reprodutibilidade dos resultados.

A avaliação do modelo foi realizada através de validação cruzada estratificada com 5 partições (*n_splits=5*), assegurando que a distribuição das classes é preservada em cada subconjunto. Foi igualmente utilizado *shuffle=True*, de forma a tornar aleatória os conjuntos dos dados antes da divisão. A aplicação desta estratégia permitiu obter previsões para todas as instâncias do *dataset*, com base em subconjuntos de teste independentes, possibilitando uma avaliação mais robusta do desempenho do modelo.

5.2.4. XGBClassifier

O *XGBoost* (*Extreme Gradient Boosting*) é um algoritmo de aprendizagem supervisionada baseado em técnicas de *boosting*, que constrói modelos preditivos através da combinação sequencial de múltiplas árvores de decisão. Ao contrário de métodos como o *Random Forest*, onde as árvores são treinadas de forma independente, o XGBoost ajusta cada nova árvore com o objetivo de corrigir os erros cometidos pelas anteriores, recorrendo ao gradiente da

função de perda. Esta abordagem permite obter modelos mais precisos e robustos, especialmente em problemas de classificação.

No presente trabalho, o modelo XGBoost foi aplicado ao conjunto de dados de deteção de metáforas, utilizando como entrada as *features* textuais representadas através de TF-IDF. Esta combinação permite explorar tanto a capacidade do modelo em lidar com relações não lineares como a riqueza da representação textual.

Relativamente à configuração, foi utilizado o parâmetro *use_label_encoder=False*, desativando o codificador de etiquetas interno do XGBoost. Esta opção é recomendada nas versões mais recentes da biblioteca, evitando avisos de compatibilidade e não afetando o desempenho do modelo. Foi também definido *eval_metric='logloss'*, que corresponde à perda logarítmica, uma métrica adequada para problemas de classificação binária, uma vez que penaliza fortemente previsões incorretas com elevada confiança e incentiva uma melhor calibração das probabilidades.

Adicionalmente, foi utilizado o parâmetro *random_state*=42, garantindo a reprodutibilidade dos resultados, e *verbosity*=0, com o objetivo de suprimir mensagens de saída durante o treino, tornando o processo mais limpo e legível, especialmente em cenários de validação cruzada.

Tal como nos restantes modelos, a avaliação foi realizada através de validação cruzada estratificada com 5 partições, assegurando a preservação da distribuição das classes em cada subconjunto. A utilização desta estratégia permite obter previsões para todas as instâncias do dataset com base em dados não observados durante o treino, proporcionando uma avaliação mais robusta do desempenho do modelo.

5.3. Modelos de Deep Learning

Os modelos com tecnologia *Deep Learning* elaborados foram: BERT *e* RoBERTa com e sem *fine-tuning*

5.3.1. BERT

O modelo BERT é um modelo de *Deep Learning* baseado em arquiteturas *Transformer*, pré-treinado em grandes volumes de texto. Este modelo permite obter representações contextuais das palavras, captando relações semânticas complexas no texto.

SEM FINE-TUNING

Neste trabalho, foi utilizada a versão pré-treinada *bert-base-uncased*, sem qualquer ajuste adicional (*fine-tuning*), com o objetivo de extrair representações vetoriais (*embeddings*) das frases. Estas representações foram posteriormente utilizadas como *features* de entrada para um modelo de classificação tradicional.

A extração de *embeddings* foi realizada através da camada final do modelo BERT, utilizando o vetor associado ao token especial [CLS], que representa semanticamente toda a frase. Esta abordagem permite obter uma representação densa e contextualizada do texto, adequada para tarefas de classificação.

Após a obtenção dos *embeddings*, foi utilizado um classificador *Random Forest* para realizar a tarefa de deteção de metáforas. Esta combinação permite explorar a capacidade do BERT em capturar informação semântica, enquanto o classificador tradicional realiza a separação entre classes.

A avaliação do modelo foi realizada através de validação cruzada estratificada com 5 partições, garantindo a preservação da distribuição das classes em cada subconjunto. Esta estratégia permite uma avaliação mais robusta do desempenho, utilizando sempre dados não observados durante o treino.

Adicionalmente, foram analisadas métricas de desempenho como *precision*, *recall*, *F1-score* e *F2-score*, bem como a matriz de confusão, permitindo uma avaliação detalhada da capacidade do modelo em identificar corretamente frases metafóricas.

COM FINE-TUNING

Nesta abordagem, o modelo BERT foi adaptado especificamente à tarefa de deteção de metáforas através de um processo de *fine-tuning*. Ao contrário da abordagem anterior, onde o modelo é utilizado apenas como extrator de *features*, o *fine-tuning* permite ajustar os pesos do modelo às características específicas do *dataset*.

O modelo utilizado foi novamente o *bert-base-uncased*, ao qual foi adicionada uma camada de classificação (*classification head*) para distinguir entre frases com e sem metáfora. Os dados foram previamente processados através de tokenização, incluindo truncagem e preenchimento (*padding*) para um comprimento máximo de 128 tokens, garantindo uniformidade na entrada do modelo.

O treino foi realizado utilizando uma estratégia de validação cruzada estratificada com 5 partições, permitindo avaliar o modelo de forma robusta ao longo de diferentes subconjuntos de dados. Em cada iteração, o modelo foi treinado durante 3 épocas, utilizando o otimizador AdamW com uma taxa de aprendizagem de 2e-5. Foi também aplicado um *scheduler* linear para ajuste dinâmico da taxa de aprendizagem ao longo do treino.

Esta abordagem permite que o modelo aprenda diretamente padrões linguísticos associados à presença de metáforas, explorando a sua capacidade de modelar relações contextuais complexas no texto.

Tal como nos restantes modelos, o desempenho foi avaliado com base em métricas como *precision*, *recall*, *F1-score* e *F2-score*, bem como através da análise da matriz de confusão.

5.3.2. RoBERTa

O modelo RoBERTa é uma evolução do modelo BERT, baseada na mesma arquitetura *Transformer*, mas otimizada ao nível do processo de pré-treino, nomeadamente através da utilização de maiores volumes de dados e estratégias de treino mais eficientes. Estas melhorias permitem obter representações linguísticas mais robustas e com melhor capacidade de generalização.

SEM FINE-TUNING

Neste trabalho, foi utilizada a versão pré-treinada RoBERTa-base, sem aplicação de *fine-tuning*, com o objetivo de extrair representações vetoriais (*embeddings*) das frases. À semelhança da abordagem adotada com o BERT, o modelo foi utilizado exclusivamente como extrator de *features*, sendo posteriormente combinado com um classificador tradicional.

A extração dos *embeddings* foi realizada a partir da última camada do modelo, utilizando o vetor correspondente ao *token* inicial da sequência (equivalente ao *token* [CLS] no BERT), que sintetiza a informação semântica global da frase. Esta abordagem permite obter representações densas e contextualizadas do texto, adequadas para tarefas de classificação.

Os *embeddings* obtidos foram utilizados como entrada para um classificador *Random Forest*, responsável por realizar a distinção entre frases com e sem metáforas. Esta combinação tira partido da capacidade do modelo RoBERTa em capturar relações semânticas complexas, mantendo simultaneamente a simplicidade e eficiência de um classificador tradicional.

Tal como nos restantes modelos, a avaliação foi realizada através de validação cruzada estratificada com 5 partições, assegurando a preservação da distribuição das classes em cada subconjunto. Esta metodologia permite obter uma estimativa mais robusta do desempenho do modelo, utilizando sempre dados não observados durante o treino.

O desempenho foi avaliado com base em métricas como *precision*, *recall*, *F1-score* e *F2-score*, bem como através da análise da matriz de confusão.

COM FINE-TUNING

Nesta abordagem, o modelo RoBERTa foi adaptado especificamente à tarefa de deteção de metáforas através de um processo de *fine-tuning*. Ao contrário da utilização do modelo apenas como extrator de *embeddings*, esta estratégia permite ajustar os pesos do modelo ao domínio específico das letras de músicas e às características linguísticas associadas à presença de metáforas.

Foi utilizada a versão pré-treinada *RoBERTa-base*, à qual foi adicionada uma camada de classificação para distinguir entre frases com metáfora e frases sem metáfora. As frases do *dataset* foram previamente tokenizadas, aplicando truncagem e *padding*, com um comprimento máximo de 128 tokens, de forma a garantir uma representação uniforme das entradas.

Para a avaliação do modelo, foi utilizada validação cruzada estratificada com 5 partições, assegurando que a distribuição das classes fosse preservada em cada subconjunto. Em cada iteração, os dados foram divididos em conjuntos de treino e validação, sendo o modelo treinado sobre os dados de treino e avaliado sobre exemplos não observados durante essa fase. O treino foi realizado durante 2 épocas, utilizando o otimizador *AdamW* com uma taxa de aprendizagem de 2e-5. Esta configuração permitiu ajustar o modelo à tarefa sem introduzir uma complexidade excessiva no processo de treino.

Tal como nos restantes modelos, o desempenho foi avaliado com base em métricas como *precision*, *recall*, *F1-score* e *F2-score*, bem como através da análise da matriz de confusão. Esta avaliação permitiu medir a capacidade do modelo em identificar corretamente frases metafóricas e não metafóricas.

Adicionalmente, após a fase de validação cruzada, foi treinado um modelo final sobre a totalidade do *dataset*, com o objetivo de o utilizar posteriormente na fase de inferência. Esta etapa foi particularmente importante no contexto deste trabalho, uma vez que o modelo final foi utilizado para identificar metáforas em novas letras de músicas e, a partir dessas previsões, extrair *features* posteriormente integradas na tarefa de reconhecimento de emoções.

Desta forma, o modelo RoBERTa com *fine-tuning* desempenhou não apenas um papel de avaliação, mas também uma função central na construção da pipeline experimental do projeto. Este modelo revelou-se como o melhor para a tarefa de deteção de metáforas.

5.4. Extração de features

Após a seleção do modelo RoBERTa com *fine-tuning* como abordagem mais eficaz para a deteção de metáforas, este foi utilizado para extrair *features* a partir de um novo conjunto de dados composto por letras completas de músicas.

Numa primeira fase, cada letra foi segmentada em unidades menores, correspondentes a linhas individuais. Esta divisão permitiu tratar cada linha como uma unidade textual independente, aproximando-se do conceito de frase e facilitando a aplicação do modelo de classificação. Foram removidas linhas vazias ou compostas apenas por espaços, garantindo a consistência dos dados analisados.

De seguida, cada linha foi processada pelo modelo previamente treinado, que produziu uma previsão binária indicando a presença ou ausência de metáfora, bem como a probabilidade associada a essa previsão. Esta abordagem permitiu analisar cada letra de forma granular, identificando padrões ao longo do texto.

Com base nas previsões obtidas, foram definidas várias *features* ao nível da música, agregando a informação extraída das suas diferentes linhas. As *features* consideradas foram as seguintes:

- **NumLines**: número total de linhas da letra da música;
- **MetaphorCount**: número de linhas identificadas como contendo metáforas;
- **ContainsMetaphor**: variável binária que indica a presença (1) ou ausência (0) de metáforas na letra;
- **MetaphorDensity**: proporção de linhas com metáforas em relação ao número total de linhas da música.

Estas *features* permitem caracterizar quantitativamente a presença e distribuição de metáforas ao longo das letras, constituindo uma representação compacta e informativa do uso desta figura de estilo, como exemplificado na Tabela 2.

Tabela 2 - Exemplo base de dados das features de metáforas nas letras das músicas

Música	Nº de metáforas	Contém metáfora	Densidade de metáforas
The Lady Of Shalott	12	1	0,0952380952380952

Por fim, as *features* extraídas foram integradas no conjunto de dados original, ficando associadas a cada música. Este novo *dataset* foi posteriormente utilizado na fase de classificação de emoções, permitindo avaliar o contributo das metáforas na identificação do quadrante emocional das músicas.

Esta página foi intencionalmente deixada em branco.

6. Resultados

Este capítulo apresenta, em esquema, os resultados de todos os testes feitos aos modelos descritos no capítulo anterior. Para além disso, identifica também qual o melhor modelo e quais os resultados do projeto.

6.1. Identificação de metáforas

Para a identificação de metáforas, como já analisamos anteriormente, os melhores resultados serão aqueles que têm menos falsos previstos, quer sejam eles negativos ou positivos, e maior métrica F2-Score. De seguida, iremos analisar ao detalhe os resultados dos modelos elaborados, fazendo uma matriz de confusão e avaliando as métricas

Para cada modelo analisamos qual o tipo de metáfora que o modelo mais teve dificuldade em detetar (Falsos negativos) e os resultados estão apresentados na Tabela 3. A análise dos falsos negativos por tipo de metáfora permite identificar padrões relevantes no comportamento dos diferentes modelos.

Tabela 3 - Tipos de metáforas nos falsos negativos

Modelo	Tipo I	Tipo II	Tipo III	Tipo IV	Tipo V
Regressão Logística	23	7	8	11	5
Random Forest Classifier	37	22	30	26	14
SVC	23	5	9	10	5
XGBClassifier	44	27	38	35	23
BERT s/ *fine-tuning*	16	0	11	6	2
BERT c/ *fine-tuning*	17	1	2	3	0
RoBERTa s/ *fine-tuning*	15	1	12	4	1
RoBERTa c/ *fine-tuning*	4	4	4	3	3

De forma geral, observa-se que os modelos tradicionais, como o Random Forest e o XGBoost, apresentam maiores dificuldades na deteção de todos os tipos de metáforas, com especial incidência nos tipos I, III e IV, evidenciando uma limitação na captura de relações linguísticas mais complexas. Por outro lado, os modelos baseados em *Transformers*,

nomeadamente BERT e RoBERTa, demonstram um desempenho significativamente superior, registando um número inferior de falsos negativos em todos os tipos.

Em particular, o RoBERTa com *fine-tuning* destaca-se como o modelo mais eficaz, apresentando um desempenho equilibrado entre os diferentes tipos de metáforas e reduzindo de forma consistente os erros de classificação. Adicionalmente, verifica-se que alguns tipos de metáfora, como o tipo I, tendem a ser mais difíceis de identificar para a maioria dos modelos, possivelmente também devido à sua maior frequência na base de dados.

Estes resultados sugerem que a capacidade de detetar corretamente diferentes tipos de metáforas está fortemente associada à complexidade do modelo utilizado, sendo os modelos de *Deep Learning* mais adequados para capturar padrões linguísticos subtis e variados.

6.1.1. Regressão Logística

Os resultados obtidos, apresentados na Tabela 4, com o modelo de Regressão Logística demonstram um desempenho global consistente na tarefa de deteção de metáforas. Na configuração sem remoção de *stopwords*, o modelo apresenta um equilíbrio razoável entre *precision* e *recall* para ambas as classes, destacando-se uma ligeira superioridade na identificação de frases não metafóricas. No entanto, observa-se uma diminuição do *recall* na classe "Metáfora", indicando alguma dificuldade em identificar todas as ocorrências desta classe.

Com a inclusão de *stopwords*, verifica-se uma maior uniformidade entre as métricas das duas classes, traduzindo-se num desempenho mais equilibrado. Em particular, o *recall* da classe "Metáfora" melhora, sugerindo que a presença destas palavras pode contribuir positivamente para a identificação de padrões linguísticos associados às metáforas.

Tabela 4 - Relatório de classificação deteção de metáforas - RL

	Classificação	Precision	Recall	F1-score	F2-score
Sem StopWords	Não Metáfora	0,80	0,88	0,84	0,87
	Metáfora	0,87	0,78	0,83	0,80
Com StopWords	Não Metáfora	0,82	0,80	0,81	0,80
	Metáfora	0,81	0,83	0,82	0,82

A análise da matriz de confusão, Tabela 5, confirma estes resultados, evidenciando uma redução dos falsos negativos na classe "Metáfora" quando são consideradas *stopwords*. Assim, conclui-se que a inclusão de *stopwords* conduz a um desempenho mais equilibrado do modelo, ainda que as diferenças globais entre as duas abordagens sejam relativamente moderadas.

Tabela 5 - Matriz de confusão deteção de metáforas - RL

			Previsto	
			Não metáfora	Metáfora
Sem StopWords	Verdadeiro	Não metáfora	221	29
		Metáfora	54	196
Com StopWords		Não metáfora	200	50
		Metáfora	43	207

6.1.2. Random Forest Classifier

Os resultados obtidos com o modelo *Random Forest Classifier* evidenciam um desempenho global razoável, mas com limitações significativas na identificação de metáforas. Através da Tabela 6, conseguimos ver que na configuração com remoção de *stopwords*, o modelo apresenta um *recall* muito elevado para a classe "Não metáfora", indicando uma forte capacidade para identificar corretamente frases não metafóricas. No entanto, o *recall* da classe "Metáfora" é consideravelmente reduzido, refletindo um elevado número de falsos negativos e uma dificuldade em detetar este tipo de construções linguísticas.

Na ausência de remoção de *stopwords*, observa-se uma melhoria no desempenho da classe "Metáfora", nomeadamente ao nível do *recall* e das métricas F1 e F2, embora o modelo continue a apresentar um desequilíbrio entre classes.

Tabela 6 - Relatório de classificação deteção de metáforas - RFC

	Classificação	Precision	Recall	F1-score	F2-score
Sem StopWords	Não Metáfora	0,65	0,96	0,78	0,88
	Metáfora	0,92	0,48	0,64	0,53
Com StopWords	Não Metáfora	0,70	0,93	0,80	0,87
	Metáfora	0,90	0,60	0,72	0,64

A análise da matriz de confusão, Tabela 7, confirma esta tendência, evidenciando uma redução dos falsos negativos quando as *stopwords* são mantidas. Apesar desta melhoria, o modelo revela limitações face a abordagens mais avançadas, particularmente na capacidade de capturar padrões linguísticos complexos associados às metáforas.

Tabela 7 - Matriz de confusão deteção de metáforas - RFC

			Previsto	
			Não metáfora	Metáfora
Sem StopWords	Verdadeiro	Não metáfora	240	10
		Metáfora	129	121
Com StopWords		Não metáfora	233	17
		Metáfora	100	150

6.1.3. Support Vector Classifier

Os resultados obtidos com o modelo Support Vector Classifier (SVC) demonstram um desempenho global equilibrado na tarefa de deteção de metáforas. Na configuração com remoção de stopwords, o modelo apresenta valores elevados de precision e recall para ambas as classes, com um ligeiro destaque para a classe "Não metáfora". Ainda assim, através da

Tabela 8 verifica-se uma pequena redução no recall da classe "Metáfora", indicando a existência de alguns falsos negativos.

Na ausência de remoção de stopwords, observa-se uma maior uniformidade entre as métricas das duas classes, com valores de precision, recall e F1-score bastante próximos. Esta configuração conduz a um desempenho mais equilibrado entre a identificação de frases metafóricas e não metafóricas, sugerindo que a presença de stopwords contribui positivamente para a generalização do modelo.

Tabela 8 - Relatório de classificação deteção de metáforas - SVC

	Classificação	Precision	Recall	F1-score	F2-score
Sem StopWords	Não Metáfora	0,81	0,88	0,84	0,86
	Metáfora	0,87	0,79	0,83	0,81
Com StopWords	Não Metáfora	0,82	0,80	0,81	0,81
	Metáfora	0,81	0,82	0,82	0,82

A análise da matriz de confusão, Tabela 9, confirma esta tendência, evidenciando uma redução do número de falsos negativos na classe "Metáfora" quando as stopwords são mantidas. De forma geral, o modelo SVC apresenta um desempenho estável e consistente, sendo uma das abordagens mais equilibradas entre as analisadas.

Tabela 9 - Matriz de confusão deteção de metáforas - SVC

			Previsto	
			Não metáfora	Metáfora
Sem StopWords	Verdadeiro	Não metáfora	220	30
		Metáfora	52	198
Com StopWords		Não metáfora	201	49
		Metáfora	44	206

6.1.4. XGBoost Classifier

Os resultados obtidos com o modelo *XGBoost Classifier* evidenciam um comportamento fortemente dependente da configuração de pré-processamento adotada. Na presença de remoção de *stopwords*, o modelo apresenta um desempenho muito elevado na identificação da classe "Não metáfora", mas revela uma degradação acentuada na deteção de metáforas, com valores reduzidos de *recall* e F1-score, indicando um elevado número de falsos negativos, como mostra a Tabela 10.

Por outro lado, quando as *stopwords* são mantidas, observa-se uma melhoria significativa no equilíbrio entre as duas classes, com valores mais homogéneos nas métricas de avaliação. Esta configuração permite ao modelo identificar de forma mais consistente tanto frases metafóricas como não metafóricas.

Tabela 10 - Relatório de classificação deteção de metáforas - XGBoost Classifier

	Classificação	Precision	Recall	F1-score	F2-score
Sem StopWords	Não Metáfora	0,59	0,98	0,74	0,86
	Metáfora	0,93	0,33	0,49	0,38
Com StopWords	Não Metáfora	0,67	0,63	0,65	0,64
	Metáfora	0,65	0,69	0,67	0,68

A análise da matriz de confusão - Tabela 11 - confirma esta diferença, evidenciando uma redução substancial dos erros na classe "Metáfora" quando as *stopwords* são incluídas. Ainda assim, o modelo apresenta maior variabilidade no desempenho quando comparado com abordagens baseadas em *Deep Learning*, sugerindo uma menor robustez na generalização.

Tabela 11 - Matriz de confusão deteção de metáforas - XGBoost Classifier

			Previsto	
			Não metáfora	Metáfora
Sem StopWords	Verdadeiro	Não metáfora	244	6
		Metáfora	167	83
Com StopWords		Não metáfora	157	93
		Metáfora	77	173

6.1.5. BERT sem *fine-tuning*

Os resultados obtidos com o modelo BERT sem *fine-tuning* demonstram um desempenho elevado e equilibrado na tarefa de deteção de metáforas. Através da Tabela 12, conseguimos perceber que o modelo apresenta valores consistentes de *precision*, *recall* e F1-score para ambas as classes, evidenciando uma boa capacidade de generalização. Em particular, observa-se um equilíbrio na identificação de frases metafóricas e não metafóricas, sem enviesamento significativo para uma das classes.

Tabela 12 - Relatório de classificação deteção de metáforas - BERT sem fine-tuning

Classificação	Precision	Recall	F1-score	F2-score
Não metáfora	0,86	0,90	0,88	0,89
Metáfora	0,89	0,86	0,87	0,86

A análise da matriz de confusão, pela Tabela 13, confirma este comportamento, evidenciando um número reduzido de falsos negativos na classe “Metáfora”. Estes resultados indicam que as representações contextuais geradas pelo BERT são suficientemente informativas para capturar padrões semânticos relevantes, mesmo sem adaptação específica à tarefa.

Tabela 13 - Matriz de confusão deteção de metáforas - de BERT sem fine-tuning

		Previsto	
		Não metáfora	Metáfora
Verdadeiro	Não metáfora	224	26
	Metáfora	35	215

No entanto, quando comparado com a versão com *fine-tuning*, verifica-se que o modelo ainda apresenta margem de melhoria, nomeadamente na adaptação ao domínio específico do problema.

6.1.6. BERT com *fine-tuning*

Os resultados obtidos com o modelo BERT com *fine-tuning* evidenciam um desempenho superior face à versão sem ajuste, destacando-se como uma das abordagens mais eficazes na tarefa de deteção de metáforas. O modelo apresenta valores elevados e equilibrados de *precision, recall* e *F1-score* para ambas as classes, demonstrando uma forte capacidade de generalização e de adaptação ao domínio específico do problema.

Em particular, e como podemos ver através da Tabela 14, verifica-se uma melhoria na identificação da classe "Metáfora", com um *recall* elevado, indicando que o modelo consegue detetar a maioria das ocorrências desta classe. Este comportamento reflete-se também nos valores de *F2-score*, que favorecem o *recall*, sendo particularmente relevantes nesta tarefa.

Tabela 14 - Relatório de classificação deteção de metáforas - BERT com fine-tuning

Classificação	Precision	Recall	F1-score	F2-score
Não metáfora	0,92	0,89	0,91	0,90
Metáfora	0,90	0,92	0,91	0,92

A matriz de confusão, na Tabela 15, confirma estes resultados, evidenciando um número reduzido de falsos negativos, o que demonstra a eficácia do modelo na deteção de metáforas. Estes resultados sugerem que o processo de *fine-tuning* permite ao modelo ajustar-se às

características específicas do dataset, capturando de forma mais precisa padrões linguísticos associados à presença de metáforas.

Tabela 15 - Matriz de confusão deteção de metáforas - BERT com fine-tuning

		Previsto	
		Não metáfora	Metáfora
Verdadeiro	Não metáfora	290	30
	Metáfora	26	294

6.1.7. RoBERTA sem *fine-tuning*

Os resultados obtidos com o modelo RoBERTa sem *fine-tuning* demonstram um desempenho bastante consistente e equilibrado na tarefa de deteção de metáforas. Na Tabela 16, o modelo apresenta valores idênticos de *precision*, *recall*, F1-score e F2-score para ambas as classes, indicando uma capacidade uniforme de identificação tanto de frases metafóricas como não metafóricas.

Tabela 16 - Relatório de classificação deteção de metáforas - RoBERTa sem fine-tuning

Classificação	Precision	Recall	F1-score	F2-score
Não metáfora	0,87	0,87	0,87	0,87
Metáfora	0,87	0,87	0,87	0,87

A análise da matriz de confusão, Tabela 17, confirma este comportamento equilibrado, evidenciando um número semelhante de erros em ambas as classes e uma boa capacidade geral de classificação. Em particular, observa-se um número reduzido de falsos negativos na classe “Metáfora”, o que demonstra a eficácia do modelo na deteção deste tipo de construções linguísticas.

Tabela 17 - Matriz de confusão deteção de metáforas - RoBERTa sem fine-tuning

		Previsto	
		Não metáfora	Metáfora
Verdadeiro	Não metáfora	218	32
	Metáfora	33	217

Estes resultados sugerem que as representações contextuais geradas pelo RoBERTa, mesmo sem *fine-tuning*, são altamente informativas e permitem capturar padrões semânticos relevantes para a tarefa. De forma geral, o modelo apresenta um desempenho sólido e comparável ao BERT sem *fine-tuning*, destacando-se pela sua estabilidade e equilíbrio entre as diferentes métricas.

6.1.8. RoBERTA com *fine-tuning*

Os resultados obtidos com o modelo RoBERTa com *fine-tuning* evidenciam o melhor desempenho entre todos os modelos analisados na tarefa de deteção de metáforas. O modelo apresenta valores elevados e equilibrados de *precision, recall*, *F1-score* e *F2-score* para ambas as classes, demonstrando uma elevada capacidade de generalização e uma distinção eficaz entre frases metafóricas e não metafóricas.

Em particular, pela Tabela 18, observa-se um excelente desempenho na identificação da classe "Metáfora", com valores de recall elevados e um número reduzido de falsos negativos, o que indica que o modelo consegue detetar a grande maioria das ocorrências desta classe. Este comportamento é especialmente relevante, uma vez que a deteção de metáforas constitui o principal objetivo desta tarefa.

Tabela 18 - Relatório de classificação deteção de metáforas - RoBERTa com fine-tuning

Classificação	Precision	Recall	F1-score	F2-score
Não metáfora	0,93	0,91	0,92	0,91
Metáfora	0,91	0,93	0,92	0,93

A análise da matriz de confusão presente na Tabela 19, confirma estes resultados, evidenciando um número muito reduzido de erros de classificação em ambas as classes.

Estes resultados sugerem que o processo de *fine-tuning* permitiu ao modelo ajustar-se de forma eficaz às características específicas do dataset, explorando plenamente o seu potencial na modelação de relações linguísticas complexas.

Tabela 19 - Matriz de confusão deteção de metáforas - RoBERTa com fine-tuning

		Previsto	
		Não metáfora	Metáfora
Verdadeiro	Não metáfora	293	30
	Metáfora	22	299

De forma geral, o RoBERTa com *fine-tuning* destaca-se como o modelo mais robusto e eficaz, constituindo a abordagem mais adequada para a deteção de metáforas no contexto deste trabalho.

6.2. Comparação Global dos Modelos

De forma geral, os modelos tradicionais, como a Regressão Logística, o SVC, o *Random Forest e o XGBoost,* apresentam desempenhos razoáveis, sendo capazes de identificar padrões relevantes nos dados. No entanto, estes modelos demonstram, em alguns casos, limitações na deteção da classe "Metáfora", evidenciando uma tendência para gerar falsos negativos, particularmente quando não conseguem capturar relações semânticas mais complexas.

A inclusão ou remoção de *stopwords* revelou ter um impacto variável consoante o modelo, sendo, em alguns casos, determinante para melhorar o equilíbrio entre as classes. Ainda assim, mesmo nas melhores configurações, os modelos tradicionais não atingem o mesmo nível de desempenho das abordagens baseadas em *Deep Learning*.

Por outro lado, os modelos baseados em arquiteturas *Transformer*, como o BERT e o RoBERTa, destacam-se claramente, mesmo quando utilizados sem *fine-tuning*. Estes modelos demonstram uma maior capacidade de capturar o contexto e as relações semânticas presentes nas frases, resultando em desempenhos mais equilibrados e robustos.

A aplicação de *fine-tuning* conduz a melhorias adicionais significativas, permitindo adaptar os modelos ao domínio específico do problema. Em particular, o modelo RoBERTa com *fine-tuning* apresenta o melhor desempenho global, combinando elevados valores de *precision* e *recall* para ambas as classes e evidenciando uma forte capacidade de generalização.

Assim, conclui-se que a utilização de modelos baseados em *Transformers*, especialmente com *fine-tuning*, constitui a abordagem mais eficaz para a deteção de metáforas no contexto deste trabalho, sendo esta a solução adotada nas etapas subsequentes do projeto.

6.3. Aplicação das features na identificação de emoções

Após a extração das *features* baseadas em metáforas a partir das 180 letras de músicas descritas anteriormente, foi construído um novo *dataset* que integra estas características com as restantes *features* previamente definidas. Este processo permitiu garantir que a informação relativa às metáforas está diretamente associada ao mesmo conjunto de dados utilizado na tarefa de classificação emocional.

De seguida, procedeu-se à avaliação do impacto destas características na tarefa de reconhecimento de emoções em músicas. Para tal, foram desenvolvidos e testados diferentes modelos de classificação, considerando dois cenários distintos: um em que apenas foram utilizadas as *features* previamente existentes e outro em que foram incluídas as *features* associadas às metáforas.

O objetivo desta análise consiste em comparar o desempenho dos modelos em ambos os cenários, de forma a avaliar o contributo das metáforas na identificação do quadrante emocional das músicas. Para esse efeito, foram utilizados cinco modelos de classificação, incluindo abordagens tradicionais e modelos baseados em *Deep Learning*, permitindo analisar diferentes capacidades de generalização e adaptação aos dados.

Os resultados obtidos são apresentados sob a forma de métricas de avaliação e tabelas comparativas, sendo posteriormente analisados e interpretados de modo a identificar padrões, tendências e limitações das abordagens utilizadas. Esta análise permitirá compreender se a inclusão de *features* baseadas em metáforas contribui de forma significativa para a melhoria do desempenho dos modelos na tarefa de reconhecimento de emoções.

6.3.1. Random Forest Classifier

Os resultados obtidos com o modelo *Random Forest* na tarefa de reconhecimento de emoções evidenciam um impacto reduzido da inclusão das *features* baseadas em metáforas. De forma global, pelas Tabela 20 e Tabela 21 observa-se uma ligeira melhoria na média do F1-score, passando de 0,67 para 0,68, o que indica um contributo positivo, ainda que pouco expressivo.

Tabela 20 - Relatório de classificação - RFC sem features de metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,68	0,57	0,62	0,67
Q2	0,83	0,73	0,78	
Q3	0,59	0,76	0,67	
Q4	0,64	0,60	0,62	

Tabela 21 - Relatório de classificação - RFC com as features das metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,68	0,59	0,63	0,68
Q2	0,83	0,73	0,78	
Q3	0,59	0,76	0,67	
Q4	0,66	0,60	0,62	

Analisando individualmente os quadrantes emocionais, verifica-se que o desempenho se mantém praticamente inalterado na maioria das classes, com exceção de uma ligeira melhoria no quadrante Q1 e Q4 ao nível do *recall* e F1-score. Esta evolução sugere que as *features* de metáforas podem introduzir alguma informação adicional útil na distinção destes quadrantes, embora o seu impacto seja limitado.

A análise da matriz de confusão nas Tabela 22 e Tabela 23 confirma esta tendência, evidenciando diferenças mínimas entre os dois cenários. A maioria das classificações corretas mantém-se estável, sendo as alterações pouco significativas em termos de redistribuição dos erros entre classes.

Tabela 22 - Matriz de confusão - RFC sem as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	25	1	8	10
	Q2	1	30	10	0
	Q3	4	4	39	4
	Q4	7	1	9	25

Tabela 23 - Matriz de confusão - RFC com as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	26	1	8	9
	Q2	1	30	10	0
	Q3	4	4	39	4
	Q4	7	1	9	25

Assim, conclui-se que a inclusão de *features* de metáforas tem um impacto residual no desempenho do modelo, não sendo suficiente para melhorar de forma significativa a sua capacidade de classificação emocional.

6.3.2. Regressão Logística

Os resultados obtidos, presentes nas Tabela 24 e Tabela 25, com o modelo de Regressão Logística evidenciam um impacto ligeiramente positivo da inclusão das *features* baseadas em metáforas na tarefa de reconhecimento de emoções. Verifica-se um aumento marginal da média do F1-score de 0,69 para 0,70, indicando uma melhoria global pouco expressiva.

Ao nível dos diferentes quadrantes emocionais, as melhorias concentram-se sobretudo nos quadrantes Q1 e Q3, com ligeiros aumentos nos valores de *recall* e F1-score. No entanto, nos restantes quadrantes, o desempenho mantém-se praticamente inalterado, não se observando ganhos consistentes.

Tabela 24 - Relatório de classificação - RL sem as features das metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,71	0,57	0,63	0,69
Q2	0,84	0,76	0,79	
Q3	0,65	0,82	0,72	
Q4	0,61	0,60	0,60	

Tabela 25 - Relatório de classificação - RL com as features das metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,74	0,59	0,66	0,70
Q2	0,84	0,76	0,79	
Q3	0,65	0,84	0,74	
Q4	0,62	0,60	0,61	

A análise das matrizes de confusão, presentes nas Tabela 26 e Tabela 27 confirma que as alterações são residuais, traduzindo-se apenas em pequenas variações no número de classificações corretas.

Tabela 26 - Matriz de confusão - RL sem as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	25	3	7	9
	Q2	1	31	7	2
	Q3	2	2	42	5
	Q4	7	1	9	25

Tabela 27 - Matriz de confusão - RL com as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	26	3	7	8
	Q2	1	31	7	2
	Q3	1	2	43	5
	Q4	7	1	9	25

Deste modo, conclui-se que, embora as *features* de metáforas possam introduzir alguma informação adicional, o seu impacto no desempenho da Regressão Logística é reduzido e não altera significativamente a capacidade global de classificação do modelo.

6.3.3. Support Vector Classifier

Os resultados obtidos com o modelo *Support Vector Classifier* (SVC) indicam que a inclusão das *features* baseadas em metáforas não teve um impacto significativo no desempenho global do modelo. A média do F1-score mantém-se inalterada em 0,70 em ambos os cenários (como podemos observar na Tabela 28 e na Tabela 29) sugerindo que estas *features* não introduzem melhorias relevantes na capacidade de classificação do modelo.

Tabela 28 - Relatório de classificação - SVC sem as features das metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,63	0,77	0,69	0,70
Q2	0,90	0,63	0,74	
Q3	0,63	0,80	0,71	
Q4	0,77	0,55	0,64	

Tabela 29 - Relatório de classificação - SVC com as features das metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,62	0,75	0,68	0,70
Q2	0,90	0,63	0,74	
Q3	0,65	0,84	0,74	
Q4	0,77	0,55	0,64	

Ao nível dos diferentes quadrantes emocionais, observam-se apenas variações pontuais. Em particular, verifica-se uma ligeira melhoria no quadrante Q3, com um aumento no *recall* e no F1-score, indicando uma melhor capacidade de identificação desta classe. No entanto, esta melhoria é compensada por pequenas variações noutros quadrantes, resultando num desempenho global praticamente idêntico.

A Tabela 30 e a Tabela 31 representam as matrizes de confusão que confirmam esta estabilidade, evidenciando diferenças mínimas na distribuição das classificações corretas e incorretas entre os dois cenários. As alterações observadas são residuais e não alteram de forma significativa o comportamento do modelo.

Tabela 30 - Matriz de confusão - SVC sem as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	34	0	6	4
	Q2	5	26	10	0
	Q3	6	1	41	3
	Q4	9	2	8	23

Tabela 31 - Matriz de confusão - SVC com as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	33	0	6	5
	Q2	6	26	9	0
	Q3	5	1	43	2
	Q4	9	2	8	23

Deste modo, conclui-se que, para o SVC, a inclusão de *features* de metáforas não contribui de forma relevante para a melhoria do desempenho, confirmando a reduzida influência destas características na tarefa de classificação emocional.

6.3.4. MLP

Os resultados obtidos com o modelo *Multi-Layer Perceptron* (MLP) evidenciam um impacto negativo da inclusão das *features* baseadas em metáforas na tarefa de reconhecimento de emoções. Verifica-se na Tabela 32 e na Tabela 33 uma diminuição da média do F1-score de 0,67 para 0,63, indicando uma degradação do desempenho global do modelo.

Tabela 32 - Relatório de classificação - MLP sem as features das metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,68	0,59	0,63	0,67
Q2	0,78	0,78	0,78	
Q3	0,63	0,67	0,65	
Q4	0,58	0,62	0,60	

Tabela 33 - Relatório de classificação - MLP com as features das metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,68	0,59	0,63	0,63
Q2	0,70	0,73	0,71	
Q3	0,63	0,65	0,64	
Q4	0,53	0,57	0,55	

Analisando os diferentes quadrantes emocionais, observa-se que esta redução é transversal à maioria das classes, com particular destaque para os quadrantes Q2 e Q4, onde se verificam descidas nos valores de *precision*, *recall* e F1-score. Estas alterações sugerem que a introdução das *features* de metáforas poderá ter introduzido ruído ou redundância na representação dos dados, dificultando o processo de aprendizagem do modelo.

A análise das matrizes de confusão presentes na Tabela 34 e na Tabela 35 confirma esta tendência, evidenciando um aumento dos erros de classificação em vários quadrantes, nomeadamente uma maior dispersão das previsões incorretas. Este comportamento indica uma menor capacidade do modelo em distinguir corretamente entre os diferentes estados emocionais após a inclusão das novas *features*.

Tabela 34 - Matriz de confusão - MLP sem as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	26	4	6	8
	Q2	2	32	5	2
	Q3	4	4	34	9
	Q4	6	1	9	26

Tabela 35 - Matriz de confusão - MLP com as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	26	4	3	11
	Q2	3	30	6	2
	Q3	5	5	33	8
	Q4	4	4	10	24

Deste modo, conclui-se que, no caso do MLP, a utilização de *features* baseadas em metáforas não só não contribui para a melhoria do desempenho, como pode comprometer a eficácia do modelo na tarefa de classificação emocional.

6.3.5. DNN

Os resultados obtidos com o modelo *Deep Neural Network* (DNN) evidenciam um impacto negativo da inclusão das *features* baseadas em metáforas na tarefa de reconhecimento de emoções. Observa-se na Tabela 36 e na Tabela 37 uma diminuição da média do F1-score de 0,68 para 0,64, indicando uma redução do desempenho global do modelo.

Tabela 36 - Relatório de classificação - DNN sem as features das metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,65	0,59	0,62	0,68
Q2	0,82	0,76	0,78	
Q3	0,73	0,73	0,73	
Q4	0,55	0,64	0,59	

Tabela 37 - Relatório de classificação - DNN com as features das metáforas

Classificação	Precision	Recall	F1-score	Média F1-Score
Q1	0,64	0,66	0,65	0,64
Q2	0,71	0,71	0,71	
Q3	0,60	0,67	0,63	
Q4	0,60	0,50	0,55	

Ao nível dos diferentes quadrantes emocionais, verifica-se uma degradação generalizada das métricas, com particular incidência nos quadrantes Q2 e Q3, onde se registam reduções nos valores de precision, recall e F1-score. Embora o quadrante Q1 apresente uma ligeira melhoria no recall, esta não é suficiente para compensar as perdas observadas nas restantes classes.

A análise da matriz de confusão confirma esta tendência (Tabela 38 e Tabela 39), evidenciando um aumento da dispersão das previsões e um crescimento dos erros de classificação em vários quadrantes. Este comportamento sugere que a introdução das *features* de metáforas poderá ter interferido com a capacidade do modelo em aprender padrões relevantes, possivelmente devido à redundância ou baixa relevância destas características no contexto da tarefa.

Tabela 38 - Matriz de confusão - DNN sem as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	26	2	5	11
	Q2	2	31	4	4
	Q3	4	3	37	7
	Q4	8	2	5	27

Tabela 39 - Matriz de confusão - DNN com as features das metáforas

		Previsto			
		Q1	Q2	Q3	Q4
Verdadeiro	Q1	29	2	6	7
	Q2	3	29	8	1
	Q3	6	5	34	6
	Q4	7	5	9	21

Deste modo, conclui-se que, no caso do DNN, a inclusão de *features* baseadas em metáforas não contribui para a melhoria do desempenho, tendo, pelo contrário, um efeito negativo na capacidade de classificação emocional do modelo.

6.4. Comparação dos modelos

A análise dos resultados obtidos permite verificar que a inclusão de *features* baseadas em metáforas tem um impacto limitado na tarefa de reconhecimento de emoções em músicas. Embora se observem melhorias pontuais em alguns modelos, como a Regressão Logística e o *Random Forest*, estas são pouco expressivas e não se traduzem num aumento consistente do desempenho global.

Por outro lado, em modelos mais complexos, nomeadamente o MLP e o DNN, a inclusão destas *features* conduz mesmo a uma degradação do desempenho, sugerindo que a

informação adicional introduzida poderá ser redundante ou insuficientemente discriminativa para esta tarefa. No caso do SVC, o desempenho mantém-se praticamente inalterado, reforçando a ideia de que as *features* de metáforas não exercem uma influência significativa na capacidade de classificação emocional.

Estes resultados indicam que, apesar da relevância das metáforas na expressão linguística e na deteção de emoções ao nível local (frase), o seu contributo ao nível global da música, quando representado através de *features* agregadas, é reduzido. Assim, verifica-se que a utilização destas características, por si só, não é suficiente para melhorar de forma consistente o desempenho dos modelos na tarefa de reconhecimento de emoções, sendo necessário explorar abordagens complementares ou representações mais ricas para capturar a complexidade emocional presente nas letras das músicas.

6.5. Feature selection

Com o objetivo de melhorar o desempenho dos modelos de classificação e reduzir a dimensionalidade do conjunto de dados, foi aplicada uma técnica de *feature selection*. Este processo permite identificar as variáveis mais relevantes para a tarefa de classificação, eliminando *features* redundantes ou com baixo contributo, o que pode levar a modelos mais eficientes e com melhor capacidade de generalização.

Para este efeito, foi utilizado o algoritmo *ReliefF*, uma técnica baseada em instâncias que avalia a importância de cada *feature* com base na sua capacidade de distinguir exemplos próximos pertencentes a diferentes classes. Este método é particularmente adequado para problemas de classificação multiclasse e tem a vantagem de considerar interações entre *features*, ao contrário de métodos univariados.

Inicialmente, o algoritmo foi aplicado ao conjunto completo de *features*, composto por 247 variáveis. Como resultado, foi obtida uma pontuação de importância para cada *feature*, permitindo ordená-las por relevância. Entre as *features* mais importantes destacam-se *Tone*, *AvgArousal_Q2*, *AvgValence* e *tone_neg*, evidenciando a relevância de características semânticas e emocionais na tarefa de classificação.

Com base nesta análise, foram selecionadas as 100 *features* mais relevantes, formando um novo subconjunto de dados reduzido. Este subconjunto foi posteriormente utilizado como entrada para os diferentes modelos de classificação, de forma a avaliar o impacto da seleção de *features* no desempenho dos modelos.

Para garantir uma avaliação robusta, foi utilizada validação cruzada estratificada (*Stratified K-Fold*) com 5 divisões, assegurando a preservação da distribuição das classes em cada subconjunto. Foram testados cinco modelos de classificação: Regressão Logística, Support Vector Classifier (SVC), Random Forest, Multi-Layer Perceptron (MLP) e uma *Deep Neural Network* (DNN).

Os resultados obtidos demonstram que o modelo SVC apresentou o melhor desempenho, com um F1-score médio de aproximadamente 0.72, seguido do *Random Forest* com um valor de cerca de 0.70. Por outro lado, o modelo MLP apresentou o desempenho mais baixo, com um F1-score de 0.58, evidenciando maior dificuldade em generalizar com o conjunto reduzido de *features*. Os resultados estão resumidos na Tabela 40

Tabela 40 - Resultados deteção de emoções - melhores features

Modelo	**F1-Score**
SVC	0,722
Random Forest	0,704
DNN	0,701
Logistic Regression	0,668
MLP	0,587

7. Conclusão

O presente projeto teve como principal objetivo estudar o papel das metáforas nas letras de músicas, no contexto da Recuperação da Informação Musical (MIR) e do Reconhecimento de Emoções nas Músicas (MER), procurando perceber se esta figura de estilo pode contribuir para a melhoria na deteção de emoções.

Para atingir este objetivo, foram desenvolvidos diferentes modelos de deteção de metáforas, recorrendo tanto a abordagens tradicionais como a técnicas de *Deep Learning*. Após a comparação entre os vários modelos, verificou-se que os modelos baseados em *Transformers*, em particular o RoBERTa com *fine-tuning*, apresentaram o melhor desempenho. Com base neste modelo, que obteve os melhores resultados, procedeu-se à extração de *features* relacionadas com a presença de metáforas nas letras das músicas, nomeadamente a contagem, a densidade e a sua existência. Estas *features* foram posteriormente integradas num conjunto de modelos de classificação emocional, permitindo avaliar o seu impacto na tarefa de MER.

Os resultados obtidos demonstram que, na tarefa de deteção de metáforas, os modelos de *Deep Learning* superam claramente os modelos tradicionais, com o RoBERTa com *fine-tuning* a atingir os melhores resultados. No entanto, no contexto do reconhecimento de emoções, a inclusão das *features* baseadas em metáforas revelou um impacto limitado. Em alguns modelos, observou-se uma ligeira melhoria, enquanto noutros o desempenho se manteve inalterado ou até diminuiu, especialmente nos modelos mais complexos. Assim, conclui-se que as metáforas, quando representadas através de *features* simples e agregadas, não contribuem de forma significativa para a melhoria do desempenho na tarefa de classificação emocional.

Deste modo, embora as metáforas sejam elementos linguisticamente relevantes e frequentemente associadas à expressão de emoções, a sua representação neste trabalho não foi suficiente para capturar a complexidade emocional presente nas letras das músicas. Estes resultados sugerem que a relação entre linguagem figurativa e emoção é mais complexa do que pode ser modelada através de *features* simples como a deteção de metáforas.

Este trabalho contribui para a área ao explorar a utilização de metáforas como *features* na tarefa de MER, bem como ao apresentar uma comparação sistemática entre diferentes

abordagens de modelação. Adicionalmente, foi desenvolvida uma pipeline completa, desde a deteção de metáforas até à sua integração na classificação emocional, permitindo uma análise aprofundada do seu impacto.

Apesar dos contributos apresentados, este estudo apresenta algumas limitações. O tamanho reduzido do *dataset*, composto por 180 músicas, pode ter condicionado os resultados obtidos. Para além disso, as *features* utilizadas para representar as metáforas são relativamente simples, podendo não captar toda a riqueza semântica destas construções. Acresce ainda a perda de contexto ao nível da música, uma vez que as *features* foram agregadas, e a natureza subjetiva da própria tarefa de reconhecimento de emoções.

Como trabalho futuro, sugere-se a exploração de abordagens mais avançadas para a representação das metáforas, nomeadamente através de *embeddings* contextuais. Poderá também ser relevante considerar o tipo de metáfora como *feature*, bem como adotar abordagens multimodais que integrem simultaneamente áudio e letra. Adicionalmente, a aplicação de modelos baseados em *Transformers* diretamente à tarefa de MER poderá permitir captar de forma mais eficaz a complexidade emocional presente nas músicas.

Em suma, este trabalho reforça a importância da linguagem figurativa na análise textual, mas evidencia também a necessidade de abordagens mais sofisticadas para capturar o seu verdadeiro impacto na expressão emocional.

8. Referências

Agrawal, Y., Shanker, R. G., & Alluri, V. (6 de 1 de 2021). Transformer-based approach towards music emotion recognition from lyrics. *Lecture notes in computer science*, pp. 167-175.

Andrade, E. d. (1951). Urgentemente. Em E. d. Andrade, *Até Amanhã*.

Avram, A.-M., Păiş, V. V., & Tufiş, D. D. (setembro de 2021). A Tool for Multilingual Legal Document Classification with EuroVoc Descriptors. pp. 92-101.

Babieno, M., Takeshita, M., Radisavljevic, D., Rzepka, R., & Araki, K. (17 de 02 de 2022). Miss RoBERTa WiLDe: Metaphor Identification Using Masked Language Model with Wiktionary Lexical Definitions. *Applied Sciences, 12*(4), p. 2081.

Basbaum, S. R. (2003). Synesthesia and digital perception. *Originally presented at the 3rd Subtle Technologies Conference, Toronto*.

Birke, J. (11 de Julho de 2005). A clustering approach for the unsupervised recognition of nonliteral language.

Camões, L. d. (1572). Os Lusiadas. Em L. d. Camões, *Os Lusiadas*.

Editora, P. (2025). *Infopédia*. Obtido de https://www.infopedia.pt/artigos/$figuras-de-estilo

Fass, D. (1991). met*: A method for discriminating metonymy and metaphor by computer. *Computational linguistics, 17*(1), pp. 49-90.

Gabrielsson, A. (2002). Emotion Perceived and Emotion Felt: Same or Different? *Musicae Scientiae (special issue), European Society for the Cognitive Sciences of Music (ESCOM)*, pp. 123-147.

Goldberg, N. (1986). *Writing Down the Bones*. SHAMBHALA PUBLICATIONS INC.

Hevner, K. (1936). Experimental Studies of the Elements of Expression in Music. *American Journal of Psychology*, pp. 246-268.

Hughes, E. (10 de 7 de 2024). Obtido de https://www.musicalmum.com/songs-with-metaphors/

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), pp. 217-238.

Kesarwani, V., Inkpen, D., Szpakowicz, S., & Tanasescu, C. (Agosto de 2017). Metaphor Detection in a Poetry Corpus. *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pp. 1-9.

Krennmayr, T., & Steen, G. (2017). VU Amsterdam Metaphor Corpus. Em *Handbook of linguistic annotation* (pp. 1053-1071).

Krishnakumaran, S., & Zhu, X. (abril de 2007). Hunting elusive metaphors using lexical resources. *Proceedings of the Workshop on Computational approaches to Figurative Language*, pp. 13-20.

Lakoff, G., & Mark Johnson. (1980). *Metaphors We Live By.* University of Chicago press.

Malheiro, R. (Agosto de 2016). Emotion-based Analysis and Classification of Music Lyrics (Doctoral dissertation).

Martinazzo, B. (2010). Um Método de Identificação de Emoções em Textos Curtos para o Português.

Minixhofer, B., Pfeiffer, J., & Vulić, I. (Julho de 2023). Where's the point? self-supervised multilingual punctuation-agnostic sentence segmentation. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 1*, pp. 7215-7235.

Mohler, M., Brunson, M., Rink, B., & Tomlinson, M. (2016). Introducing the LCC Metaphor Datasets. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 4221-4227.

Nordquist, R. (1 de 5 de 2025). *The Different Types of Metaphors*. Obtido de https://www.thoughtco.com/ways-of-looking-at-a-metaphor-1691815

Oktavia, D. (2023). Figurative language in Tame Impala's Song Lyrics.

Panda, R., Malheiro, R., & Paiva, R. P. (2020). Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing, 11*(4), pp. 614-626.

Pessoa, F. (1986). Meu pensamento é um rio subterrâneo. Em F. Pessoa, *Obra Poética e em Prosa*.

Russell, J. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, pp. 1161–1178.

Seyeditabari, A., Tabari, N., Gholizade, S., & Zadrozny, W. (19 de 6 de 2019). Emotional Embeddings: Refining Word Embeddings to Capture Emotional Content of Words.

Shakespeare, W. (1597). *Romeo and Juliet*.

Shakespeare, W. (1602). *Hamlet*.

Shakespeare, W. (1623). *As You Like It*.

Shutova, E., Teufel, S., & Korhonen, A. (2013). Statistical Metaphor Processing. *Computational Linguistics, 39*(2), pp. 301-353.

Tambun, M. P., & Sianipar, Y. A. (2014). FIGURES OF SPEECH IN WESTLIFE'S SELECTED. *Linguistica, 3*(3), p. 148300.

Tanasescu, C., Kesarwani, V., & Inkpen, D. (2018). Metaphor Detection by Deep Learning and the Place of Poetic Metaphor in Digital Humanities. *FLAIRS*, pp. 122-127.

Tsvetkov, Y., Boytsov, L., Gershman, A., Nyberg, E., & Dyer, C. (Junho de 2014). Metaphor detection with cross-lingual model transfer. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 248-258.

Wallace, E., Gardner, M., & Singh, S. (2020). Metaphor Detection Using Contextual Word Embeddings From Transformers. *Proceedings of the Second Workshop on Figurative Language Processing*, pp. 250-255.

Yang, Y.-H., & Chen, H. H. (1 de May de 2012). Machine Recognition of Music Emotion: A Review. *ACM Transactions on Intelligent Systems and Technology (TIST), 3*(3), pp. 1-30.

Esta página foi intencionalmente deixada em branco.

9. Anexos

Neste anexo são apresentadas as principais bibliotecas utilizadas ao longo do desenvolvimento deste projeto. Estas ferramentas foram essenciais para a implementação dos diferentes modelos de processamento de linguagem natural, bem como para a manipulação de dados, extração de *features* e avaliação dos resultados.

9.1. Anexo A – Bibliotecas importadas em python

a. pandas
b. numpy
c. re
d. string
e. nltk
f. nltk.corpus.stopwords
g. nltk.stem.PorterStemmer
h. sklearn.model_selection
i. train_test_split
j. StratifiedKFold
k. cross_val_predict
l. cross_val_score
m. TfidfVectorizer
n. LogisticRegression
o. RandomForestClassifier
p. SVC
q. sklearn.metrics
r. classification_report
s. confusion_matrix
t. accuracy_score
u. ConfusionMatrixDisplay
v. f1_score
w. sklearn.preprocessing

x. StandardScaler

y. LabelEncoder

z. Pipeline

aa. ReliefF

bb. torch

cc. torch.utils.data

dd. DataLoader

ee. TensorDataset

ff. AdamW

gg. torch.nn

hh. transformers

ii. BertTokenizer

jj. BertModel

kk. RoBERTaTokenizer

ll. RoBERTaModel

mm. RoBERTaForSequenceClassification

nn. set_seed

oo. get_linear_schedule_with_warmup

pp. Sequential

qq. tensorflow.keras.layers

rr. Dense

ss. Dropout

tt. EarlyStopping

uu. XGBClassifier

vv. matplotlib.pyplot

ww. seaborn

xx. tqdm

yy. random

zz. collections.Counter