UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Mensuração de drift conceitual de avaliações de produtos da Amazon utilizando *text embeddings*

Miguel Del Ben Galdiano

Trabalho de Conclusão de Curso MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Miguel Del Ben Galdiano

Mensuração de drift conceitual de avaliações de produtos da Amazon utilizando *text embeddings*

Trabalho de conclusão de curso apresentado ao Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Opção: Data Science

Orientadora: Prof. Dr. Edson Takashi Matsubara

Versão original

São Carlos

2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

S856m

Del Ben, Miguel Galdiano
Mensuração de drift conceitual de avaliações de produtos da Amazon utilizando *text embeddings* / Miguel Del Ben Galdiano ; orientador Edson Takashi Matsubara. – São Carlos, 2023.
45 p. : il. (algumas color.) ; 30 cm.

Monografia (Graduação em ...) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. LaTeX. 2. abnTeX. 3. Classe USPSC. 4. Editoração de texto. 5. Normalização da documentação. 6. Tese. 7. Dissertação. 8. Documentos (elaboração). 9. Documentos eletrônicos. I. Matsubara, Edson Takashi, orient. II. Título.

Miguel Del Ben Galdiano

Measurement of conceptual drift in Amazon product reviews using text embeddings

Term paper submitted to the Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, in partial fulfillment of the requirements for the degree of the Specialist in Artificial Intelligence and Big Data.

Advisor: Prof. Dr. Edson Takashi Matsubara

Original version

São Carlos

2023

Folha de aprovação em conformidade
com o padrão definido
pela Unidade.
No presente modelo consta como
folhadeaprovacao.pdf

AGRADECIMENTOS

Agradeço a minha mãe, tanto pelo apoio financeiro quanto pelo apoio moral. Ficar acordada até tarde para garantir que eu entregasse o trabalho há tempo não é para qualquer um.

Ao meu padastro, por ter comprado energético e Pepsi sem eu nem precisar pedir. Alimentou muitas noites mal dormidas.

Ao meu pai, que me chamou para ir no bar quando eu deveria estar trabalhando nesse projeto. Paz de espírito também é importante.

A minha namorada, por ter lidado com minhas emoções instáveis. Não é fácil perguntar se alguém está bem quando perguntar se ela está bem faz ela não ficar bem.

Agradeço ao Prof. Edson Takashi Matsubara pela orientação e suporte.

Por último agradeço meu segundo monitor, sem o qual teria sido impossível escrever esse trabalho.

RESUMO

DEL BEN, M. G. **Mensuração de drift conceitual de avaliações de produtos da Amazon utilizando *text embeddings***. 2023. 45p. Monografia (Trabalho de Conclusão de Curso) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

A manutenção de modelos de aprendizado de máquina para tarefas de linguagem natural é um tópico de especial atenção atualmente, com o aumento de aplicações disponíveis para o público. Com esse intuito, é importante acompanhar e mensurar o *drift* conceitual que pode ocorrer com a eventual mudança dos dados interpretados por esse modelo. Para o contexto de PLN, uma possível estratégia de mensuração envolve a extração de *text embeddings* e aplcação de métricas de distância para acompanhar a variação dessas representações. Neste projeto, aplicamos esse método para a evolução ao longo dos anos de avaliações de produtos na Amazon de diferentes categorias: *Luxuary Beauty*, *Musical Instruments* e *Office Products*. Observamos a presença de *drift* conceitual para a categoria *Office Products*, o que reflete a revolução tecnológica por qual essa área passou durante os anos avaliados.

Palavras-chave: PLN. BERT. DistilBERT. Representação textual. Drift.

ABSTRACT

DEL BEN, M. G. **Measurement of conceptual drift in Amazon product reviews using text embeddings**. 2023. 45p. Monograph (Conclusion Course Paper) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

The maintenance of machine learning models for natural language tasks is currently a topic of particular attention due to the increasing availability of applications for the public. It is crucial to monitor and measure conceptual drift that may occur with any changes in the data interpreted by these models. For the context of Natural Language Processing (NLP), one potential measurement strategy involves extracting text embeddings and applying distance metrics to track the variation of these representations. In this project, we applied this method to assess the evolution over the years of product reviews on Amazon from different categories: Luxury Beauty, Musical Instruments, and Office Products. We observed the presence of conceptual drift in the Office Products category, reflecting the technological revolution that this area underwent during the assessed years.

Keywords: NLP. BERT. DistilBERT. Text embedding. Drift.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

PLN	Processamento de Linguagem Natural
RN	Rede Neural
FNN	*Feedforward Neural Network*
RNN	*Recurrent Neural Network*

SUMÁRIO

1 INTRODUÇÃO

Algoritmos de aprendizado de máquina e inteligência artificial são tópicos que vêm crescendo dentro do vocabulário público há anos. Uma série de diferentes modelos foram desenvolvidos, possuindo uma série de aplicações em áreas distintas: mercado financeiro, saúde, redes sociais, comércio, infraestrutura e logística, dentre outros. Suas aplicações tornam-se cada vez mais presentes na realidade da vivência cotidiana (2).

Dentre desse contexto, o desenvolvimento de modelos para o processamento de linguagem natural tem um destaque especial. Uma grande parcela do conteúdo produzido pela humanidade e sua principal forma de comunicação é no formato de dados textuais. Logo, a capacidade de compreendê-los e extrair análises relevantes a partir de textos é uma tarefa comumente estudada no contexto de aprendizado de máquina (3).

Atualmente, modelos de linguagem natural já são utilizados pelo público leigo, sendo integrados em diversas áreas. Um claro exemplo disso é o ChatGPT, que hoje é praticamente ubíquo para uma considerável parcela da população. Considerando a capacidade destes modelos e sua aplicações, torna-se indispensável garantir que sua performance seja continuamente monitorada e mantida.

Portanto, é quase inevitável que um modelo que tenha sido colocado em produção e disponibilizado para uso diário enfrente o que é chamado de *drift*. A realidade na qual essas ferramentas não são estáticas, de forma que o contexto em que elas foram treinadas evoluam com o tempo. Os conceitos e contexto nos quais o algoritmo foi treinado pode mudar com o tempo, resultando num decaimento da performance do modelo.

Logo, o acompanhamento e mensuração do *drift* de um modelo é um aspecto essencial do ciclo de vida prático de uma ferramenta desse tipo. Este projeto visa avaliar e aplicar estratégias de mensuração de *drift* dentro do contexto de tarefas de processamento de linguagem natural.

Para esse objetivo, propomos a análise de sentimento de avaliações de produtos de diferentes categorias vendidos na Amazon. Os dados utilizados desta base possuem uma distribuição temporal considerável, sendo assim possível de capturar possíveis variações conceituais com o tempo. Além disso, é possível comparar a existência e intensidade do *drift* para diferentes categorias.

O presente projeto tem como objetivo identificar e quantificar a evolução da variação conceitual dos textos de avaliações de produtos de diferentes categorias ao longo dos anos, fazendo uso de métricas apropriadas para o caso de processamento de linguagem.

2 FUNDAMENTAÇÃO TEÓRICA

Este cápitulo apresenta os principais fundamentos necessários para o desenvolvimento do projeto, com especial ênfase no tópico de processamento de linguagem natural (PLN), essencial para compreensão do trabalho.

2.1 Processamento de Linguagem Natural

Processamento de linguagem natural é uma área das ciências computacionais cujo objetivo é a utilização de técnicas e algoritmos para aprender, entender e reproduzir a linguagem humana, introduzida na década de 1950 como a intersecção entre inteligência artificial e linguística (4). Atualmente, PLN é utilizada em uma série de diferentes contextos: tradução automática, mineração de texto, predição de escrita, filtros de e-mail, monitoração de redes sociais, etc.

Inicialmente, estratégias de PLN envolviam a codificação concreta das regras gramaticais e vocabulário de uma dada linguagem. Rapidamente percebeu-se a limitação de tal técnica, principalmente devido ao uso de metáforas e outras figuras de linguagem, resultando em textos cuja compreensão semântica era muito dificultada.

Com o passar dos anos, principalmente na década de 80, houve um deslocamento para a utilização de técnicas de análise estatística para o desenvolvimento de ferramentas de PLN. Para capturar as estruturas tipicamente presentes na linguagem natural, vastas quantidades de textos foram utilizados em estratégias de aprendizado de máquina.

Hoje, os modelos de PLN mais refinados fazem uso de complexas redes neurais treinadas utilizando conjuntos de dados compostos por uma enorme quantidade e variedade de textos. Nas próximas seções, descreveremos o funcionamento de alguns dos principais modelos de linguagem, e apresentaremos os algoritmos de principal relevância.

2.2 Modelos de Linguagem

Modelos de linguagem são distribuições probabilísticas definidas sobre uma dada sequência ou conjunto de palavras (5). Matematicamente, para uma dada sequência de m palavras, o modelo define uma probabilidade $P(w_1, ..., w_m)$ para a toda sequência. Logo, o modelo também pode ser capaz de prever qual seria a próxima palavra, atribuindo a ela uma probabilidade $P(w_{m+1})$.

As capacidades de atribuir uma probabilidade a uma dada sequência e de prever qual seria a próxima palavra coerente com as demais são ferramentas úteis para diversos cenários presentes em nosso cotidiano: reconhecimento de voz, correção gramatical, tradução

automática e geração de texto, entre outros. Ao digitar uma mensagem, pesquisar algo, escrever um texto, em todas essas situações, há o uso de modelos de linguagem.

O desenvolvimento e obtenção de tais modelos é uma área de grande importância para o processamento de linguagem natural. Nas próximas seções, apresentaremos alguns exemplos dos principais modelos utilizados, detalhando seu funcionamento e eficácia. Essas seções são baseadas no conteúdo presente no livro *Speech and Language Processing*, dos autores Dan Jurafsky e James H. Martin, 2023 (5).

2.2.1 N-gramas

Consideremos um cenário onde queremos obter a probabilidade de uma palavra w dado um histórico h, isso é, desejamos calcular $P(w|h)$. Suponha que o histórico seja "o garoto foi para a casa de" e queremos saber qual a probabilidade da próxima palavra ser "ônibus":

$$P(\text{ônibus}|\text{o garoto foi para a casa de}). \tag{2.1}$$

Uma forma de estimar esse valor seria por meio do método de contagem relativa: com um corpus suficientemente grande, contamos o número de ocorrência da frase "o garoto foi para a casa de" e o número de ocorrências dessa mesma frase seguida por "ônibus". Com isso, podemos calcular

$$P(\text{ônibus}|\text{o garoto foi para a casa de}) = \frac{C(\text{o garoto foi para a casa de ônibus})}{C(\text{o garoto foi para a casa de})}, \tag{2.2}$$

isto é, de todas as vezes que observamos h, quantas vezes ela foi procedida por w.

Apesar de ser um método eficaz para alguns casos, a realidade é que não há um *corpus* grande o suficiente para englobar todas as possíveis combinações de termos. Isso faz com que nosso modelo não seja capaz de estimar a probabilidade em muitos casos. Dessa forma, torna-se necessário desenvolver outro método para estimar a probabilidade desejada.

Consideremos uma sequência n de palavras w_1, ..., w_n, também representada por $w_{1:n}$. Suponha que desejamos determinar a probabilidade de termos essa dada sequência, $P(w_1, ..., w_n)$. Para calcular esse valor, podemos fazer uso da regra de cadeia da probabilidade, obtendo:

$$\begin{aligned} P(w_{1:n}) &= P(w_1)P(w_2|w_1)P(w_3|w_{1:2})\dots P(w_n|w_{1:n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_{1:k-1}). \end{aligned} \tag{2.3}$$

A equação acima mostra que é possível obter a probabilidade de uma dada sequência de palavras pelo produto de um número de diferentes probabilidades condicionais. Entretanto, essa propriedade não resolve o nosso problema: prever uma palavra dada um

longo histórico. A linguagem está em constante evolução, tornando extremamente provável que nos deparemos com um contexto totalmente novo.

Uma simples intuição que pode simplificar nosso problema é a seguinte: em vez de consideramos todo o histórico que antecede uma palavra, consideramos apenas a palavra precedente. Matematicamente, isso significa que assumimos que a seguinte aproximação é válida:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1}). \tag{2.4}$$

Essa suposição é chamada de suposição de Markov. Modelos de Markov são uma classe de modelos probabilísticos que assumem que podem prever probabilidade futuras sem olhar para todo o histórico relevante. A Equação 2.4 se refere a um bigrama, o caso no qual consideramos apenas uma palavra antecedente. Podemos generalizar para um N-grama utilizando a seguinte equação:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-N+1:n-1}). \tag{2.5}$$

As equações acimas fornecem os termos necessários para a construção de um modelo de N-grama, capazes de serem utilizados para aplicação de processamento de linguagem. O próximo passo é estimar o valor das probabilidades acima. Podemos estimar esse valor fazendo a contagem a partir de um corpus e normalizando esse valor:

$$P(w_n|w_{n-N+1:n-1}) = \frac{C(w_{n-N+1:n-1}\ w_n)}{C(w_{n-N+1:n-1})}, \tag{2.6}$$

ou seja, a probabilidade é dada pela razão entre a frequência observada de uma dada sequência $(w_{n-N+1:n-1}\ w_n)$ e a frequência observada do prefixo $(w_{n-N+1:n-1})$.

Apesar de modelos de n-grama serem capazes de obter forte poder de predição de palavras, eles ainda apresentam certas limitações: não lidam bem com históricos muito extensos e tem dificuldade em generalização de contextos similares, mas não idênticos. Logo, há a motivação de utilizar ferramentas diferentes para obtenção de modelos de linguagem. Na próxima seção, discutiremos sobre uma delas: redes neurais.

2.2.2 Redes Neurais

Redes neurais (RNs) são uma ferramenta computacional fundamental no contexto de processamento de linguagem natural. Construídas a partir do neurônio de McCulloch-Pitts (6), redes neurais modernas são compostas por redes de pequenas unidades de processamento, as quais recebem um vetor de entrada e retornam um único valor de saída.

Redes neurais apresentam um aspecto de muita utilidade para tarefas de PLN. A partir de dados crus, ao longo do processo computacional, as redes são capazes de aprender características relevantes dos dados, aprimorando seu algoritmo. Logo, redes neurais são

especialmente eficientes para tarefas que oferecem suficiente quantidade de dados para esse aprendizado.

Inciaremos com a apresentação da rede mais simples, a rede neural *feedforward* (*feedforward neural network*, FNN). Ela é composta por múltiplas camadas onde cada unidade de processamento está conectada sem ciclos: as saídas das unidades em uma camada são passadas apenas para a próxima camada e não para camadas anteriores (Figura 1).

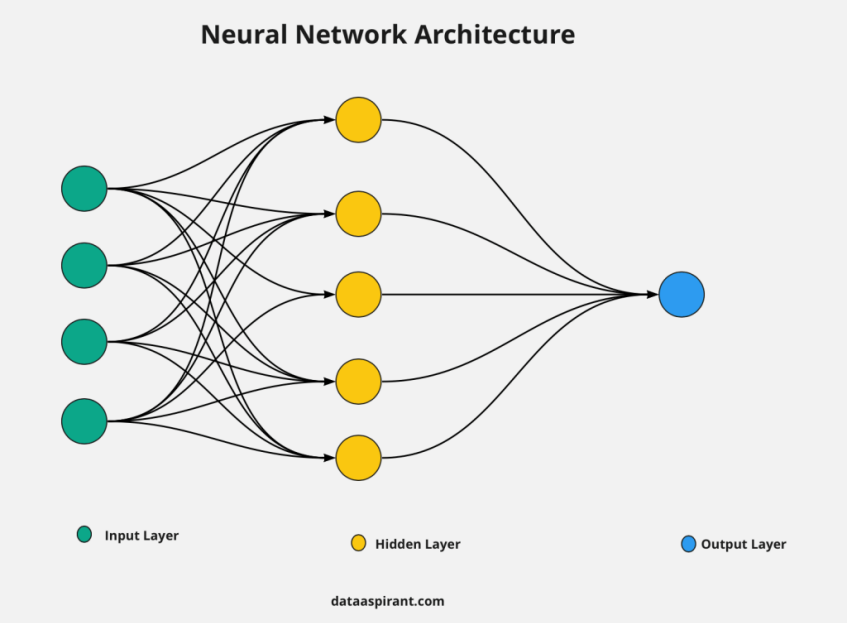

Figura 1 – Arquitetura de uma rede neural *feedforward*. Possui uma camada de entra (*input layer*, uma camada de saída (*output layers*) e uma ou mais camadas ocultas (*hidden layers*). Por simplicidade, a figura apresenta uma única camada oculta. Fonte: dataaspirant.com

Apesar de simples, é possível utilizar uma rede desse tipo para o desenvolvimento de um modelo de linguagem (7). A rede recebe como entrada em um instante t a representação de um dado número de palavras prévias (w_{t-1}, w_{t-2}, etc.) e então retorna uma distribuição de probabilidade das próximas possíveis palavras.

De forma análoga aos modelos de n-grama, nosso modelo neural aproxima a probabilidade dado todo o contexto prévio considerando apenas um número N de palavras anteriores:

$$P(w_t|w_1, \ldots, w_{t-1}) \approx P(w_t|w_{t-N+1}, \ldots, w_{t-1}) \tag{2.7}$$

Uma das principais vantagens destes modelos é sua capacidade de generalização para casos com palavras inéditas. Isso é possível devido ao método que as redes neurais utilizam para representar palavras: *word embeddings*. Este método envolve representar palavras como vetores, de forma que palavras com significados semânticos próximos também tenham vetores próximos no espaço vetorial.

Para entender como esse método pode útil para a generalização, consideremos que nosso *corpus* utilizado para treinamento contenha a seguinte frase:

"Eu preciso ver se o gato comeu.",

não contenha nenhum caso em que a palavra "cachorro" seja seguida por "comeu". Suponha que desehamos prever a palavra seguida do contexto "Eu não sei se o cachorro". Para um modelo de n-grama, isso poderia ser um complicador. Um modelo de rede neural entretanto, sabendo que "gato" e "cachorro" possuem representações vetoriais próximas, seria capaz de generalizar a partir do contexto de "gato" para atribuir uma alta probabilidade a "comeu".

Apesar das vantagens descritas acima, modelos de linguagem utilizando FNNs possuem dificuldade em capturar uma das principais características da linguagem: sua temporalidade. O modelo descrito acessa simultaneamente um contexto prévio definido por um número fixo de palavras de interesse. Para predições seguintes, o contexto é deslizado para incorporar a nova saída obtida pelo processo e então realiza uma nova previsão, independente da anterior.

Para resolver esse problema, podemos fazer uso de redes neurais recorrentes (*recurrent neural networks*, RNNs) cuja arquitetura é capaz de lidar com o aspecto temporal da linguagem.

2.2.3 Redes Neurais Recorrentes

A principal diferença na arquitetura de RNNs é a presença de ciclos: as saídas das unidades de processamento em uma camada também são passadas para camadas anteriores. Isso significa que o valor de saída de uma rede recorrente num determinado instante de tempo t depende de seu resultado no instante anterior $t - 1$ (Figura 2).

Dado essa arquitetura, as camadas ocultas da rede servem como um tipo de memória, que registra procedimentos anteriores e informa as decisões efetuadas em instantes seguintes. Em comparação com FFNs, que sempre observavam um contexto prévio de tamanho pré-definido, o contexto presente nas camadas ocultas pode, teoricamente, conter informação que se refere até ao começo da sequência.

Esses atributos de RNNs são de grande utilidade no contexto de modelos de linguagem. Modelos de RRN processam a sequência de entrada uma palavra por vez, tentando prever a próxima palavra utilizando a palavra atual e o estado anterior da camada oculta (8). Logo, tais modelos apresentam certas vantagens quando comparados aos apresentados anteriormente: não possuem o contexto limitado de n-gramas, nem o contexto prévio fixado de modelos de FFN possuem.

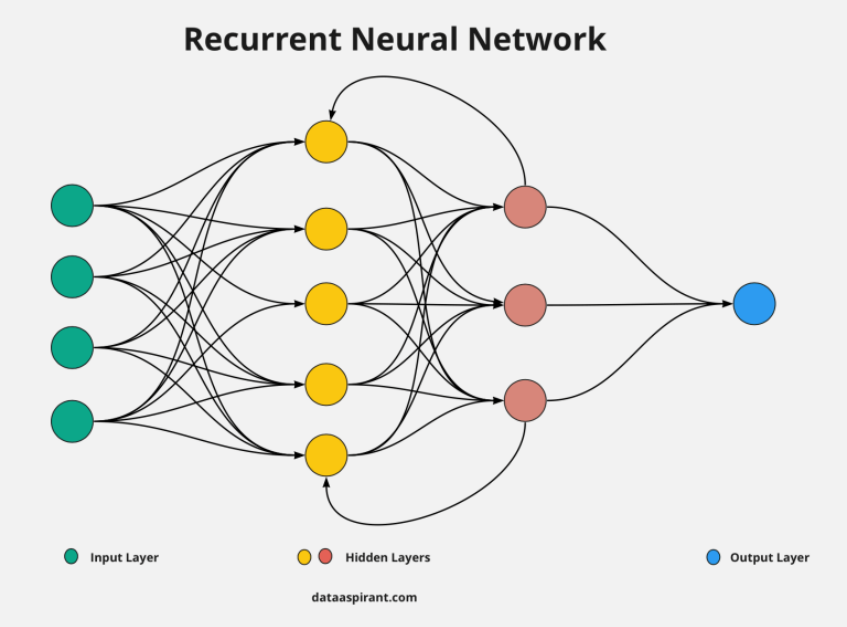

Figura 2 – Arquitetura de uma rede neural recorrente (RNN). A principal diferença é a característica cíclica representada pelas setas que indicam que a saída de uma camada é utilizada com entrada de uma cada anterior. Fonte: dataaspirant.com

2.3 Transformadores

Por muito tempo, modelos de linguagem construídos utilizando RNNs foram o estado da arte, mas eles não eram isentos de limitações. Sua natureza sequencial necessariamente impossibilitava estratégias de paralelização para treinamento, um problema que torna-se crítico quando consideramos sequências de texto longos.

Outra limitação diz respeito a memória da RNN em relação a elementos anteriores muito distantes. Tipicamente, quanto mais distante for um termo passado, menor é seu impacto durante o processo de previsão das próximas palavras. Para sanar essa limitação, foram introduzidos mecanismos de atenção, que permitiam que o modelo considerasse devidamente palavras passadas, independente de sua distância (9).

Considerando o impacto dos mecanismos de atenção na performance desse modelos, em 2017 foi proposta uma arquitetura que fazia uso apenas de atenção, sem a capacidade recorrente de RNNS (1). Esse novo modelo foi chamado de *transformer* e depende exclusivamente desses mecanismos para traçar relações globais entre os elementos de entrada e saída.

Apesar de ter sido desenvolvido para tarefas tipicamente sequenciais de PLN, como tradução e resumo de texto, sua arquitetura não é intrinsecamente sequencial como no caso de RNNs. *Transformers* utilizam atenção para extrair informações relevantes sobre o contexto de uma palavra, e assim criando sua representação vetorial (*word embedding*. Logo, *transformers* são naturalmente mais fáceis de serem paralelizados, aumentando a

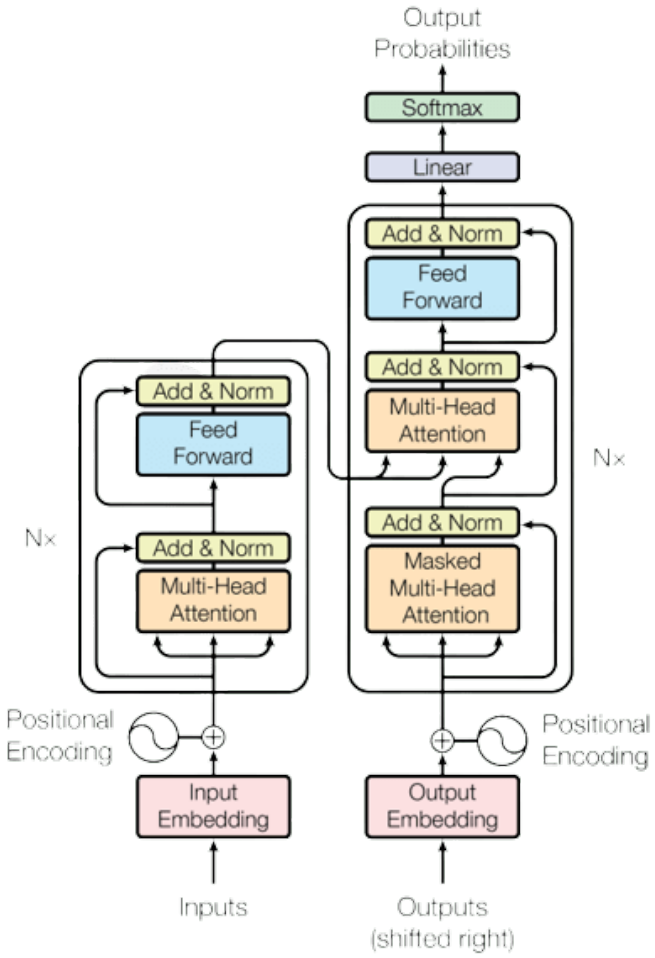

Figura 3 – Arquitetura de um *transformer*. Baseada na estrutura *coder-encoder* (codificador-decodificador), ambos são compostos por estrutura de módulos que podem ser empilhados uma sobre a outra. Fonte: *Attention Is All You Need* (1).

velocidade de treinamento e possibilitando o desenvolvimento de modelos de linguagem massivos.

2.3.1 GPT

Transformadores geradores pré-treinados (*generative pre-trained transformers*, GPT) são um tipo de modelo grande de linguagem (*large language model* LLM), pré-treinados em uma grande quantidade de textos não catalogados.

Inicialmente, GPT foi introduzido em 2018 pela OpenAI (10), o primeiro de uma série que seriam desenvolvida pela empresa. Cada um desses modelos foi desenvolvido com uma maior número de parâmetros e conteúdo para o treinamento que o seus anteriores. Como ponto de referência, o GPT-1 tinha 117 milhões de parâmetros. Em comparação, é estimado que a sua versão mais recente, o GPT-4, tenha 1,7 trilhões de parâmetros.

Esses modelos tipicamente servem como base para a construção de novos modelos

com objetivos e tarefas mais específicas. Um exemplo conhecido pela população é o ChatGPT, um serviço de *chatbot* construído utilizando um modelo GPT.

2.3.2 BERT

BERT (*Bidirectional Encoder Representations from Trasnformer*) se refere a uma família de modelos introduzida em 2018 por pesquisadores do Google (11). A arquitetura de tais modelos é composta apenas por codificadores, de forma distina daquela inicialmente introduizda para transformadores.

Como é o caso para o GPT, BERT também é um modelo pré-treinado, considerando duas tarefas distintas: modelagem de linguagem, na qual o objetivo é prever a próxima palavra dada um contexto e predição de frases, na qual o objetivo é determinar se duas frases aleatórias aparecem sequencialmente no corpus utilizado.

Para o presente trabalhado, foi utilizado uma versão reduzida deste modelo, chamado DistilBERT (12). O número de parâmetros utilizados é reduzido por um fator de 40% , mas é capaz de manter 95% da performance quando comparado ao modelo original.

2.4 Drift

Drift, dentro do contexto de aprendizado de máquina, se refere ao decaimento da performance de um dado modelo com o tempo após sua publicação. De uma forma geral, ocorre quando a os dados variam com o tempo de forma que invalidam o modelo inicialmente treinado.

Drifts podem ser classificados em diferentes tipos. O de interesse para esse projeto é o chamado *drift* conceitual, que ocorre quando o contexto dos dados a serem analisados evoluiu e muda com o tempo, resultado em um novo contexto experimental para qual o modelo não foi adequadamente treinado.

2.4.1 Detecção de Drift

Para o caso de *drift* conceitual, podemos detectar se houve uma mudança nos nossos dados considerando a evolução de sua distribuição com o tempo. Dessa forma, existe uma série de testes estatísticos que são capazes de avaliar se duas amostras fazer parte de uma mesma distribuição, dentre eles:

- **Teste Kolmogorov-Smirnov (KS):** teste estatístico não-paramétrico utilizado para determinar se duas amostras são provenientes da mesma distribuição.
- **Índice de Estabilidade Populacional (PSI):** medida estatística utilizada para comparar a distribuição de uma variável categórica em duas amostras distintas.

- **Método Page-Hinkley:** método estatístico que detecta variações no valor médio de uma série de dados ao longo do tempo.

Os métodos descritos acima são úteis e tipicamente utilizados para o monitoramento de *drift* de modelos em produção, podendo ser usados como indicadores de que o modelo precisa ser atualizado ou corrigido. Entretanto, essas técnicas tem como foco principal dados estruturados, o que dificulta sua aplicação para o caso de modelos que lidam com textos ou imagens.

Para lidar com essa limitação, dados não-estruturados são transformados em representações vetoriais. Para o caso específico de análise textuais, utilizamos modelo de PLN para extrair seus *embeddings*. Com essas representações, podemos utilizar métricas que medem a diferença entre os *embeddings* de duas amostras, como distância euclidiana e distância de cosseno.

3 METODOLOGIA

3.1 Base de Dados

Para a análise de evolução temporal de conceitos, foi utilizada a base de dados *Amazon Review Data (2018)* (13). Esses dados consistem nas avaliações sobre diferentes produtos feitas por usuários entre o período de maio de 1996 até Outubro de 2018. O conjunto de dados possuí os seguintes atributos:

- **`revierID`:** ID do avaliador (ex: A2SUAM1J3GNN3B).
- **`asin`:** ID do produto (ex: 0000013714).
- **`reviewerName`:** nome do avaliador.
- **`vote`:** número de votos de utilidade da avaliação.
- **`style`:** um dicionário da metadata do produto (ex: "Formato": "Capa dura").
- **`reviewText`:** texto da avaliação.
- **`overall`:** nota do produto.
- **`summary`:** resumo da abaliação.
- **`unixReviewTime`:** instante em que foi feita a avaliação (horário UNIX).
- **`reviewTime`:** instante em que foi feita a avaliação.
- **`image`:** imagens pulicadas por usuários após terem recebido o produto.

Para o presente projeto, apenas os atributos `reviewText`, `overall` e `reviewTime` foram utilizados. Os dados são separados em subconjuntos definidos pela categoria dos produtos, totalizando 29 subconjuntos diferentes. A Tabela 1 lista quais foram as categorias utilizadas durante o trabalho, assim como o número de avaliações presentes:

Tabela 1 – Número de avaliações por categoria de produto

Categoria de Produto	Conjunto Total	Subconjunto Denso
Luxury Beauty	574.628	34.278
Musical Instruments	1.512.530	231.392
Office Products	5.581.313	800.357

Fonte: Elaborada pelos autores.

Para cada categoria, há duas versões da base de dados disponíveis: o conjunto completo e um subconjunto denso, reduzido de forma que cada usuário e produto tenha pelo menos cinco avaliações. Devido a limitações de tempo de processamento, o subconjunto denso foi utilizado para as seguintes categorias: *Musical Instruments* e *Office Products*.

3.2 Pré-processamento

Os passos descritos nessa seção são aplicados separadamente para cada categoria de produtos analisada. Inicialmente, uma limpeza de dados é realizada para garantir que o atributo `reviewText` contenha apenas textos. Caso uma avaliação contenha algum tipo de arquivo adicional, ela é removida. Em seguida, uma nova coluna é criada, chamada `label`, de forma a classificar as avaliações em dois grupos: positivas e negativas. A nova coluna é definida da seguinte forma:

$$\texttt{label} = \begin{cases} 1, \text{ se } \texttt{overall} \geq 3 \\ 0, \text{ se } \texttt{overall} < 3 \end{cases} \tag{3.1}$$

Em seguida, dividimos os dados considerando a data indicada pela coluna `reviewTime`, de forma a criar um subconjunto por ano. Para todos os casos, a divisão é inicialmente feita considerando todos os anos no intervalo entre 1996 e 2018. Entretanto, algumas das categorias analisadas possuem avaliações apenas a partir de um determinado ano. Nesse caso, desconsideramos os anos anteriores.

Por último, como é padrão para tarefas de PLN, cada avaliação passa por um processo de "tokenização". Para isso, foi utilizado um tokenizador DistilBERT pré-treinado.

3.3 Extração dos *Embeddings*

Para extração das representação vetoriais das avaliações (*text embeddings*), utilizamos o modelo DistilBERT para classificação de textos, implementado por meio da biblioteca `Transformers` (12, 14). Para cada categoria, um ano de referência para comparação é selecionado, tendo como critério o ano mais antigo com uma quantidade estatisticamente relevante de avaliações.

O modelo é treinado utilizando um subconjunto do ano de referência como dados para treinamento, com o objetivo de classificar as avaliações entre positivas e negativas. Em seguida, os *embeddings* são extraídos para os conjuntos separados de cada ano.Os *embeddings* gerados são vetores numéricos não-normalizados de 128 dimensões.

Uma segunda extração é realizada para o ano de referência, considerando nesse caso os demais dados do ano que não foram utilizados durante o treinamento. Esse conjunto sera utilizado com o intuito de servir como ponto de comparação para os demais anos.

3.4 Métricas de Distância

Para quantificar a evolução do drift conceitual ao longo dos anos para cada categoria de produto, foram utilizadas duas métricas de distância: distância euclidiana e distância de cosseno. Para ambos os casos, um vetor médio dos *embeddings* é obtido para cada ano utilizando a seguinte fórmula:

$$\mathbf{v}_a = \frac{1}{n} \sum_{i=1}^{n} \mathbf{w}_i^a, \tag{3.2}$$

onde $\mathbf{w}_i^a$ é o *embedding* de cada avaliação no ano a e n o número total de avaliações. Em seguida, utilizando os vetores médios obtidos, a distância euclidiana entre o vetor médio de um ano a e o vetor médio do ano de referência r é dado por:

$$d_a^{\text{euc}} = \sqrt{\sum_{i=1}^{128} \left(v_i^a - v_i^r\right)^2} \tag{3.3}$$

sendo v_i^a e v_i^r os elementos que compõe, respectivamente, os vetores $\mathbf{v}_a$ e $\mathbf{v}_r$. De forma análoga, a distância do cosseno é calculada usando a seguinte fórmula:

$$d_a^{\cos} = 1 - \frac{\mathbf{v}_a \cdot \mathbf{v}_r}{\|\mathbf{v}_a\| \|\mathbf{v}_r\|} \tag{3.4}$$

A distância de cosseno é definida de tal forma que é igual a 0 quando o ângulo entre os vetores é 0° e tem valor máximo igual a 2 quando o ângulo é 180°. Como o principal objetivo é observar a variação relativa dessas distâncias, os valores obtidos para ambas as métricas são normalizados pela distância entre os dois subconjuntos do ano de referência.

4 RESULTADOS E DISCUSSÃO

Os resultados obtidos são apresentados nas figuras abaixo:

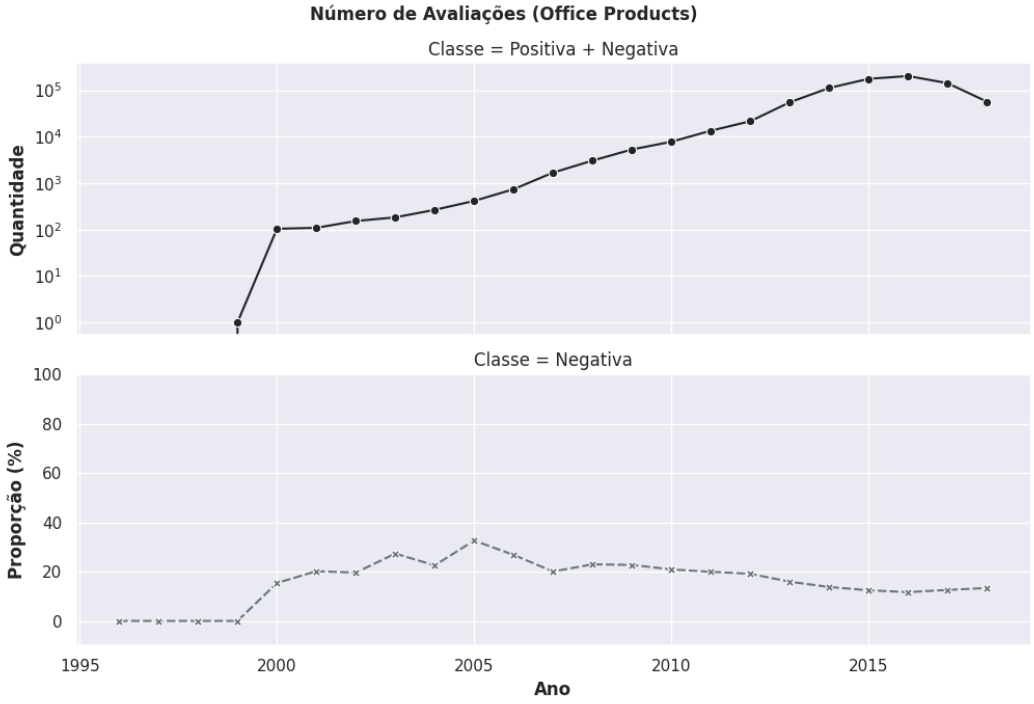

Figura 4 – Número de avaliações ao longo dos anos de produtos na categoria *Office Products* (material de escritório). Figura superior indica o número total de avaliações em escala logarítmica. Figura inferior indica a proporção das avaliações que são negativas.

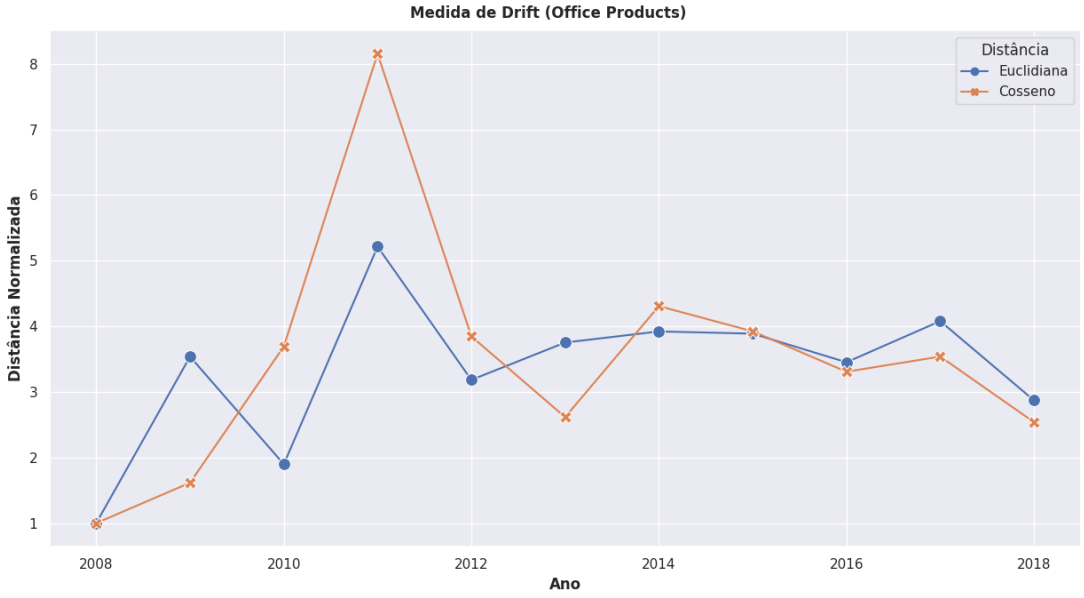

Figura 5 – Evolução de métricas de distâncias ao longo dos anos e produtos na categoria *Office Products* (material de escritório). Valores estão normalizados em relação ao ano mais antigo.

Podemos observar na Figura 5 um aumento de ambas as métricas de distância com o decorrer dos anos, possuindo um comportamento similar. Isso refletiria uma variação conceitual no conteúdo das avaliações deste produto. Isso pode ser explicado pela revolução tecnológica e digital que ocorreu dentre os anos de 2008 e 2018: aquilo que se encaixa como material de escritório mudou com a rápida incorporação de ferramentas tecnológicas no contexto profissional.

A interpretação para o caso de *Office Products* pode ser corroborada pelo comportamento observado em Figura 7 e Figura 9. Para estas categorias, não houve um aumento nas métricas de *drift*. Utilizando nossa interpretação inicial, também podemos desenvolver uma justificativa para esse comportamento: não houve nenhuma grande revolução nessas áreas neste período. O que qualifica um produto como sendo cosmético ou instrumento musical não mudou significativamente no intervalo temporal analisado.

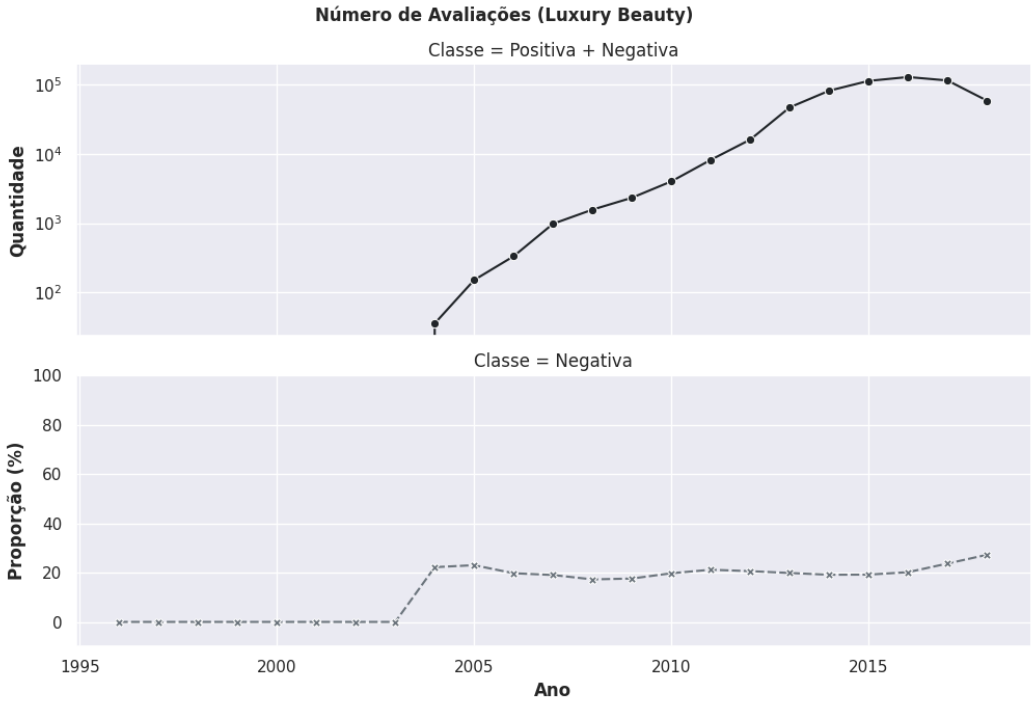

Figura 6 – Número de avaliações ao longo dos anos de produtos na categoria *Luxury Beauty* (cosméticos de luxo). Figura superior indica o número total de avaliações em escala logarítmica. Figura inferior indica a proporção das avaliações que são negativas.

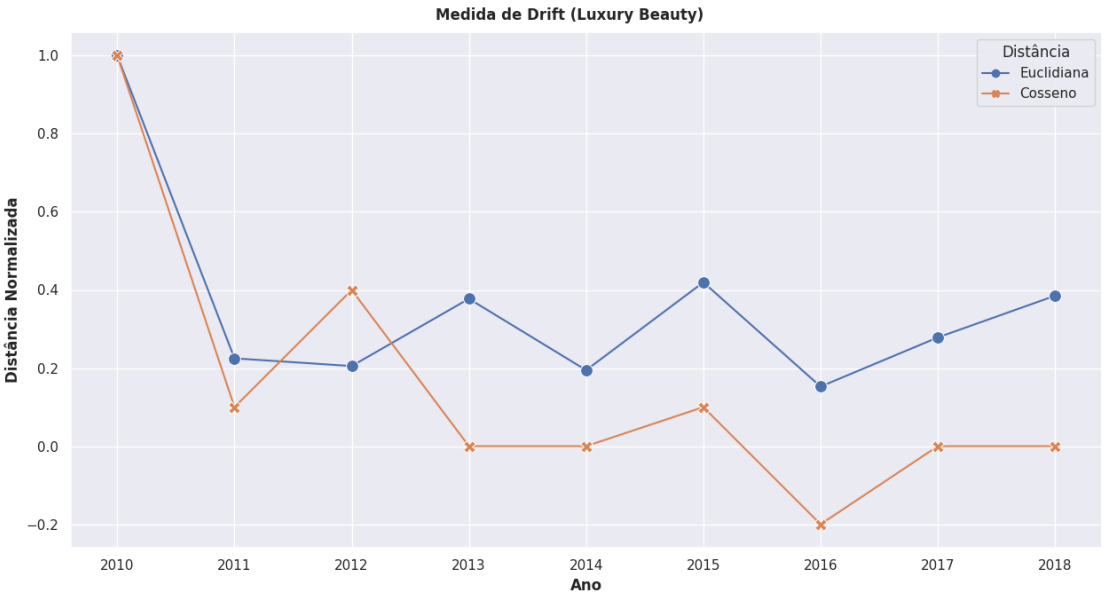

Figura 7 – Evolução de métricas de distâncias ao longo dos anos e produtos na categoria *Luxury Beauty* (cosméticos de luxo). Valores estão normalizados em relação ao ano mais antigo.

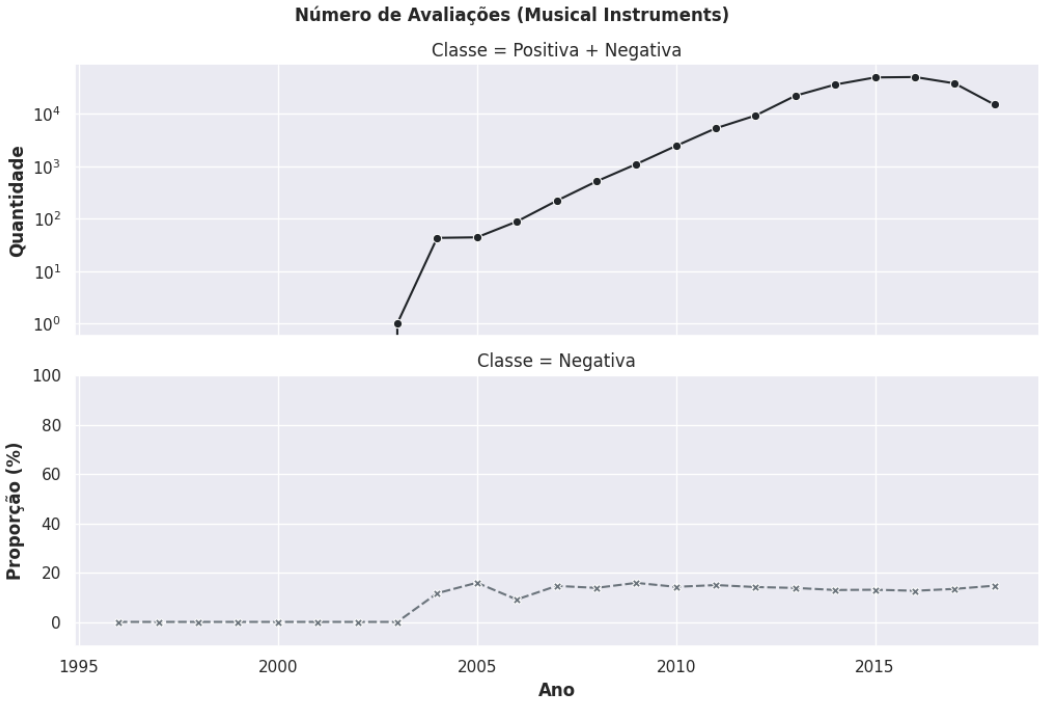

Figura 8 – Número de avaliações ao longo dos anos de produtos na categoria *Musical Instruments* (instrumentos musicais). Figura superior indica o número total de avaliações em escala logarítmica. Figura inferior indica a proporção das avaliações que são negativas.

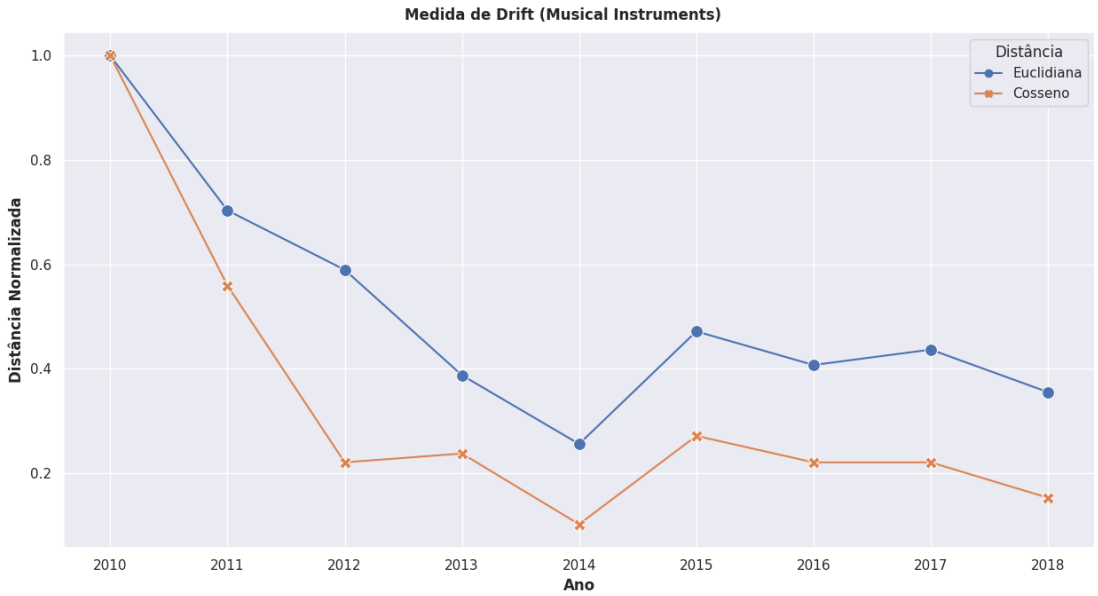

Figura 9 – Evolução de métricas de distâncias ao longo dos anos e produtos na categoria *Musical Instruments* (instrumentos musicais). Valores estão normalizados em relação ao ano mais antigo.

5 CONCLUSÃO

Com base nos resultado obtidos, podemos confirmar que foi possível identificar a presença de um *drift* conceitual nas avaliações de produtos da categoria *Office Products*. Essa evolução foi mensurada utilizando duas métricas de distância distintas, euclidiana e de cosseno, que obtiverem resultado equivalentes.

A evolução conceitual observada para esse caso pode ser explicada pela revolução tecnológica observada durante os anos de 2008 e 2018, e considerando como materiais de escritório foram impactados por ela. De forma análoga, podemos justificar a ausência de uma variação conceitual para as categorias *Luxury Beauty* e*Musical Instruments*, por não ter tido nenhuma grande mudança nessas áreas.

Desta forma, pode-se afirmar que o modelo DistilBERT utilizado para extração dos *text embeddings* foi capaz de extrair características semânticas dos textos avaliados, refletindo numericamente a mudança contextual esperada.

Como próximos passos, podemos implementar outras técnicas estatísticas em conjunto com as métricas utilizadas, como o Teste Kolmogorov-Smirnov, para tornar mais robusto e generalizável os resultados obtidos. Além disso, também abre-se a possibilidade de utilizar o *drift* mensurado para melhorar a performance de um modelo treinado em um contexto para tarefas de classificações em um contexto distinto.

REFERÊNCIAS

1 VASWANI, A. *et al.* **Attention Is All You Need**. 2023.

2 SARKER, I. Machine learning: Algorithms, real-world applications and research directions. **SN Computer Science**, v. 2, 03 2021.

3 PRIYA, B.; J.M, N.; THANGAVEL, G. An analysis of the applications of natural language processing in various sectors. *In*: _____. [*S.l.: s.n.*], 2021. ISBN 9781643682020.

4 NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. **Journal of the American Medical Informatics Association**, v. 18, n. 5, p. 544–551, 09 2011. ISSN 1067-5027. Disponível em: https://doi.org/10.1136/amiajnl-2011-000464.

5 JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing**. [*S.l.: s.n.*], forthcoming. Disponível em: https://web.stanford.edu/~jurafsky/slp3/. Acesso em: 31 jun. 2023.

6 MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, Springer Science and Business Media LLC, v. 5, n. 4, p. 115–133, dez. 1943. Disponível em: https://doi.org/10.1007/bf02478259.

7 BENGIO, Y. *et al.* A neural probabilistic language model. **J. Mach. Learn. Res.**, JMLR.org, v. 3, n. null, p. 1137–1155, mar 2003. ISSN 1532-4435.

8 MIKOLOV, T. *et al.* Recurrent neural network based language model. *In*: **Interspeech 2010**. ISCA, 2010. Disponível em: https://doi.org/10.21437/interspeech.2010-343.

9 BAHDANAU, D.; CHO, K.; BENGIO, Y. **Neural Machine Translation by Jointly Learning to Align and Translate**. 2016.

10 RADFORD, A.; NARASIMHAN, K. Improving language understanding by generative pre-training. *In*: . [*S.l.: s.n.*], 2018. Disponível em: https://api.semanticscholar.org/CorpusID:49313245.

11 DEVLIN, J. *et al.* **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. 2019.

12 SANH, V. *et al.* **DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter**. 2020.

13 NI, J.; LI, J.; MCAULEY, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. *In*: INUI, K. *et al.* (ed.). **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**. Hong Kong, China: Association for Computational Linguistics, 2019. p. 188–197. Disponível em: https://aclanthology.org/D19-1018.

14 WOLF, T. *et al.* **Transformers: State-of-the-Art Natural Language Processing**. Zenodo, 2022. Disponível em: https://doi.org/10.5281/zenodo.7391177.