



## **Processamento de Linguagem Natural**

### **Seminário 2** **Departamento de Computação**

**Profa. Dr. Helena Caseli**  
*Professor Responsável*

#### ***Integrantes do Grupo***

Enzo Dezem Alves	-----RA: 801743
Rodrigo Takizawa Yamauchi	-----RA: 800226
Thales Winther	-----RA: 802499
Luís Fernando do Carmo Lourenço	-----RA: 800210
Lucca Couto Barberato	-----RA: 800257
Matheus Menecucci	-----RA: 800310

# 1. Introdução

Neste seminário, propomos implementar um modelo de classificação que seja capaz de prever a nota (de 1 a 5 estrelas) atribuída por usuários a filmes, utilizando unicamente as avaliações textuais escritas por eles. Este problema se insere no domínio da análise de sentimentos e da inferência de opinião, áreas conhecidas e estudadas em PLN, especialmente em contextos onde há a necessidade de automatizar o entendimento da percepção do usuário sobre produtos, serviços ou conteúdos culturais, como filmes.

A tarefa pode ser descrita como uma classificação supervisionada de texto com múltiplas classes, onde cada instância é composta por um texto livre (a avaliação) e um rótulo discreto (número de estrelas). A motivação para essa abordagem é baseada em aplicações reais, como aquelas encontradas em plataformas como IMDb, Rotten Tomatoes e Letterboxd, onde milhões de usuários escrevem avaliações textuais acompanhadas de notas numéricas. Automatizar essa previsão não só pode auxiliar na moderação e detecção de inconsistências, como também pode ser utilizado em sistemas de recomendação, sumarização de opiniões e estudos de mercado [1].

O desafio central dessa tarefa está na subjetividade da linguagem natural. Uma mesma opinião pode ser expressa de diferentes formas, e diferentes usuários podem usar tons, ironias e expressões idiomáticas de maneira distinta para comunicar suas impressões. Além disso, a correspondência entre texto e nota pode não ser direta; por exemplo, uma resenha pode conter trechos negativos e ainda assim resultar em uma nota alta por consideração a aspectos técnicos do filme. Isso torna o problema significativamente mais complexo do que uma simples classificação binária (positivo vs. negativo), como a realizada em tarefas clássicas de análise de sentimentos [2].

Com essa tarefa buscamos não apenas aplicar os conhecimentos teóricos desenvolvidos ao longo da disciplina, como também explorar as dificuldades práticas de um problema realista, desde o tratamento de dados ruidosos até a avaliação da performance de modelos em tarefas subjetivas. Acreditamos que essa implementação pode oferecer contribuições relevantes para o entendimento da aplicabilidade de modelos de PLN em contextos cotidianos e comerciais.

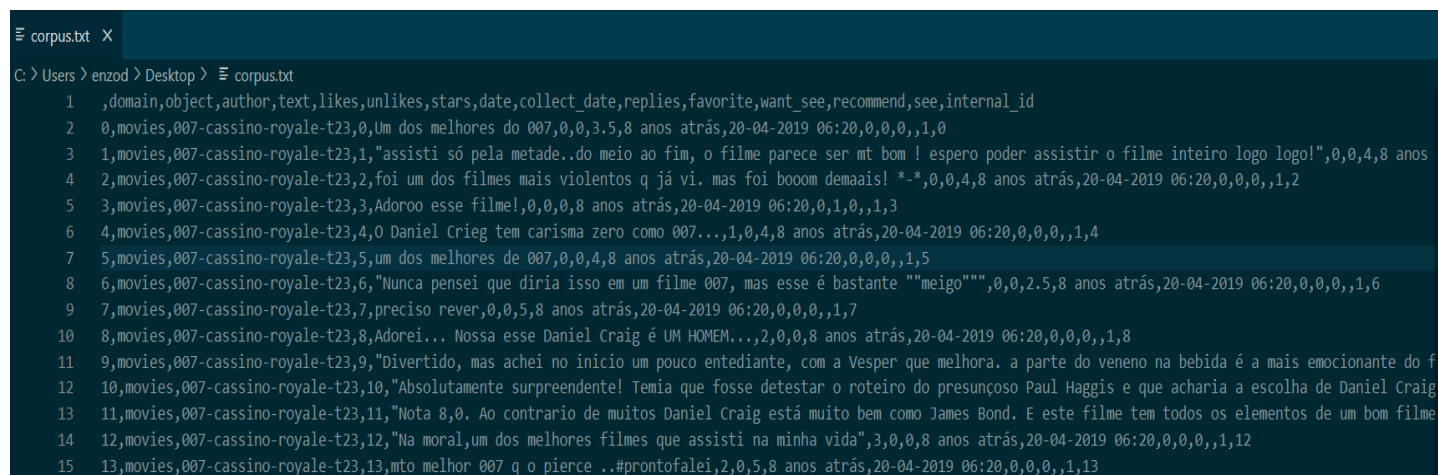
## 2. Dataset Escolhido - UTLCorpus

### 2.1 Visão Geral

- O dataset escolhido foi o recomendado no slide 23 da Aula 8. segue link:
  - <https://github.com/RogerFig/UTLCorpus/tree/master>
- Escolhemos este dataset por conter uma grande variedade de avaliações em português, com linguagem informal e espontânea, o que torna a tarefa mais desafiadora e realista. Além disso, ele está publicamente disponível e pronto para uso, facilitando a experimentação e a reprodutibilidade.
- Domínios:
  - **Movies:** opiniões/avaliações de filmes;
  - **Apps:** avaliações de aplicativos para smartphones.
- **Foco do projeto:** apenas as linhas com domain = "movies".

## 2.2 Estrutura do UTLCorpus (campos relevantes)

Amostra do arquivo corpus.csv



```
1 ,domain,object,author,text,likes,unlikes,stars,date,collect_date,replies,favorite,want_see,recommend,see,internal_id
2 0,movies,007-cassino-royale-t23,0,Um dos melhores do 007,0,0,3.5,8 anos atrás,20-04-2019 06:20,0,0,0,,1,0
3 1,movies,007-cassino-royale-t23,1,"assisti só pela metade..do meio ao fim, o filme parece ser mt bom ! espero poder assistir o filme inteiro logo logo!",0,0,4,8 anos
4 2,movies,007-cassino-royale-t23,2,foi um dos filmes mais violentos q já vi. mas foi booom demais! *-*,0,0,4,8 anos atrás,20-04-2019 06:20,0,0,0,,1,2
5 3,movies,007-cassino-royale-t23,3,Adoro esse filme!,0,0,0,8 anos atrás,20-04-2019 06:20,0,1,0,,1,3
6 4,movies,007-cassino-royale-t23,4,0 Daniel Crieg tem carisma zero como 007...,1,0,4,8 anos atrás,20-04-2019 06:20,0,0,0,,1,4
7 5,movies,007-cassino-royale-t23,5,um dos melhores de 007,0,0,4,8 anos atrás,20-04-2019 06:20,0,0,0,,1,5
8 6,movies,007-cassino-royale-t23,6,"Nunca pensei que diria isso em um filme 007, mas esse é bastante ""meigo""",0,0,2.5,8 anos atrás,20-04-2019 06:20,0,0,0,,1,6
9 7,movies,007-cassino-royale-t23,7,preciso rever,0,0,5,8 anos atrás,20-04-2019 06:20,0,0,0,,1,7
10 8,movies,007-cassino-royale-t23,8,Adorei... Nossa esse Daniel Craig é UM HOMEM...,2,0,0,8 anos atrás,20-04-2019 06:20,0,0,0,,1,8
11 9,movies,007-cassino-royale-t23,9,"Divertido, mas achei no início um pouco entediante, com a Vesper que melhora. a parte do veneno na bebida é a mais emocionante do f
12 10,movies,007-cassino-royale-t23,10,"Absolutamente surpreendente! Temia que fosse detestar o roteiro do presunçoso Paul Haggis e que acharia a escolha de Daniel Craig
13 11,movies,007-cassino-royale-t23,11,"Nota 8,0. Ao contrario de muitos Daniel Craig está muito bem como James Bond. E este filme tem todos os elementos de um bom filme
14 12,movies,007-cassino-royale-t23,12,"Na moral,um dos melhores filmes que assisti na minha vida",3,0,0,8 anos atrás,20-04-2019 06:20,0,0,0,,1,12
15 13,movies,007-cassino-royale-t23,13,mta melhor 007 q o pierce ..#prontofalei,2,0,5,8 anos atrás,20-04-2019 06:20,0,0,0,,1,13
```

*Imagem 1: Amostra do início do corpus*

## 2.3 Tamanho e Cobertura

Para analisar o UTLCorpus, foi feito um código no Colab (Figuras 2 e 3), e foi necessário tratar cuidadosamente a leitura do arquivo CSV: muitas linhas apresentavam EOF ou vírgulas internas em campos de texto, o que quebrava a estrutura padrão. Utilizamos uma estratégia de “rsplit” para isolar as 11 últimas colunas (sem vírgulas internas) e um “split(..., 3)” para extrair corretamente os quatro primeiros campos (domain, object, author, text). Após filtrar e descartar linhas malformadas, obtivemos um DataFrame com as seguintes características:

- **Dimensões finais do DataFrame:**  
2 739 421 linhas × 15 colunas
- **Colunas do DataFrame:**  
['domain', 'object', 'author', 'text', 'likes', 'unlikes', 'stars', 'date', 'collect\_date', 'replies', 'favorite', 'want\_see', 'recommend', 'see', 'internal\_id']

A seguir, detalhamos as estatísticas principais desse subconjunto “limpo”:

### 2.3.1 Contagem por domínio

- **Filmes (“movies”):** 1.697.683 resenhas
- **Apps (“apps”):** 1.041.738 resenhas

O total de linhas no documento original é 2.8 milhões, portanto uma pequena porcentagem foi descartada por estar mal formatada.

### 2.3.2 Estatísticas de “stars” (nota decimal)

- **Média de “stars” em apps:** 3.9574
- **Média de “stars” em filmes:** 3.3125

Observa-se que, em média, as opiniões de aplicativos são mais positivas (quase 4 estrelas) do que as avaliações de filmes ( $\approx 3,3$  estrelas).

### 2.3.3 Transformação em classes inteiras (1 a 5)

Para facilitar o trabalho no futuro, convertemos o valor decimal stars (variando de 0.0 a 5.0) em cinco classes discretas, conforme o critério:

- **Classe 1:**  $0.0 \leq \text{stars} < 1.5$
- **Classe 2:**  $1.5 \leq \text{stars} < 2.5$
- **Classe 3:**  $2.5 \leq \text{stars} < 3.5$
- **Classe 4:**  $3.5 \leq \text{stars} < 4.5$
- **Classe 5:**  $4.5 \leq \text{stars} \leq 5.0$

**Após a conversão, obtivemos a distribuição geral (ambos domínios combinados):**

- **Classe 1:** 512.884 resenhas
- **Classe 2:** 104.297 resenhas
- **Classe 3:** 249.711 resenhas
- **Classe 4:** 566.666 resenhas
- **Classe 5:** 1.305.863 resenhas

Nota-se um claro viés para notas altas, especialmente na classe 5 (próximo de 48 % do total).

### 2.3.4 Médias das classes 1–5 por domínio

- **Média da classe em apps (1–5):** 3.9574
- **Média da classe em filmes (1–5):** 3.6191

Mesmo convertendo para inteiros, as resenhas de apps continuam ligeiramente mais positivas do que as de filmes.

Classe	Faixa de stars	Quantidade	%
1 ★	$0.0 \leq x < 1.5$	512.884	18,72%

2 ★	$1.5 \leq x < 2.5$	104.297	3,81%
3 ★	$2.5 \leq x < 3.5$	249.711	9,11%
4 ★	$3.5 \leq x < 4.5$	566.666	20,69%
5 ★	$4.5 \leq x \leq 5.0$	1 305 863	47,67%

***Tabela 1:** Distribuição de classe no dataset domínios movies e apps.*

Classe	Faixa de stars	Quantidade	%
1 ★	$0.0 \leq x < 1.5$	372.065	20,22%
2 ★	$1.5 \leq x < 2.5$	62.360	3,38%
3 ★	$2.5 \leq x < 3.5$	194.589	10,57%
4 ★	$3.5 \leq x < 4.5$	485.095	26,36%
5 ★	$4.5 \leq x \leq 5.0$	725.742	39,44%

***Tabela 2:** Distribuição de classe no domínio movies.*

## 2.4 Reformulação do Dataset

Para facilitar o treinamento dos modelos e reduzir o viés causado por desequilíbrio de classes, optamos por realizar um balanceamento do conjunto de dados. Identificamos que a **classe 2** era a mais representada, com aproximadamente **60.000 amostras** no domínio de filmes. Para mitigar esse desequilíbrio, selecionamos **10.000 amostras** da classe 2 e equilibramos o dataset escolhendo a **mesma quantidade (10.000)** para cada uma das demais classes: 1, 3, 4 e 5.

Essa reformulação teve dois principais benefícios:

- **Redução do viés** do modelo para a classe mais frequente;
- **Otimização do processamento**, tornando o treinamento mais leve e eficiente.

O dataset original continha aproximadamente **300.000 resenhas de filmes**, mas após a reformulação, trabalhamos com um subconjunto balanceado de **50.000 amostras no total** (10.000 por classe).

## 2.5 Utilização do Mesmo Dataset para Ambos os Modelos

Para garantir uma comparação justa e precisa entre os modelos, utilizamos **o mesmo conjunto de dados balanceado** para ambos. O dataset foi dividido em **80% para treinamento** (40.000 amostras) e **20% para teste** (10.000 amostras), assegurando que todos os modelos fossem avaliados sob as mesmas condições.

No caso do **modelo de análise de sentimentos baseado em léxico**, ele foi aplicado **apenas sobre o conjunto de teste**, uma vez que esse tipo de abordagem **não requer fase de treinamento supervisionado**. Isso permite avaliar seu desempenho diretamente sobre os dados reais, sem ajustes prévios com base no conjunto de treino.

## 3. Metodologia I: Análise de Sentimentos Baseada em Léxico

### 3.1 Objetivo

Esta etapa do projeto teve como objetivo avaliar o sentimento textual presente nas resenhas de filmes utilizando uma abordagem **léxica supervisionada por regras**, sem o uso direto de modelos de aprendizado de máquina. O foco foi verificar se um léxico emocional em português seria capaz de prever a nota atribuída por usuários (estrelas) com base apenas no conteúdo textual das resenhas.

### 3.2. Pré-processamento

Para o tratamento dos textos, utilizou-se a biblioteca nltk com o seguinte pipeline:

- **Normalização de e-mails:** substituição de padrões por <EMAIL>
- **Remoção de tags HTML**
- **Normalização de números:** substituição por <NUM>
- **Remoção de espaços extras**
- **Remoção de stopwords em português** - foi adaptado para não remover negadores e intensificadores utilizados no código.
- **Stemming com RSLPStemmer - não foi utilizado:** Ao aplicar o stemmer RSLP, palavras como “super” são reduzidas a “sup” e “bacana” a “bacan”, saindo do formato esperado pelo léxico. Com isso, termos intensificadores (“super”) e adjetivos comuns (“bacana”) deixam de ser reconhecidos corretamente.

Função final de pré-processamento:

```
def preprocessamentoDeTextoNLTK(texto: str) -> str:
    texto = normalizaEmails(texto)
    texto = removeTagsHTML(texto)
    texto = normalizaNumeros(texto)
    texto = removeEspacosExtras(texto)
    texto = removeStopwordsNLTK(texto)
    return texto
```

### 3.3. Léxico Utilizado

Foi utilizado o léxico **AffectPT-BR** (LaCAFÉ), um recurso lexical para o português brasileiro desenvolvido por [Silva et al., 2012], que categoriza palavras com base em sentimentos positivos e negativos.

- Fonte: AffectPT-BR v1.0 - LACAFE (Linguistic Annotation of Concepts Affectives in Brazilian Portuguese).
- Link: <https://github.com/LaCAfe/AffectPT-br/blob/master/AffectPT-br>
- Lexemas categorizados como:
  - '31' → **Posemo (positivo)**
  - '32' → **Negemo (negativo)**

O arquivo affectpt-br.txt foi processado para extrair os stems positivos e negativos. Tokens com \* foram truncados para indicar stems em vez de palavras inteiras.

### 3.4. Regra de Pontuação

Para cada texto, foi aplicada uma função de escore emocional com base em regras simples:

+1 → Para cada palavra positiva

-1 → Para cada palavra negativa

#### 3.4.1. Tratamento de negações

As palavras **negadoras** foram tratadas com inversão de polaridade para a palavra seguinte. O conjunto de negações consideradas foi:

{'não', 'nem', 'nunca', 'jamais'}

Assim, frases como:

"Não gostei" → Escore positivo (inversão de um termo negativo)

#### 3.4.2. Adição de intensificadores

Foram considerados intensificadores como:

{'muito', 'super', 'extremamente', 'bem', 'tão', 'totalmente', 'mega', 'bastante'}

Quando uma palavra positiva ou negativa era precedida por um intensificador, seu peso emocional era **dobrado** ( $\pm 2$ ). Exemplo:

"muito bom"  $\rightarrow$  score +2

"super ruim"  $\rightarrow$  score -2

### 3.5. Normalização da Pontuação

A pontuação bruta (lex\_score) foi normalizada para o intervalo de **1 a 5 estrelas**, simulando uma nota discreta. Utilizamos a abordagem de normalização por intervalo:

```
def map_score_faixa(s):  
    if s <= -1: return 1  
    elif s <= 0: return 2  
    elif s <= 1: return 3  
    elif s <= 2: return 4  
    else:      return 5
```

### 3.6. Exemplos de Resultados

Texto curto	Score Léxico	Estrela real	Estrela prevista
"A D O R OOOOO!"	+0	0.0	2
"Emocionei *ooooooooooooooooo*"	+0	0.0	2
"Super bacana, leve e com o carey neh cara!sempre bom!"	+2	1.0	4
"lindo filme :)"	+1	0.0	3

### 3.7. Limitações

- Palavras fora do léxico são ignoradas, mesmo que tenham forte carga emocional.
- Ambiguidades contextuais não são capturadas.
- Ironia e sarcasmo não são tratados.
- Intensificadores e negações só afetam palavras próximas (1 posição).

### 3.8. Avaliação Quantitativa - Modelo Baseado em Léxico



### Distribuição de classes

Classe (Estrelas)	Distribuição Real (discrete_stars)	Previsão Léxica (lex_stars_q)	Diferença
1 ★	1929	1591	-338
2 ★	1818	2807	+989
3 ★	1865	2395	+530
4 ★	1839	1233	-606
5 ★	1961	1386	-575

A tabela de distribuição de classes compara a quantidade real de avaliações em cada classe de estrelas com a quantidade prevista pelo modelo léxico. É notável que o modelo tende a centralizar suas previsões: ele subestima as classes extremas (1 e 5 estrelas) e superestima as classes intermediárias, principalmente a classe 2. Por exemplo, enquanto a classe 2 possui 1.818 exemplos reais, o modelo previu 2.807 - um excesso de quase 1.000 casos. Essa distorção indica que o modelo léxico frequentemente resulta em valores neutros (próximos de zero), que são então mapeados para notas médias (2 ou 3 estrelas), mesmo que a avaliação real do usuário seja mais extrema.

### Métricas por Classe (modelo léxico)

Classe	Precision	Recall	F1-score	Support
1 ★	0.201	0.165	0.181	1929
2 ★	0.218	0.336	0.264	1818
3 ★	0.191	0.246	0.215	1865
4 ★	0.230	0.154	0.184	1839
5 ★	0.250	0.176	0.207	1961

- **Acurácia geral:** 0.214
- **Média macro:**
  - Precision: 0.218
  - Recall: 0.215
  - F1-score: 0.210
- **Média ponderada:**
  - Precision: 0.218
  - Recall: 0.214

F1-score: 0.210

A tabela de Métricas por Classe apresenta importantes métricas de avaliação: precisão, revocação e F1-score - para cada uma das cinco classes. Os resultados reforçam o comportamento observado na distribuição: as classes centrais (2 e 3) possuem os melhores scores, especialmente a classe 2, que atinge o maior recall (0.336). Em contrapartida, classes como 1, 4 e 5 têm desempenho significativamente inferior, revelando que o modelo tem maior dificuldade em identificar sentimentos mais extremos. A acurácia geral do modelo é de apenas 21,4%, muito próxima do valor esperado para uma classificação aleatória (20% com cinco classes balanceadas).

### Matriz de Confusão (Modelo Léxico)

Real \ Previsto	1 ★	2 ★	3 ★	4 ★	5 ★
Real 1 ★	319	654	573	228	155
Real 2 ★	506	611	347	177	177
Real 3 ★	313	527	458	254	313
Real 4 ★	220	481	460	283	395
Real 5 ★	233	534	557	291	346

A matriz de confusão detalha como as previsões do modelo se distribuíram em relação às classes reais. As linhas representam as classes verdadeiras, enquanto as colunas indicam as classes previstas. Percebe-se que há uma forte concentração de previsões nas colunas 2 e 3, independentemente da classe real. Por exemplo, das 1.929 resenhas com nota real 1, apenas 319 foram corretamente classificadas como tal, enquanto 654 e 573 foram erroneamente rotuladas como classes 2 e 3, respectivamente. Isso confirma que o modelo, ao encontrar pouco sinal emocional nos textos (por ausência de termos no léxico ou presença de termos opostos), tende a prever classes mais neutras. A matriz evidencia claramente o viés do modelo em evitar os extremos.

Após analisar as três tabelas, confirma-se que as classes 2 e 3 estrelas receberam muito mais avaliações do que deveriam. Isso pode ser explicado pela estratégia usada: o modelo calcula uma pontuação bruta (*lex\_score*) baseada na soma de palavras positivas e negativas, mas muitas resenhas não utilizam exatamente as palavras do léxico. Como consequência, as pontuações acabam próximas de zero, e a maioria dos textos cai nas classes centrais (2 e 3 estrelas), mesmo quando deveriam estar nos extremos.

### 3.9. Avaliação Qualitativa - Modelo Baseado em Léxico

Texto (resumo)	Estrelas reais	lex_score	Estrelas previstas
----------------	----------------	-----------	--------------------

<b>“A D O R OOOOO!”</b>	1 ★	0	2 ★
<b>“Emocionei oooooooooooooooooooo”</b>	1 ★	0	2 ★
<b>“Filme de uma delicadeza incrível. Direção de arte maravilhosa.”</b>	1 ★	3	5 ★
<b>“lindo filme :)”</b>	1 ★	1	3 ★
<b>muito lenta a história</b>	2 ★	0	2 ★
<b>Desleixado e fora de contexto. Deveriam ter esperado um tempo maior e desenvolvido o projeto com mais calma e inteligência.</b>	2 ★	3	5 ★
<b>Tinha q ser mais dinâmico, daria até pra colocar o desfecho de cenas q ficaram pra subentender no final.Digo q serve</b>	3 ★	0	2 ★
<b>Achei muito parecido e considereei uma cópia de As Duas Vidas de Audrey Rose</b>	3 ★	0	2 ★
<b>Muito bom as cenas Pós créditos. O filme é bom.A cruzada de pernas me tirou risadas</b>	4 ★	3	5 ★
<b>Trilha sonora linda demais e o Gerard de Fantasma está lindo, mesmo deformado!</b>	4 ★	2	4 ★
<b>Sensacional ! ! ! !</b>	5 ★	0	2 ★
<b>Profundo e sincero. Stephen Daldry é, como sempre, impecável.</b>	5 ★	2	4 ★

<b>Punk , agoniante , bom ! Legolas ? fala né ! Neo ( matrix ) perde feio ! kkkkkk</b>	5★	-1	1★
<b>É uma necessidade ver esse filme em todas as páscoas! No mínimo! &lt;3</b>	5★	0	2★
<b>Agora vontade assistir original .</b>	4★	1	3★
<b>muito bom</b>	4★	2	4★
<b>Legal .</b>	3★	1	3★
<b>A trilha sonora é a única coisa que vale a pena .</b>	3★	-1	1★
<b>Livro chato , filme bom . Gostei .</b>	2★	1	3★
<b>Não vejo sentido nesses remakes de filmes bons. Seria muito melhor pegar um filme que não deu certo e melhorá-lo.</b>	2★	4	5★

A análise qualitativa, feita com base em exemplos de todas as classes reais (1 a 5), revelou tanto acertos quanto limitações do modelo léxico. Um dos padrões mais evidentes foi a tendência do modelo em prever notas altas quando encontra palavras positivas no texto — independentemente do contexto completo da resenha ou da nota real atribuída pelo usuário.

Em resenhas de nota 1, por exemplo, encontramos textos como “Filme muito bom e inteligente!” e “Vi a versão americana e gostei muito”, ambas classificadas pelo modelo como notas altas (4 ou 5), o que é esperado pela lógica do léxico. No entanto, o fato de essas avaliações terem nota real 1 sugere dois possíveis cenários: erro de anotação no dataset ou uso de ironia ou sarcasmo por parte do usuário. O modelo, por sua vez, não tem mecanismos para detectar ironia — ele apenas soma os termos positivos, o que gera essa discrepância.

Nas resenhas de nota 2, houve exemplos coerentes como “(?)”, onde o texto é vazio e o modelo previu nota 2, refletindo um `lex_score` neutro. No entanto, em textos como “Hoje é dia de assistir uma animação de Lego, e eu amei demais!!!”, a nota real foi 2, mas o modelo previu 5 devido à presença de intensificadores como “amei” e “demais”. Se essa nota baixa for intencional, trata-se de um uso sarcástico difícil de captar; se for erro do usuário, o modelo na verdade acertou o sentimento textual. Em outro exemplo como “Não vejo sentido

nesses remakes de filmes bons. Seria muito melhor pegar um filme que não deu certo e melhorá-lo.” o texto fala mal do filme, porém usa muitas palavras consideradas positivas pelo léxico, portanto tento uma nota 5.

Para textos com nota 3, observamos previsões tanto corretas quanto desviadas. Em “Tinha que ser mais dinâmico...”, o modelo previu nota 2 por falta de polaridade forte no texto. Já em “Filme com atuações excelentes”, a linguagem positiva elevou a previsão para nota 5, embora o usuário tenha dado apenas 3. Para o texto simples “bom” a previsão foi correta devido a simplicidade do texto — mais uma vez, reforçando o descompasso entre a percepção textual direta e a nota atribuída, talvez por fatores não textuais ou expectativa frustrada.

Nas resenhas de nota 4 e 5, os acertos foram mais frequentes, especialmente quando as palavras do léxico estavam bem representadas. Textos como “Trilha sonora linda demais” ou “Profundo e sincero. Impecável.” foram bem avaliados pelo modelo. Em contrapartida, frases mais subjetivas ou simbólicas, como “É uma necessidade ver esse filme em todas as páscoas! <3”, receberam notas baixas porque o modelo ignora emojis e interpretações mais emocionais ou religiosas do conteúdo.

No geral, o modelo se mostrou eficaz em capturar emoções expressas por palavras diretas, mas falha em lidar com ambiguidade, ironia, tom e possíveis erros humanos na anotação. Essa análise mostra que mesmo resenhas “positivas” no conteúdo podem ter nota real baixa por sarcasmo, ironia, erro de clique ou critérios subjetivos não textuais — desafios que um modelo léxico puro não tem como resolver.

### **3.10. Considerações Finais**

O modelo baseado em léxico demonstrou, ao longo das avaliações, tanto seu valor como baseline interpretável quanto suas limitações significativas diante da complexidade da linguagem natural. Para compreender melhor seu desempenho, analisamos separadamente os resultados quantitativos e qualitativos.

A avaliação quantitativa evidenciou uma acurácia geral de 21,4%, valor apenas marginalmente superior a uma aleatoriedade estimada em uma tarefa de classificação com cinco classes balanceadas. As métricas por classe revelaram um desempenho desproporcionalmente baixo nas classes extremas (1 e 5 estrelas), enquanto as classes centrais (2 e 3) concentraram a maioria das previsões do modelo. Isso se deve à maneira como o score emocional é calculado: em muitos textos, a ausência de palavras reconhecidas pelo léxico, ou a presença equilibrada de termos positivos e negativos, leva a um score próximo de zero, que é então mapeado para notas medianas. A matriz de confusão reforça essa tendência, mostrando uma alta taxa de previsões incorretas para as classes mais polarizadas, indicando que o modelo não consegue identificar sentimentos fortes de forma consistente.

Na análise qualitativa, observamos que o modelo acerta principalmente quando os textos são curtos, diretos e contêm palavras claramente positivas ou negativas presentes no léxico. Por outro lado, ele falha de maneira sistemática em casos com linguagem ambígua, uso de ironia, emojis, ou referências culturais implícitas. Textos subjetivos e mais elaborados, mesmo contendo emoções evidentes para um leitor humano, frequentemente são classificados de forma equivocada por conta da incapacidade do modelo de interpretar contexto e inferência semântica. Além disso, o modelo é particularmente vulnerável a erros em avaliações mal anotadas (como notas baixas com textos positivos), onde ele pode estar certo quanto ao sentimento, mas é penalizado na avaliação por limitações do próprio dataset.

Com base nas duas análises, concluímos que o modelo léxico é útil como ponto de partida por sua transparência, ausência de fase de treinamento e implementação direta. Ele permite uma rápida interpretação dos resultados e ajuda a entender como certas palavras impactam na classificação. No entanto, seus resultados deixam evidente que modelos baseados exclusivamente em regras e vocabulário fixo **não são suficientes** para capturar a riqueza e a subjetividade da linguagem natural, especialmente em avaliações textuais reais. Para alcançar melhores resultados, será necessário incorporar abordagens supervisionadas e modelos que considerem o contexto semântico das palavras. Ainda assim, a experiência com o modelo léxico foi fundamental para introduzir os conceitos de polaridade, scoring emocional e impacto de pré-processamento.

## 4. Metodologia II: Análise de Sentimentos Baseada em Aprendizado de Máquina

### 4.1 Objetivo

Esta etapa do projeto teve como objetivo avaliar o sentimento textual presente nas resenhas de filmes utilizando técnicas de aprendizado de máquina supervisionado. Diferentemente da abordagem léxica baseada em regras, aqui foram treinados modelos com base em exemplos rotulados do próprio dataset, buscando aprender padrões linguísticos associados às diferentes avaliações (estrelas). O foco foi investigar se algoritmos de classificação poderiam superar a abordagem léxica em precisão e desempenho, utilizando representações vetoriais do texto como entrada para os modelos.

### 4.2. Pré-processamento

O pré-processamento nesta etapa foi ligeiramente diferente da abordagem léxica, pois, ao utilizarmos a vetorização TF-IDF, foi necessário tratar diferentes elementos de forma a evitar ruído e melhorar a qualidade das representações numéricas dos textos. Foi montado o seguinte pipeline:

- **Normalização de texto:** padronização para letras minúsculas e substituição de padrões volumosos por placeholders (<NUM>, <EXCL>, <QST>) para reduzir a esparsidade e preservar sinais emocionais.
- **Remoção de ruídos estruturais:** eliminação de URLs, tags HTML e pontuação residual irrelevante.
- **Conversão de emojis:** transformação de emojis em shortcodes no formato <EMOJI\_nome> para preservar informações emocionais de forma controlada.
- **Remoção e ajuste de stopwords:** uso da lista padrão do spaCy, com remoção de palavras polarizadas (ex.: “não”) e adição de termos neutros frequentes no domínio (ex.: “filme”, “história”) para melhorar a discriminação do TF-IDF.
- **Tokenização e lematização:** aplicação do modelo **pt\_core\_news\_sm** do spaCy para reduzir palavras à sua forma base (lematização), descartando stopwords e reduzindo a esparsidade dos dados.
- **Vetorização TF-IDF:** geração de representações numéricas com filtragem de termos muito raros ou muito frequentes para reforçar o sinal relevante ao modelo.

### 4.3. Modelos de Aprendizado de Máquina

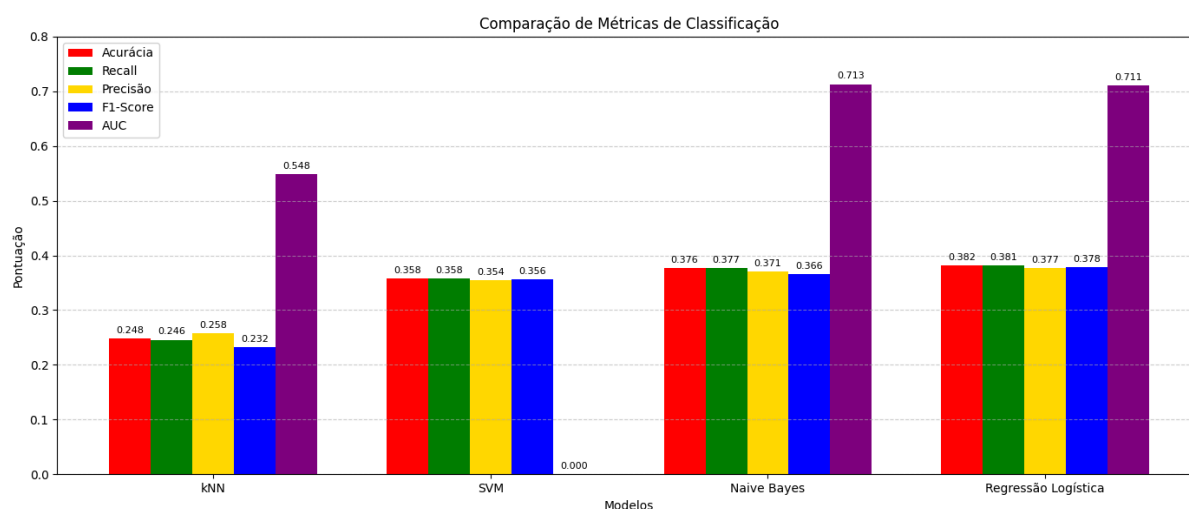
Foram avaliados quatro algoritmos clássicos de aprendizado supervisionado:

- **K-Nearest Neighbors (k-NN)**: classifica uma nova instância com base nas classes majoritárias de seus vizinhos mais próximos no espaço vetorial, sem etapa de treinamento explícita.
- **Naive Bayes**: modelo probabilístico baseado no teorema de Bayes, que assume independência entre as features e costuma apresentar bom desempenho em tarefas de classificação de texto.
- **Regressão Logística**: modelo linear que estima a probabilidade de cada classe por meio de uma função sigmoide, sendo eficiente e interpretável em espaços de alta dimensionalidade como o TF-IDF.
- **Support Vector Classifier (SVC)**: busca maximizar a margem entre classes no espaço de vetores, sendo eficaz para conjuntos de dados esparsos e com muitas dimensões.

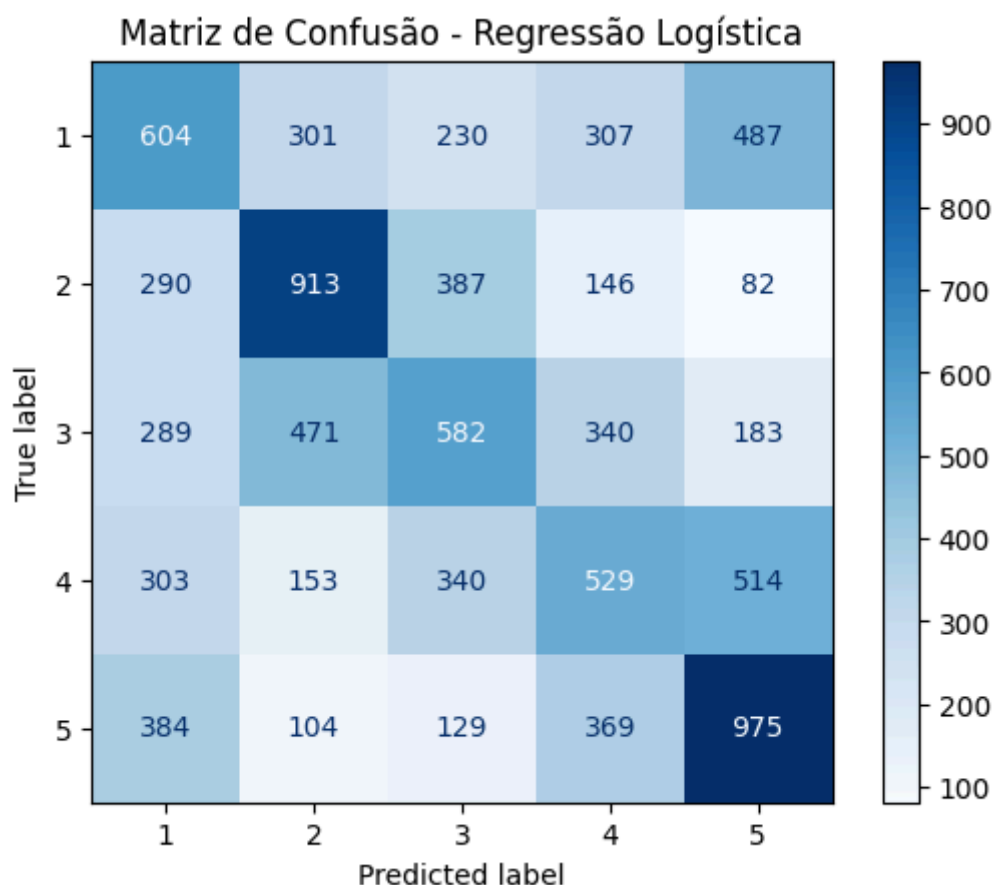
### 4.4. Resultados

#### 4.4.1 Avaliação Quantitativa

Os modelos foram avaliados utilizando o conjunto de treinamento e medimos seu desempenho sobre o conjunto de teste, que contém 10 mil amostras, com aproximadamente 2 mil exemplos por classe. A seguir, as métricas de avaliação (acurácia, precisão, recall, F1-score e AUC) obtidas por cada modelo e a matriz de confusão do modelo de maior desempenho:



*Imagem 2: Métricas de avaliação dos modelos de aprendizado de máquina (SVM não apresenta AUC devido à implementação utilizada, que não oferece suporte à saída de probabilidades nas classificações)*



*Imagem 3: Matriz de confusão da Regressão Logística, modelo que obteve o melhor desempenho*

#### 4.4.2 Avaliação Qualitativa

Para a avaliação qualitativa manual selecionamos as predições da Regressão Logística, modelo de melhor desempenho.

Texto (resumo)	Estrelas reais	Estrelas <u>Léxico</u>	Estrelas <u>ML</u>
“A D O R OOOOO!”	1★	2★	5★
“Emocionei ooooooooooooooooooooo”	1★	2★	5★
“Filme de uma delicadeza incrível. Direção de arte maravilhosa.”	1★	5★	5★
“lindo filme :)”	1★	3★	1★
muito lenta a história	2★	2★	2★
Desleixado e fora de contexto. Deveriam ter esperado um tempo maior e desenvolvido o	2★	5★	3★

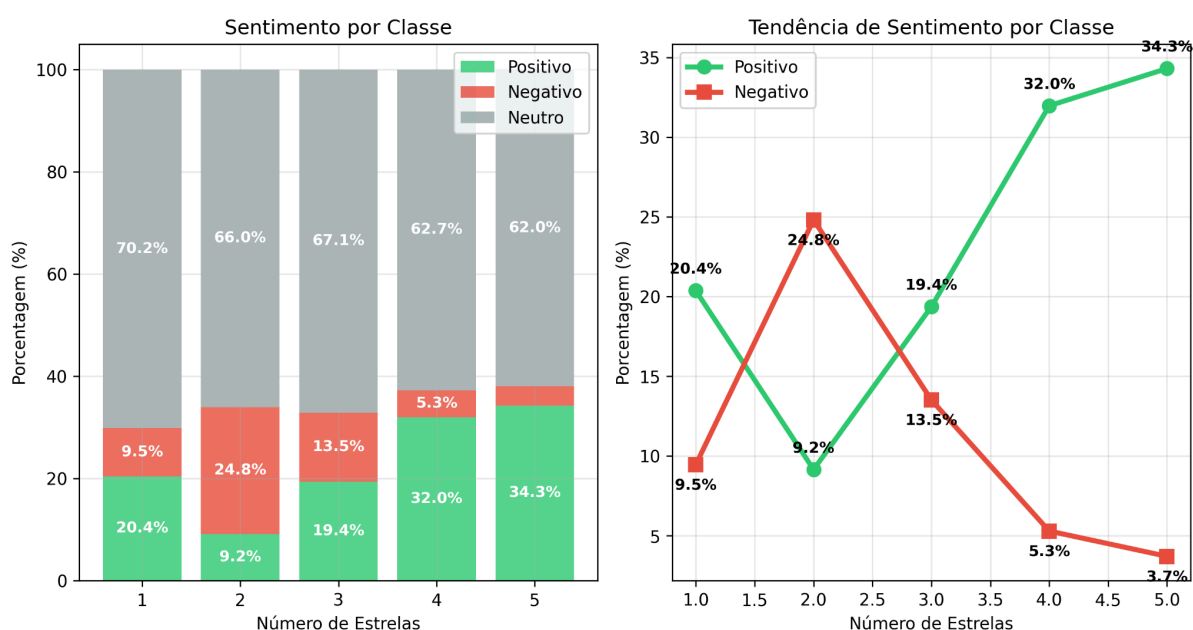


<b>projeto com mais calma e inteligência.</b>			
<b>Tinha q ser mais dinâmico, daria até pra colocar o desfecho de cenas q ficaram pra subentender no final.Digo q serve</b>	3★	2★	4★
<b>Achei muito parecido e considereei uma cópia de As Duas Vidas de Audrey Rose</b>	3★	2★	4★
<b>Muito bom as cenas Pós créditos. O filme é bom.A cruzada de pernas me tirou risadas</b>	4★	5★	4★
<b>Trilha sonora linda demais e o Gerard de Fantasma está lindo, mesmo deformado!</b>	4★	4★	4★
<b>Profundo e sincero. Stephen Daldry é, como sempre, impecável.</b>	5★	4★	4★
<b>É uma necessidade ver esse filme em todas as páscoas! No mínimo! &lt;3</b>	5★	2★	2★
<b>Livro chato , filme bom . Gostei .</b>	2★	3★	2★
<b>Sensacional !!!!!</b>	5★	2★	5★
<b>Punk , agoniante , bom ! Legolas ? fala né ! Neo ( matrix ) perde feio ! kkkkkk</b>	5★	1★	1★
<b>Agora vontade assistir original .</b>	4★	3★	4★
<b>Legal .</b>	3★	3★	3★

### 4.4.3 Discussão dos Resultados

A análise dos resultados indica um desempenho modesto, e a principal causa para esta performance pode ser atribuída à qualidade e à natureza do dataset utilizado. A acurácia de 38% em um problema de classificação de cinco classes (notas de 1 a 5), apesar de ser melhor que uma escolha aleatória (20%), mostra que o modelo enfrentou dificuldades para distinguir corretamente entre as categorias.

O fator mais impactante foi a presença de ruído e inconsistências nos dados, especialmente na classe de nota 1. Foi identificada uma quantidade expressiva de avaliações textuais com sentimento claramente positivo ("o usuário amou o filme") que, contraditoriamente, estavam associadas à nota mínima. Esse padrão de rotulagem incorreta forneceu "sinais" conflitantes ao modelo durante o treinamento. Consequentemente, o algoritmo de Regressão Logística aprendeu uma associação falsa entre vocabulário positivo e a nota 1, o que explica a confusão recorrente entre as classes 1 e 5 — as duas mais polarizadas. O modelo não conseguiu estabelecer uma fronteira de decisão clara, pois os exemplos de treinamento sugeriam que tanto palavras de exaltação quanto de repúdio poderiam levar a uma classificação de nota 1.



**Imagem 4:** Visualização de sentimento por cada classe, demonstrando como a classe 1 se comporta de maneira inesperada

Adicionalmente, o desempenho foi comprometido pela dificuldade inerente em diferenciar as notas intermediárias (2, 3 e 4). A distinção linguística entre uma avaliação de nota 2 e uma de nota 3, por exemplo, é frequentemente sutil e altamente subjetiva. Avaliações nessas faixas tendem a conter uma mistura de pontos positivos e negativos, tornando a tarefa de classificação mais complexa para um modelo como a Regressão Logística, que opera com base em relações mais lineares entre as features (palavras) e o resultado.

Apesar da baixa acurácia, a métrica AUC de 0.71 oferece uma perspectiva mais otimista. Um valor de AUC significativamente acima de 0.5 (que representaria um classificador aleatório) indica que o modelo possui uma capacidade discriminatória razoável. Ou seja, embora tenha dificuldade em acertar a classe exata, ele é melhor do que o acaso em identificar se uma avaliação é "mais positiva" do que outra. Isso sugere que o modelo

conseguiu capturar um sinal de sentimento geral, ainda que a presença massiva de dados mal rotulados o tenha impedido de traduzir esse entendimento em predições de classe precisas.

#### **4.4.4 Conclusão**

Este trabalho se propôs a explorar a aplicação de um modelo de Regressão Logística para a tarefa de classificação de notas de filmes com base em avaliações textuais. O modelo obteve uma acurácia de 38% e um AUC de 0.71, resultados que, embora não sejam ideais, forneceram insights valiosos sobre os desafios de se trabalhar com dados de usuários gerados em ambientes não controlados.

A principal conclusão deste estudo é que a qualidade do dataset é um fator muito importante para o desempenho de modelos de PLN. As inconsistências encontradas, principalmente avaliações positivas rotuladas com nota 1, prejudicaram fortemente o processo de treinamento e foram a causa primária da baixa performance do classificador, que demonstrou grande dificuldade em distinguir as classes de maior e menor nota. Além disso, a subjetividade e a semelhança linguística entre as avaliações de notas intermediárias (2, 3 e 4) representaram um desafio adicional.

Para trabalhos futuros, recomenda-se fortemente uma etapa de pré-processamento e limpeza de dados mais aprofundada. Uma abordagem poderia ser o uso de um modelo de análise de sentimentos pré-treinado para identificar e corrigir ou remover as amostras cujo rótulo numérico contradiz o sentimento do texto. Outra estratégia promissora seria a reformulação do problema: em vez de uma classificação de cinco classes, poderia se optar por uma classificação de três classes (negativo, neutro, positivo), agrupando as notas (1-2, 3, 4-5). Essa simplificação poderia mitigar os problemas de subjetividade nas notas intermediárias e gerar resultados mais robustos. Por fim, a experimentação com modelos mais avançados, como redes neurais recorrentes (LSTM) ou arquiteturas baseadas em Transformers (BERT), poderia capturar melhor as relações contextuais do texto e, potencialmente, superar as limitações encontradas.

## **Referências**

1. PANG, Bo; LEE, Lillian. Opinion Mining and Sentiment Analysis. Disponível em: <https://www.nowpublishers.com/article/Details/INR-011>. Acesso em: 29 mai. 2025.
2. SOCHER, Richard; et. al. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. Disponível em: <https://aclanthology.org/D13-1170/>. Acesso em: 29 mai. 2025.
3. DEVLIN, Jacob; et. al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Disponível em: <https://aclanthology.org/N19-1423/>. Acesso em: 29 mai. 2025.
4. SOUSA, R. F.; BRUM, H. B.; NUNES, M. G. V. A bunch of helpfulness and sentiment corpora in brazilian portuguese. In: SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. Proceedings of Symposium in Information and Human Language Technology - STIL. Salvador - BA, 2019.