

Toxic Comment Classification Challenge

Disciplina: Mineração de Dados (BCC444)

Alexander Josue Vásquez Alcántara

Matrícula: 25.1.0003

Julho de 2025

1. Descrição do Problema

Em várias plataformas de mídia social, as seções de comentários permitem que os usuários compartilhem ideias e discutam diferentes pontos de vista. No entanto, há um abuso de linguagem tóxica, especialmente no X (antigo Twitter), que não tem regras rígidas para regular esse tipo de comentário, o que está afetando a qualidade dessas conversas. Comentários que incluem ameaças, insultos, linguagem vulgar ou ódio contra certos grupos sociais, dificultando um diálogo saudável. O Desafio de Classificação de Comentários Tóxicos [1], organizado pela equipe de IA de Conversação do Jigsaw e pelo Google, propõe desenvolver um modelo de classificação com vários rótulos que identifica diferentes tipos de toxicidade em comentários extraídos de páginas de discussão da Wikipédia. Estudos anteriores como [2] mostram a viabilidade dessa abordagem. O modelo deve superar soluções atuais, como a Perspective API. E assim tornar as discussões online mais produtivas e respeitadas.

2. Motivação

Esse problema me motiva não apenas porque representa um desafio técnico interessante, mas também uma oportunidade de aplicar técnicas de mineração de dados, processamento de linguagem natural (PLN) e aprendizado de máquina com o objetivo maior de melhorar a qualidade das conversas online. Minha maior motivação é que esse problema desafiará meu conhecimento prévio em manipulação de dados, me dando outra abordagem para aplicá-lo com ferramentas tecnológicas modernas, especialmente no ecossistema Python (linguagem na qual busco me tornar especialista), que oferece grandes oportunidades para análise de texto, visualização de dados, modelagem e avaliação de modelos. Ferramentas como numpy, pandas, scikit-learn, TensorFlow, Excel, entre outras, serão essenciais nesse processo.

3. Objetivo do Trabalho

3.1. Objetivo General

Desenvolver um modelo de classificação multirrótulo para detectar diferentes tipos de toxicidade em comentários usando técnicas de mineração de dados e ferramentas de

análise de texto com Python, para que as pessoas possam se expressar online sem medo de receber comentários tóxicos.

3.2. Objetivos Especificos

1. Explore e analise o conjunto de dados obtido na página de discussão da Wikipédia.
2. Pré-processe comentários de diferentes arquivos .csv.
3. Treinamento de modelos de classificação multirrótulo para classificar comentários saudáveis e tóxicos.
4. Avaliar e comparar o desempenho dos modelos.
5. Interprete os resultados e apresente visualizações.
6. Analise quais tipos de comentários são mais difíceis de classificar.

4. Análise descritiva de cada atributo

4.1. Importação de biblioteca e carregamento de dados

Nesta etapa, foram importadas bibliotecas essenciais para manipulação, visualização e pré-processamento dos dados, como `pandas`, `numpy`, `matplotlib`, `seaborn` e `nlTK`. Em seguida, foram carregados os datasets `train.csv` e `test.csv`. Foi realizada uma verificação inicial das dimensões, tipos de variáveis e uma amostra das primeiras linhas para entender a estrutura geral.

Identificar os tipos de atributos

Os atributos do dataset foram classificados entre:

- **Atributos numéricos:** como o comprimento dos comentários e o índice de legibilidade de Flesch.
- **Atributos categóricos:** rótulos binários que indicam presença de toxicidade (*toxic*, *obscene*, *threat*, etc).
- **Atributo textual:** o conteúdo do comentário propriamente dito.

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate	length	flesch_score
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	41	66.370388
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	13	73.795735
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0	41	65.725000
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0	102	51.112030
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0	13	89.606731

Figura 1: Tabela Train

Para atributo numérico: análise descritiva e visualização

Foi calculada a estatística descritiva dos atributos numéricos, incluindo média, mediana, desvio padrão e quartis. Adicionalmente, foram gerados histogramas e boxplots para observar a distribuição dos valores e possíveis outliers, principalmente no comprimento dos comentários.

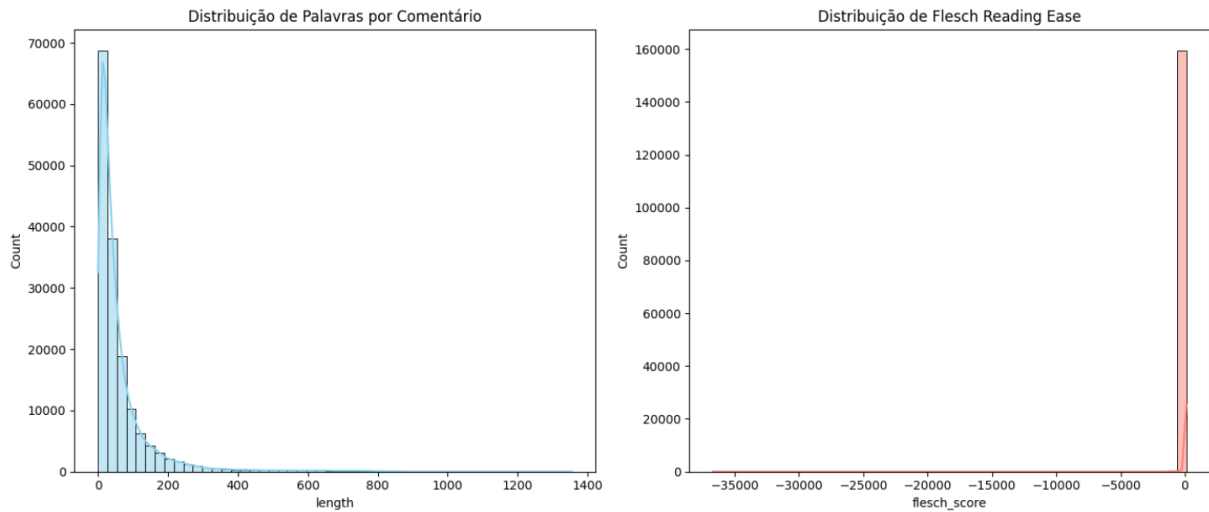


Figura 2: Gráfico de Distribuição

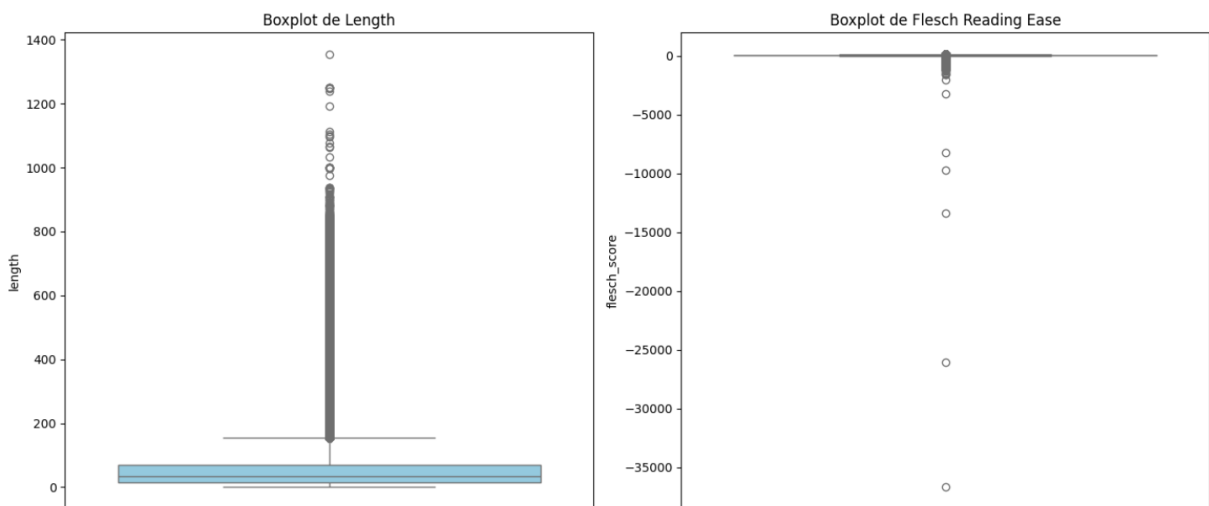


Figura 3: BoxPlot Length e Flesch Reading Ease

4.2. Distribuição de valores e popularidade dos atributos

Nesta parte, foram analisadas as distribuições das classes binárias que indicam tipos de toxicidade. Observou-se que a classe *toxic* é a mais frequente, enquanto outras como *threat* ou *identity hate* têm uma ocorrência muito menor. Também foram visualizados gráficos de barras mostrando a popularidade relativa de cada rótulo no dataset.

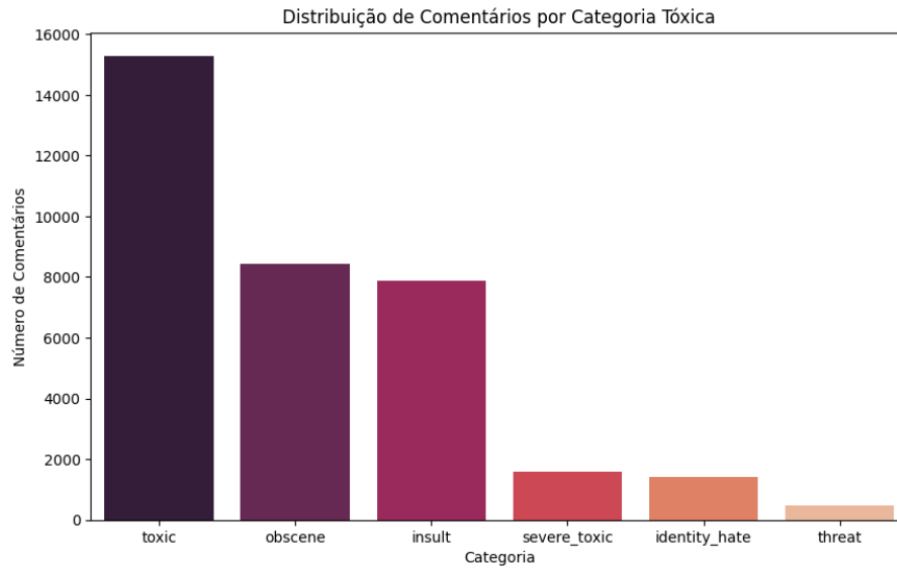


Figura 4: Comentário por Categoria

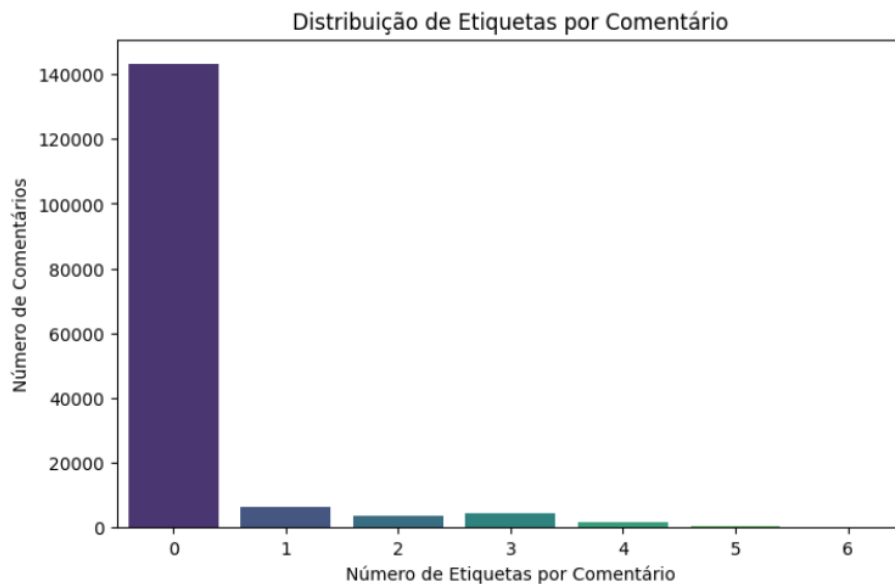


Figura 5: Etiquetas por Categoria

4.3. Valores Ausentes, Aberrantes e Inconsistências

Foi realizada uma inspeção detalhada para identificar valores ausentes no texto dos comentários e inconsistências, como linhas com apenas espaços ou registros vazios. Foram quantificadas as ocorrências de dados nulos e comentadas as possíveis implicações no pré-processamento.

4.4. Correlação entre os Atributos Numéricos

Nesta etapa, foi calculada a matriz de correlação entre as variáveis numéricas extraídas (comprimento e índice de Flesch). A visualização por meio de um heatmap permitiu verificar relações lineares e possíveis redundâncias entre os atributos.

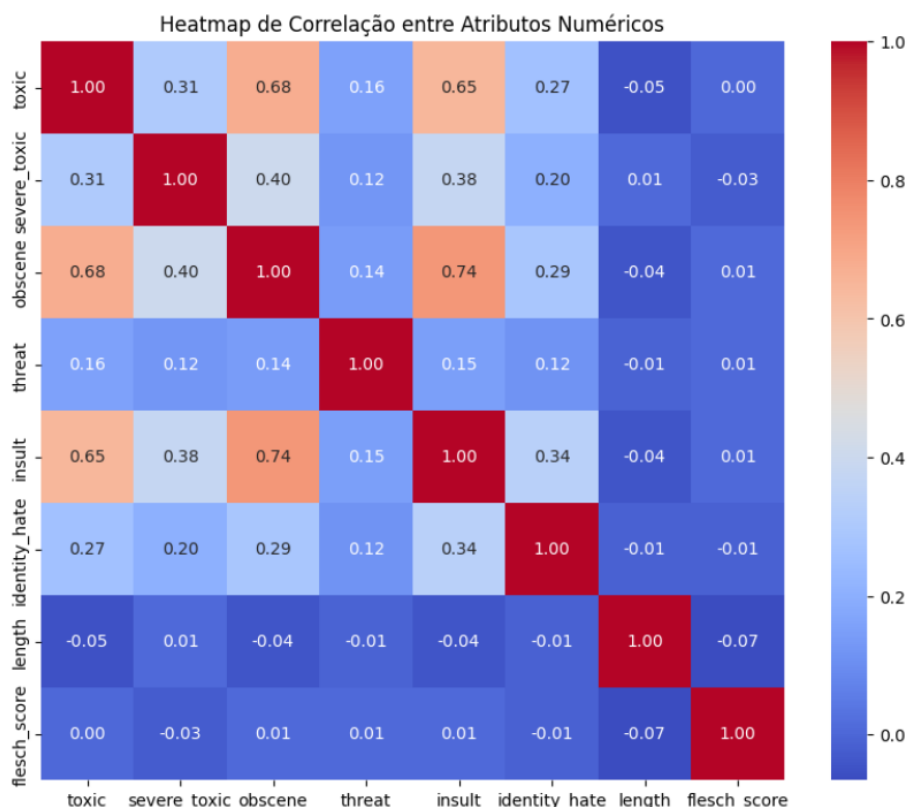


Figura 6: Heatmap de Correlação

5. Limpeza de dados e transformação

5.1. Preencha os valores ausentes

Para lidar com valores ausentes no atributo textual (`comment_text`), foram substituídos comentários nulos por strings vazias. Esta abordagem preservou a quantidade de registros originais no dataset, evitando a perda de exemplos durante o treinamento.

5.2. Suavize os ruídos

Nesta etapa, foram aplicadas diferentes técnicas de limpeza e normalização do texto:

- Conversão para minúsculas.
- Remoção de pontuação utilizando expressões regulares.
- Exclusão de stopwords com a biblioteca `nltk`.
- Filtragem de palavras excessivamente longas.
- Aplicação de stemming utilizando `SnowballStemmer`.

O resultado foi armazenado em uma nova coluna denominada `clean_comment`.

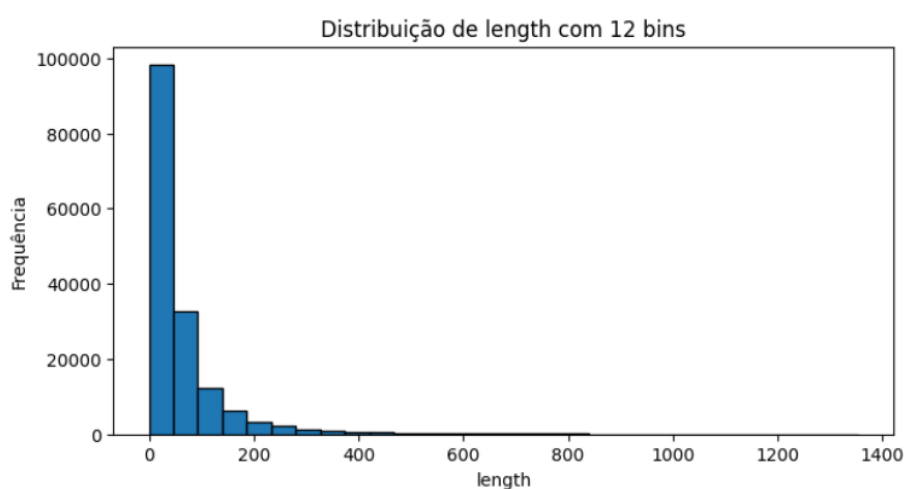


Figura 7: Distribuição de length com 12 bins

5.3. Resolva as inconsistências

Após a limpeza textual, foram identificados registros que ficaram vazios ou com menos de três caracteres. Para garantir a integridade do dataset, foi implementada uma função de verificação que removeu estes casos.

5.4. Transforme atributos categóricos nominais em vetor numérico

Os rótulos binários de toxicidade (*toxic*, *obscene*, *insult*, etc.) foram preservados como variáveis numéricas, prontas para treinamento supervisionado. Já os atributos derivados foram convertidos em representações numéricas adicionais.

```

Forma da matriz TF-IDF (treinamento): (159479, 10000)
Forma da matriz TF-IDF (teste): (152070, 10000)

```

Fila 0 (comentario):

	closur	doll	dont	edit	explan	fac	fan	gas	hardcor	metallica	...	retir	revert	sinc	talk	templat	usernam	vandal	vote	werent	york
0	0.290194	0.323237	0.105089	0.098388	0.189586	0.253809	0.205398	0.260158	0.288662	0.328837	...	0.243139	0.139814	0.13854	0.097207	0.17363	0.198416	0.137972	0.193188	0.235651	0.21976

1 rows x 24 columns

Fila 1 (comentario):

	background	colour	januari	match	stuck	talk	thank	utc
0	0.399472	0.463368	0.376183	0.387675	0.444444	0.176219	0.191516	0.262

Fila 2 (comentario):

	actual	care	constant	edit	format	guy	hey	info	inform	instead	man	page	realii	relev	remov	talk	tri	war
0	0.202501	0.229826	0.3118	0.27919	0.283085	0.243421	0.25349	0.257625	0.182985	0.235598	0.25169	0.127018	0.194168	0.249642	0.17358	0.27584	0.184785	0.228324

Figura 8: Vetor numérico

5.5. Transforme atributos categóricos ordinais em atributos numéricos seguindo a ordem entre eles

Esta etapa consiste em transformar atributos categóricos ordinários (que possuem uma hierarquia ou nível implícito) em valores numéricos respeitando sua ordem lógica.

O conjunto de dados do desafio contém apenas:

1. texto de comentário: texto livre (não ordinal)
 2. colunas como tóxico, obsceno, etc: binário (0 ou 1)
 3. lenght e flesh scaler: continuos (não ordinal)
- Portanto, não há atributos que exijam transformação ordinal.

5.6. Normalize os valores de atributos numéricos

Foram aplicadas duas técnicas de normalização:

- **Min-Max Scaling:** escalando os valores para o intervalo entre 0 e 1.
- **Padronização Z-Score:** transformando os atributos para média 0 e desvio padrão 1.

Lenght:

Tabela train

	length	length_norm	length_z
0	41	0.030281	-0.239259
1	13	0.009601	-0.534749
2	41	0.030281	-0.239259
3	102	0.075332	0.404490
4	13	0.009601	-0.534749

Tabela test

	length	length_norm	length_z
0	73	0.053914	0.098445
1	10	0.007386	-0.566409
2	5	0.003693	-0.619175
3	34	0.025111	-0.313131
4	6	0.004431	-0.608622

Figura 9: Normalização Lenght

Flesch:

Tabela train

	flesch_score	flesch_score_norm
0	66.370388	0.048078
1	73.795735	0.106214
2	65.725000	0.043025
3	51.112030	-0.071386
4	89.606731	0.230004

Tabela test

	flesch_score	flesch_score_norm
0	93.605000	0.261308
1	95.165000	0.273522
2	49.480000	-0.084163
3	73.757281	0.105913
4	30.530000	-0.232530

Figura 10: Normalização Flesch

Atributos normalizados

Após normalização, as variáveis numéricas ficaram prontas para serem utilizadas no treinamento dos modelos de classificação multirrótulo.

6. Regras de associação

6.1. Discretizar atributos numéricos

Nesta etapa, os atributos numéricos derivados (*comprimento do comentário* e *índice de legibilidade de Flesch*) foram discretizados em categorias. A discretização foi realizada com o objetivo de transformar variáveis contínuas em faixas discretas que pudessem ser interpretadas como itens transacionais. As faixas foram nomeadas de forma descritiva para facilitar a posterior interpretação das regras.

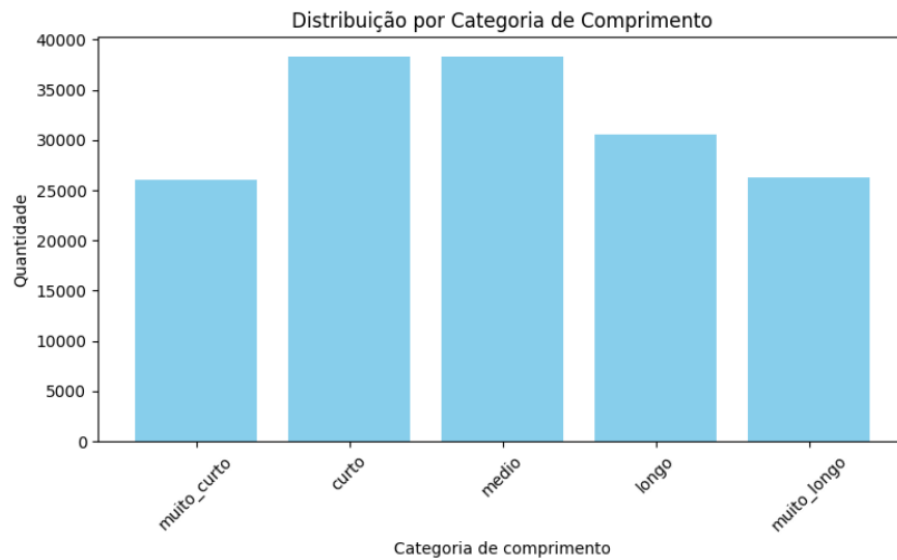


Figura 11: Distribuição por Categoria de Comprimento

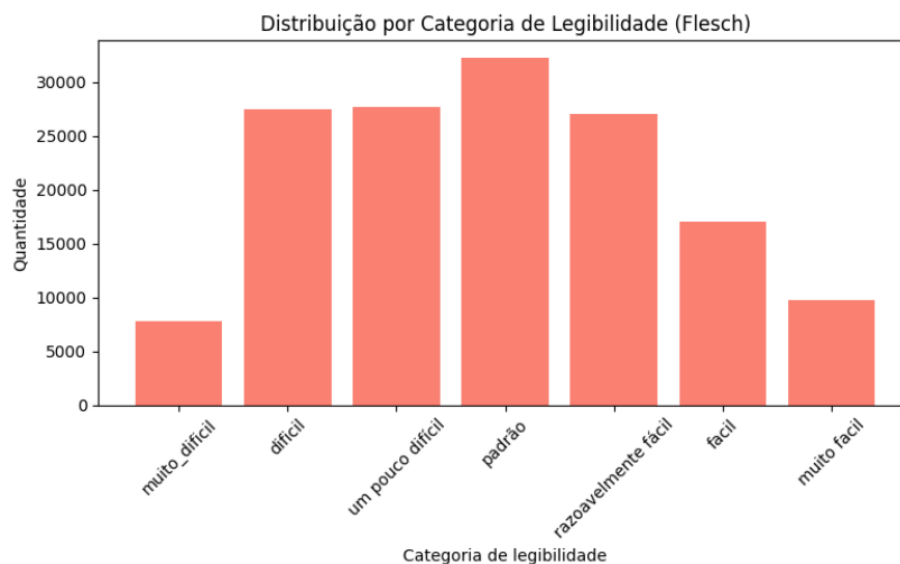
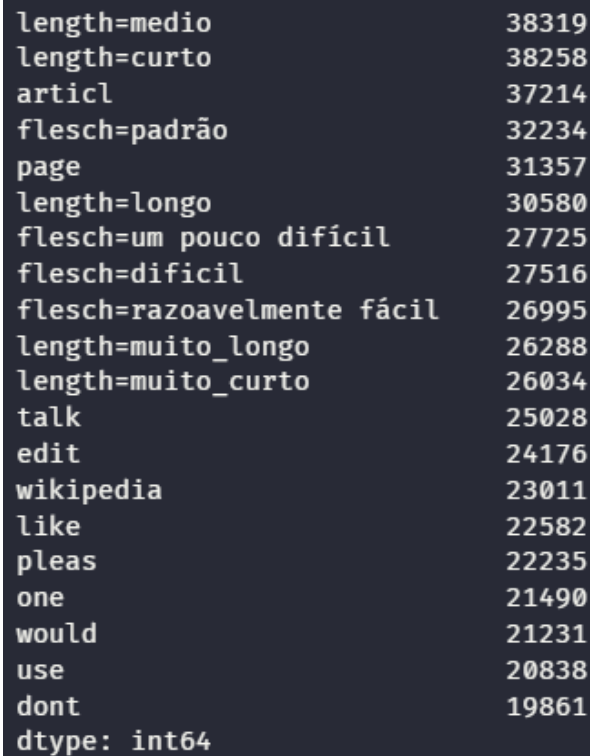


Figura 12: Distribuição por Categoria de Legibilidade

6.2. Transformação para formato transacional

Os dados foram convertidos em listas de itens por observação, contendo as categorias discretizadas e os rótulos binários ativos. Essa estrutura permitiu preparar o dataset no formato transacional necessário para aplicação dos algoritmos de mineração de regras. O `TransactionEncoder` da biblioteca `mlxtend` foi utilizado para transformar estas listas em uma matriz binária.



length=medio	38319
length=curto	38258
articl	37214
flesch=padrão	32234
page	31357
length=longo	30580
flesch=um pouco difícil	27725
flesch=difícil	27516
flesch=razoavelmente fácil	26995
length=muito_longo	26288
length=muito_curto	26034
talk	25028
edit	24176
wikipedia	23011
like	22582
pleas	22235
one	21490
would	21231
use	20838
dont	19861
dtype: int64	

Figura 13: Tabela de transações

6.3. Mineração de Regras de Associação com `mlxtend`

Foi utilizado o algoritmo Apriori da biblioteca `mlxtend` para extrair regras de associação. Os parâmetros principais incluíram:

- **Suporte mínimo:** definido com base na frequência relativa desejada.
- **Confiança mínima:** limite mínimo de confiança para as regras extraídas.

As métricas calculadas para cada regra foram suporte, confiança e lift, permitindo priorizar as associações mais relevantes.

1er Caso

No primeiro experimento, foram incluídas todas as variáveis discretizadas e rótulos binários. O objetivo foi identificar associações gerais entre características textuais e toxicidade. As principais regras extraídas evidenciaram padrões frequentes em comentários mais longos ou com baixa legibilidade, associados a rótulos como *toxic* e *obscene*.

	support	itemsets
39	0.240276	(length=medio)
37	0.239894	(length=curto)
3	0.233347	(articl)
26	0.202121	(flesch=padrão)
56	0.196621	(page)
38	0.191749	(length=longo)
28	0.173847	(flesch=um pouco difícil)
21	0.172537	(flesch=difícil)
27	0.169270	(flesch=razoavelmente fácil)
41	0.164837	(length=muito_longo)

Figura 14: Itens Sets Frequentes

	antecedents	consequents	support	confidence	lift
3	(classe=toxic)	(classe=obscene)	0.049699	0.518310	9.783359
4	(classe=obscene)	(classe=toxic)	0.049699	0.938099	9.783359
2	(classe=insult)	(classe=toxic)	0.046044	0.932326	9.723151
0	(also)	(length=muito_longo)	0.052910	0.517351	3.138568
5	(time)	(length=muito_longo)	0.045931	0.501060	3.039736
6	(talk)	(page)	0.085391	0.544111	2.767300
1	(length=muito_longo)	(articl)	0.088219	0.535187	2.293522

Figura 15: Regras de associação ordenadas por lift

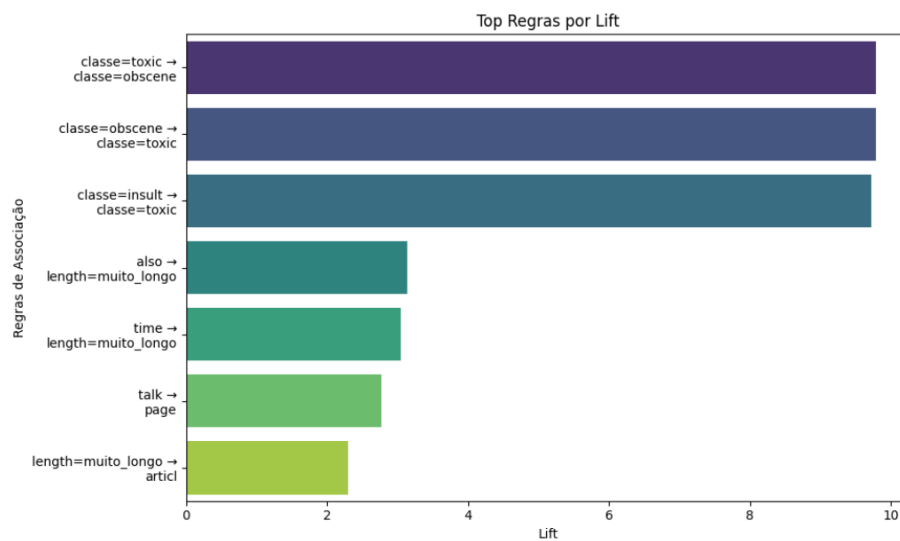


Figura 16: Gráfico Regras de Associacao

2do Caso

No segundo experimento, o foco foi em um subconjunto de atributos para verificar como determinadas discretizações isoladas se relacionavam com a presença de toxicidade. Este cenário permitiu filtrar regras mais específicas com lift elevado, destacando combinações de características menos comuns.

	support	itemsets
62	0.240276	(length=medio)
60	0.239894	(length=curto)
5	0.233347	(articl)
42	0.202121	(flesch=padrão)
83	0.196621	(page)
61	0.191749	(length=longo)
44	0.173847	(flesch=um pouco difícil)
37	0.172537	(flesch=difícil)
43	0.169270	(flesch=razoavelmente fácil)
64	0.164837	(length=muito_longo)

Figura 17: Itens Sets Frequentes 2

	antecedents	consequents	support	confidence	lift
92	(classe=insult, classe=toxic)	(classe=obscene)	0.037334	0.810840	15.305006
97	(classe=obscene)	(classe=insult, classe=toxic)	0.037334	0.704699	15.305006
94	(classe=toxic, classe=obscene)	(classe=insult)	0.037334	0.751199	15.210818
95	(classe=insult)	(classe=toxic, classe=obscene)	0.037334	0.755967	15.210818
22	(classe=insult)	(classe=obscene)	0.038594	0.781488	14.750969
23	(classe=obscene)	(classe=insult)	0.038594	0.728489	14.750969
96	(classe=toxic)	(classe=insult, classe=obscene)	0.037334	0.389354	10.088346
93	(classe=insult, classe=obscene)	(classe=toxic)	0.037334	0.967344	10.088346
26	(classe=toxic)	(classe=obscene)	0.049699	0.518310	9.783359
27	(classe=obscene)	(classe=toxic)	0.049699	0.938099	9.783359

Figura 18: Regras de associacao 2

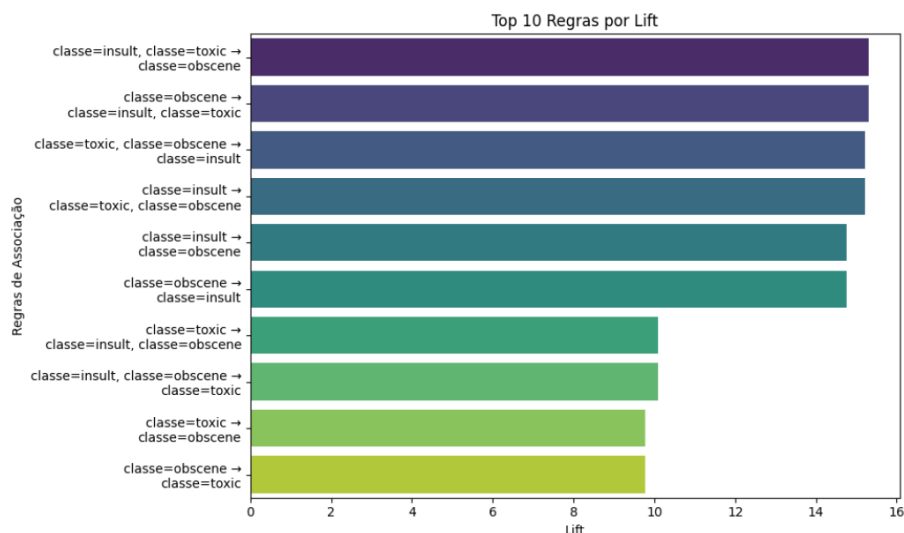


Figura 19: Grafico regras de associacao 2

6.4. Filtrar regras com consequente relacionado ao atributo alvo

Finalmente, foi realizada uma filtragem das regras extraídas para manter apenas aquelas cujo consequente estava diretamente relacionado aos rótulos de toxicidade. Esta etapa facilitou a interpretação e permitiu a geração de insights sobre como características dos comentários estão associadas ao comportamento tóxico.

Regras cujo consequente é 'Toxic':

	antecedents	consequents	support	confidence	lift
97	(classe=obscene)	(classe=insult, classe=toxic)	0.037334	0.704699	15.305006
95	(classe=insult)	(classe=toxic, classe=obscene)	0.037334	0.755967	15.210818
93	(classe=insult, classe=obscene)	(classe=toxic)	0.037334	0.967344	10.088346
27	(classe=obscene)	(classe=toxic)	0.049699	0.938099	9.783359
24	(classe=insult)	(classe=toxic)	0.046044	0.932326	9.723151

- Regra:
Se [classe=obscene] então [classe=insult, classe=toxic]
Suporte: 0.037
Confiança: 0.705
Lift: 15.305

- Regra:
Se [classe=insult] então [classe=obscene, classe=toxic]
Suporte: 0.037
Confiança: 0.756
Lift: 15.211

- Regra:
Se [classe=insult, classe=obscene] então [classe=toxic]
Suporte: 0.037
Confiança: 0.967
Lift: 10.088

- Regra:
Se [classe=obscene] então [classe=toxic]
Suporte: 0.050
Confiança: 0.938
Lift: 9.783

Figura 20: Atributo Alvo

7. Mineração de padrões sequenciais

7.1. Preparar representação sequencial

Para a mineração de padrões sequenciais, foi necessário criar uma representação ordenada das interações de cada comentário. Os registros foram agrupados por identificador (ou outra chave temporal) e ordenados conforme critérios de tempo ou sequência lógica. Cada sequência resultante incluía atributos discretizados (como comprimento e legibilidade) e indicadores binários ativos de toxicidade. Esse pré-processamento permitiu transformar o dataset em uma coleção de sequências temporais adequadas ao algoritmo de mineração.

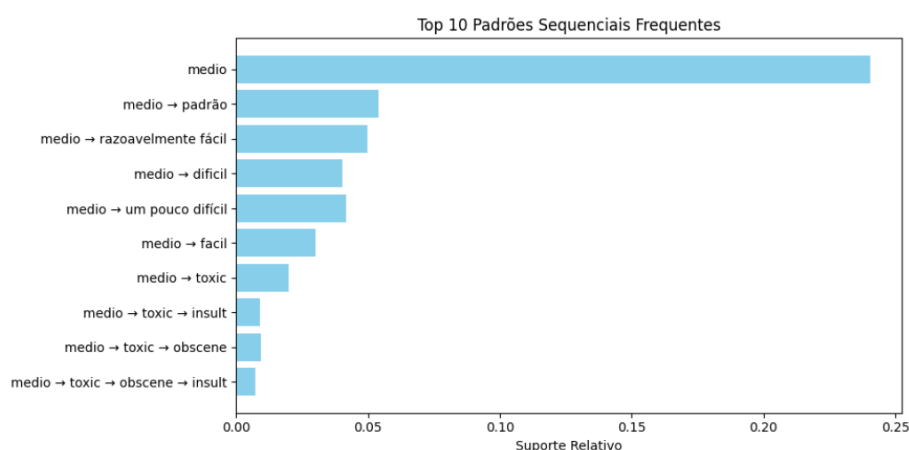


Figura 21: Sequencias

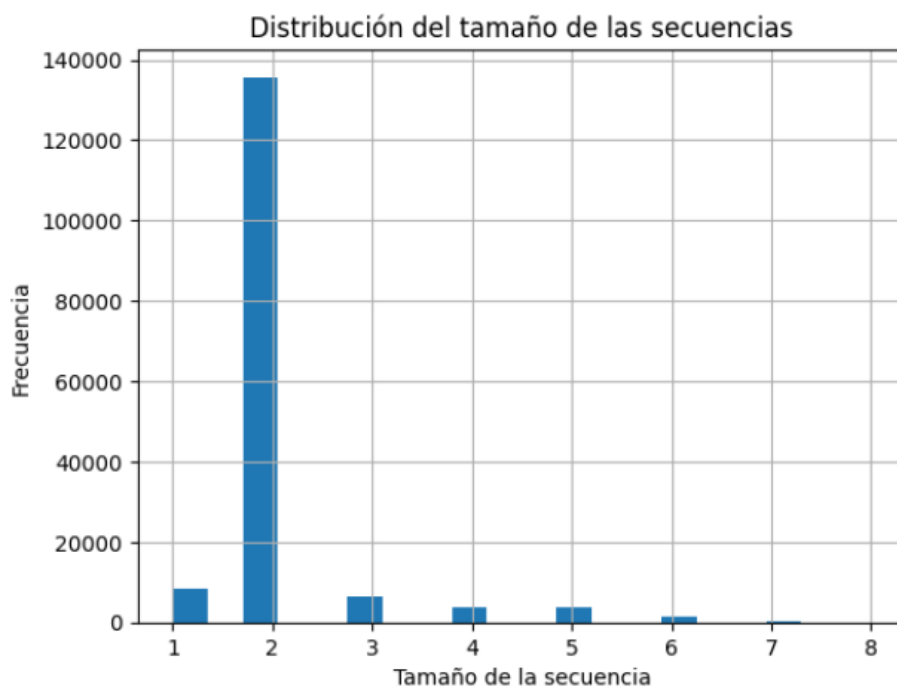


Figura 22: Grafico secuencias

7.2. Mineração de padrões com PrefixSpan

Foi utilizado o algoritmo PrefixSpan, implementado por meio da biblioteca `prefixspan`, para extrair padrões sequenciais frequentes. O parâmetro principal foi o **suporte mínimo**, definido como a proporção mínima de sequências nas quais cada padrão deveria ocorrer. O algoritmo retornou listas de itens ordenados que surgiam com frequência acima do limiar especificado.

Os principais padrões identificados incluíram combinações de discretizações de comprimento e legibilidade associadas à presença de classes de toxicidade. A interpretação desses padrões forneceu insights sobre o comportamento recorrente em comentários, permitindo observar sequências típicas de características textuais que precedem manifestações de toxicidade.

	sequencia	tamanho	suporte_absoluto	suporte_relativo
86	toxic, obscene	2	7926	0.049699
90	toxic, insult	2	7343	0.046044
94	obscene, insult	2	6155	0.038594
87	toxic, obscene, insult	3	5954	0.037334
32	curto, toxic	2	4587	0.028762
68	muito_curto, toxic	2	4134	0.025922
6	medio, toxic	2	3188	0.019990
42	razoavelmente fácil, toxic	2	3053	0.019144
115	facil, toxic	2	2732	0.017131
72	muito_curto, obscene	2	2617	0.016410
69	muito_curto, toxic, obscene	3	2555	0.016021
36	curto, obscene	2	2489	0.015607
38	curto, insult	2	2402	0.015062
33	curto, toxic, obscene	3	2387	0.014967
16	padrão, toxic	2	2344	0.014698
74	muito_curto, insult	2	2338	0.014660
35	curto, toxic, insult	3	2280	0.014297
71	muito_curto, toxic, insult	3	2255	0.014140
59	muito facil, toxic	2	2096	0.013143
101	longo, toxic	2	1944	0.012190

Figura 23: Tabela Padrões Sequenciais Frequentes

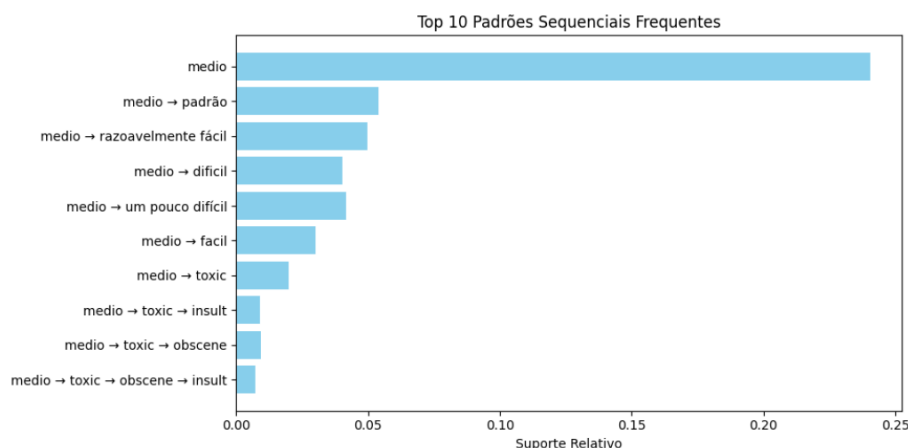


Figura 24: Grafico Padrões Sequenciais Frequentes

8. Referências Bibliográficas

- [1] Kaggle Notebooks. (2018). *Jigsaw Toxic Comment Classification Notebooks*. Disponível em: <https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/code>
- [2] Ramesh, V. (2019). *Toxic Comment Classification*. Disponível em: https://www.researchgate.net/publication/334123568_Toxic_Comment_Classification
- [3] Anandarajan, M., Hill, C., Nolan, T. (2019). *Text Preprocessing*. In: Practical Text Analytics. Advances in Analytics and Data Science, vol 2. Springer, Cham. Disponível em: https://doi.org/10.1007/978-3-319-95663-3_4
- [4] Tan, P.-N., Steinbach, M., Kumar, V. (2006). *Association Analysis: Basic Concepts and Algorithms*. In: Introduction to Data Mining. Pearson Addison Wesley. Disponível em: <http://www-users.cs.umn.edu/~kumar/dmbook/index.php>
- [5] Agrawal, R., Srikant, R. (1995). *Mining Sequential Patterns*. In: Proceedings of the 11th International Conference on Data Engineering (ICDE), IEEE. Disponível em: <https://ieeexplore.ieee.org/document/380415>