**Aprendizado de Máquina 2**

**Trabalho 1**

Classificação Multirrótulo

**Professor:** Prof. Dr. Murilo Coelho Naldi

Gabriel Lourenço de Paula Graton - 800432

Vitor Matheus da Silva - 800260

Rafael Naoki Arakaki Uyeta - 800207

São Carlos

8 de dezembro de 2023

**Índice**

[**1. Introdução 3**](#_heading=h.gjdgxs)

[**2. Escolha do Conjunto de Dados 3**](#_heading=h.30j0zll)

[**3. Estudo do Conjunto de Dados 4**](#_heading=h.1fob9te)

[3.1. Descrição do conjunto de dados 4](#_heading=h.3znysh7)

[3.2. Atributos 4](#_heading=h.2et92p0)

[3.3. Rótulos 10](#_heading=h.tyjcwt)

[**4. Tratamento de Dados 12**](#_heading=h.nwd5lnn4vccz)

[**5. PLN. 13**](#_heading=h.d95jmshk2hma)

[**6. Classificação Multirrótulo 14**](#_heading=h.3dy6vkm)

[**7. Conclusão 15**](#_heading=h.1t3h5sf)

[**8. Reflexões Finais 15**](#_heading=h.haacqdsiy43n)

[**9. Bibliografia 21**](#_heading=h.4d34og8)

## **1. Introdução**

Esse trabalho tem como objetivo exercitar os conceitos aprendidos na disciplina de Aprendizado de Máquina 2 referente aos tópicos de Classificação Multirrótulo. Para isso, será selecionado um conjunto de dados, sobre o qual serão aplicadas técnicas de ajuste para o problema específico (como conversão de atributos e normalização), além de técnicas de visualização (como histogramas e wordclouds) para, por fim, aplicar diversos algoritmos de classificação e comparar seus resultados.

Assim, mostrou-se possível exercitar os conceitos aprendidos em sala, de forma que o grupo obteve reflexões e ponderações sobre o funcionamento dos algoritmos e do processo de preparação dos dados.

## **2. Escolha do Conjunto de Dados**

O conjunto de dados escolhido para a pesquisa é denominado *Best Books* (<https://www.kaggle.com/datasets/ishikajohari/best-books-10k-multi-genre-data/data>), e contém informações sobre os 10 mil livros mais recomendados de todos os tempos.

A escolha deste conjunto se deve a presença de mais de 600 gêneros que podem ser explorados no processo de classificação multirrótulo, além da necessidade de tratamento dos atributos na medida em que os valores apresentados precisam ser convertidos de números arbitrários para uma escala linear e então normalizados.

O desafio proposto por esse conjunto de dados foi tentar identificar livros que uma pessoa em específico iria gostar, se baseando nos livros que anteriormente teria lido. Isto seria possível ao saber os gêneros que uma pessoa mais gosta, e a partir disso fazer uma seleção do conjunto de dados para demonstrar aqueles que mais se assemelham ao desejado.

O conjunto de dados contém 617 gêneros diferentes, (). Todas as manipulações dos dados e aplicações podem ser encontradas no notebook do Collab, que pode ser acessado neste link: [Link da realização do trabalho](https://colab.research.google.com/drive/1Mq1x1XSeP__lWOFhHP7iAt7HTSe8oXwm#scrollTo=wy0Y1dpNbDd7).

## **3. Estudo do Conjunto de Dados**

### **3.1. Descrição do conjunto de dados**

Os dados foram coletados na tentativa de identificar livros que uma pessoa pode gostar com base naqueles que ele já tenha lido. Para isso, neste dataset tem 10000 livros dos mais recomendados de todos.Sendo estes dados coletados de “Goodreads “ da lista -” *Books That Everyone Should Read At Least Once”.*

Esse dataset foi escolhido por alguns motivos, dentre eles foi a questão de explorar mais como funciona o Processamento de Linguagem Natural (PLN) dada a curiosidade dos integrantes do grupo; Interesse em Leitura, o modelo pode servir para ajudar a obter ótimas recomendações; Por fim, aprimorar e evidenciar os conhecimentos divulgados em sala de aula.

Ademais, trata-se um classificador de múltiplas classes, contendo mais de 600 gêneros. Ou seja, já é desafiador o problema,mas ao refletir sobre, percebe-se que são livros, classificar gêneros é árduo, pois é algo tão pessoal, que gera muitos conflitos.

### **3.2. Atributos**

Abaixo, estão descritos os atributos utilizados para a classificação dos livros.

* **#Indíce**
* **Livro**

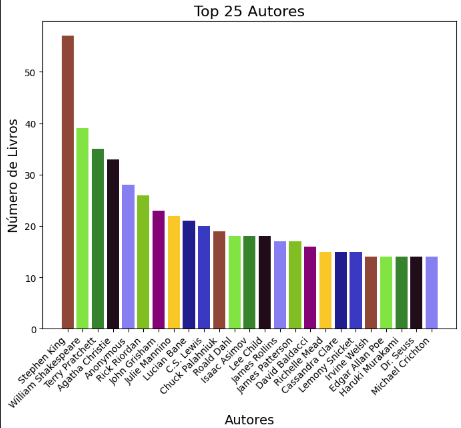
Nome do livro, algumas vezes pode aparecer detalhes da série de livros que pertence entre parênteses.



***Imagem 0 -*** *Alguns Livros*

* **Autor**

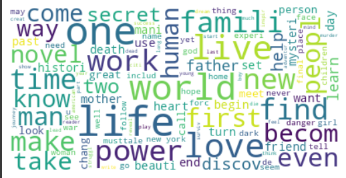
Nomes dos autores



***Imagem 1 -*** *Top25 Autores mais frequentes*

* **Descrição**

Descrição do livro segundo o Goodreads, site de catalogação de livros.



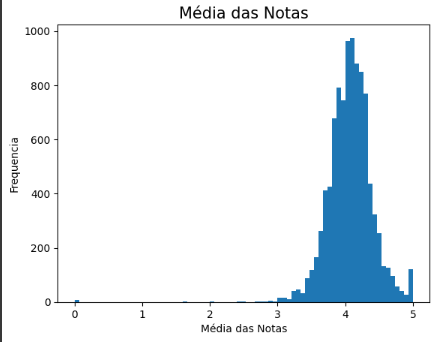
***Imagem 2 -*** *Top25 palavras mais frequentes*

******

***Imagem 3 -*** *Top1000 palavras mais frequentes*

* **Avaliação Média**

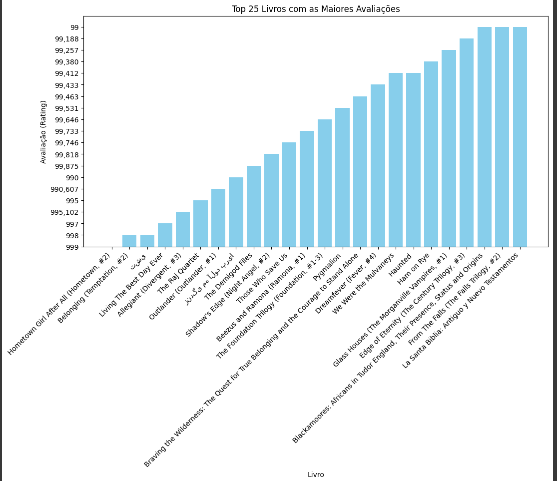
A avaliação média de cada livro é de 0 a 5.



***Imagem 4 -*** *Média das Notas*

* **Número de Avaliações**

Quantos usuários avaliaram o livro, no sentido de colocar uma nota de 0 a 5, não se considera se escreveram uma avaliação em texto sobre o livro.

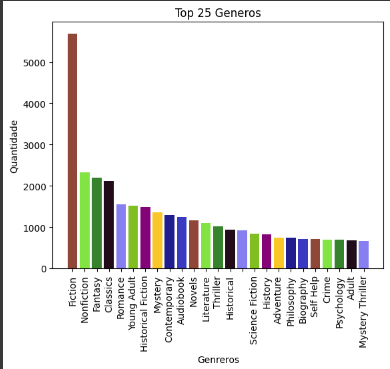


***Imagem 5 -*** *Top25 maiores número de avaliações*

* **URL**

Link de acesso aos livros no Goodreads.

### **3.3. Rótulos**



***Imagem 6 -*** *Top25 Gêneros mais frequentes*

A Coluna Gênero possui dentro de si uma lista que contém os gêneros os quais o livro pertence.Contudo, ao rodar o código, tem que tomar cuidado com os tipos, por isso a primeira abordagem e transformar essas “Strings” em realmente “Listas.



***Imagem 7 -*** *Mudar coluna de Str para List*

Depois disso, era entender quantos valores são esses, para isso foi preciso dividir todos os Gêneros, o que no final deu, ao todo, 617 Gêneros:



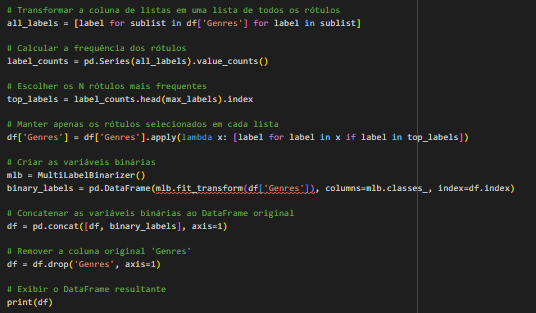
***Imagem 8 -*** *Alguns dos 617 Gêneros*

## **4. Tratamento de Dados**

Após essa série de visualizações, mais o cuidado com as Listas presentes em Gênero, percebeu-se que havia algumas descrições vazias. Sem nenhuma informação sobre o livro, não é possível fazer a classificação destes mesmos, por isso foi melhor retira-los.

Mas também, para trabalhar com Multi Label, seria necessário escolher alguma estratégia,a qual foi dada ênfase a **MultiLabelBinarizer()**que pega os rótulos e cria uma classe para cada um. Contudo, isso tem seus pontos negativos, visto que 617 colunas seriam adicionadas, mas também todos os atributos precisariam ser independentes - já que é uma abstração aceita ao utilizar esse método.

Por isso, dado a frequência com que os gêneros apareciam e por uma curiosidade do método, o número de labels foi reduzido para os 25 que têm mais quantidade.

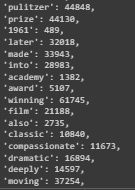
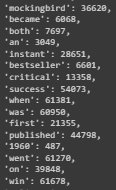
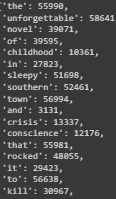


***Imagem 9 -*** *Tratamento de Dado aplicado*

## **5. PLN.**

Após isso, seria preciso explorar a descrição. Contudo, são muitas palavras e muitos termos para serem convertidos em valores numéricos e trabalhados.

Por essa razão, será abordado a biblioteca: **TfidfVectorizer(analyzer='word', max\_features = max\_feat) #Word é o padrão**

****

***Imagem 10 -*** *Frequência de Palavras*

Essa estratégia ela trabalha com a Tokenização. O TfidfVectorizer primeiro divide cada documento em palavras ou termos individuais Em seguida, ele calcula a frequência de cada termo em cada documento, chamada de Term Frequency (TF). A fórmula geral para TF é o número de vezes que um termo aparece em um documento dividido pelo número total de termos no documento.O resultado é uma matriz em que cada linha representa um documento, cada coluna representa um termo e cada valor na matriz é o peso TF-IDF desse termo no documento correspondente.

Ao passar com esse parâmetros, será utilizado as palavras (‘words’) e também um número de features, que será 10000 - para aqui foram feitos alguns testes. Escolher a menor quantidade não há como gerar modelos, mas gerar valores com um número grande de Tokens pode ocasionar em “Lixos” e confundir os modelos.

Com isso pronto, será preciso dividir os modelos de treino e teste, utilizando a função:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, binary\_labels, test\_size = 0.2, random\_state = 22)

Definindo o tamanho da proporção e como seria feita a escolha aleatória.

## **6. Classificação Multirrótulo**

Os resultados da classificação multirrótulo com cada método estão resumidos na tabela abaixo. Para avaliar o desempenho dos classificadores, utilizamos acurácia e perda de Hamming. A acurácia é uma medida que considera apenas acertos inteiros o que, em nosso caso, corresponde a situações nas quais as 7 classes foram previstas com sucesso. Já a perda de Hamming admite que podem haver acertos fracionados, o que a torna uma medida mais adequada para avaliação.

| **Modelo** | **Perda de Hamming** | **Acurácia** |
| --- | --- | --- |
| SGDClassifier OneVsRest | 0.08252 | 0.16826 |
| LogisticRegression OneVsRest | 0.10011 | 0.09421 |
| LinearSVC OneVsRest | 0.08177 | 0.17884 |
| Classifier chain com random forest | 0.11029 | 0.09068 |
| Label powerset com random forest | 0.12203 | 0.16221 |

Verificamos que o algoritmo com melhor desempenho foi o LinearSVC, com 0,08177 de perda de Hamming e 0.17884 de acurácia, ao passo que os piores algoritmos foram o Classifier chain(0,11029 de perda de Hamming e 0,09068 de acurácia) e o Logistic Regression (0,10011 de perda de Hamming e 0,09421 de acurácia).

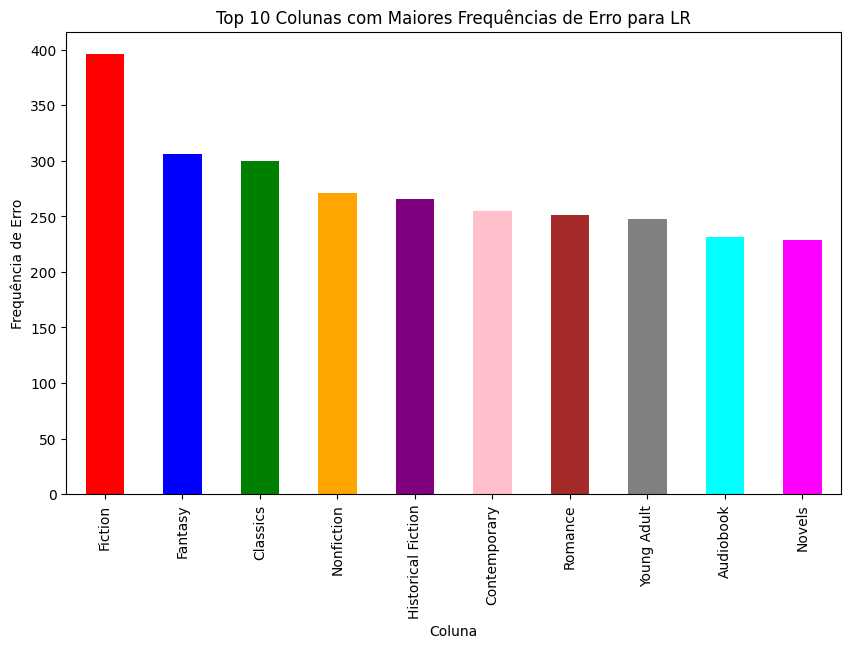
## **7. Conclusão**

Após o estudo do conjunto de dados, foi possível produzir uma série de modelos de aprendizado de máquina capazes de prever quais livros possíveis uma pessoa poderia gostar com base na própria descrição do mesmo. Dentre esses modelos, o que apresentou maior relevância, com as métricas de avaliação foi o LinearSVC, dada a menor perda de Hamming. Além disso, temos o Label Powerset com a pior perda de Hamming, devido que neste conjunto de dados tem-se muitos rótulos, não sendo uma boa utilizar o label powerset, vendo que este aumenta muito o número de classes possível, deixando o problema mais difícil.

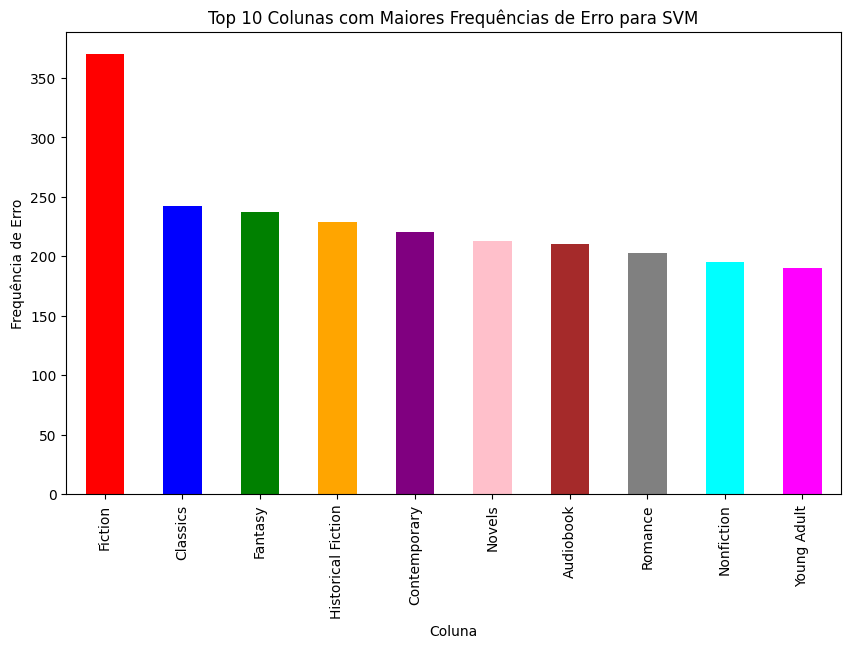
Adicionalmente, a realização deste trabalho, além de permitir revisar conceitos teóricos e suas aplicações práticas com a linguagem Python e suas bibliotecas, também permitiu o estudo de situações inesperadas para os membros da equipe, como o baixo desempenho dos classificadores na métrica de acurácia devido à presença de múltiplos rótulos.

## **8. Reflexões Finais**

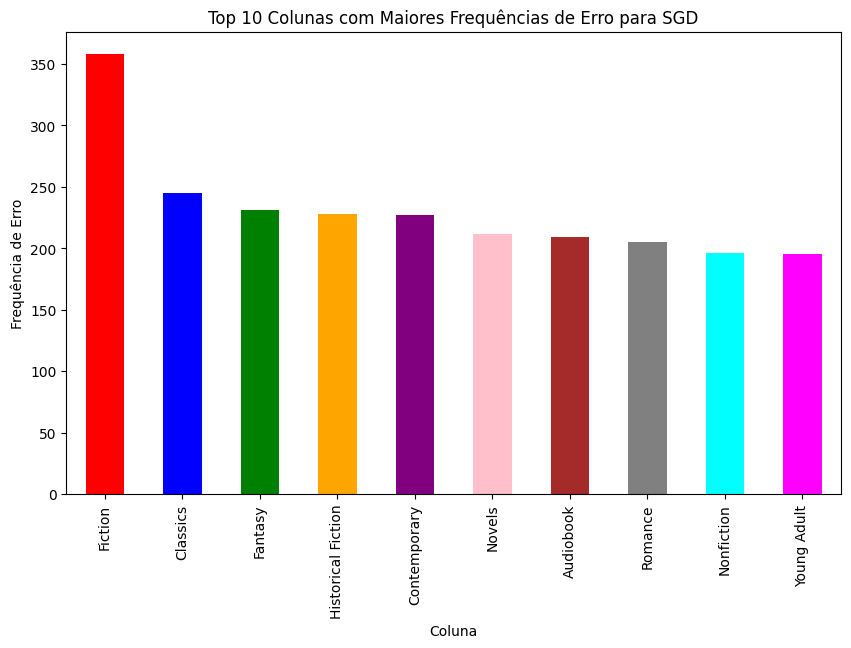
Ao considerar os resultados obtidos, houve uma certa reflexão sobre a qualidade dos modelos e buscou-se o motivo das métricas serem tão baixas. Dito isso, ao explorar os erros, achou-se uma semelhança, o 3 maiores gêneros que acabavam sendo colocados errados ou faltavam ( VF ou FV ) eram Fiction, Fantasy e Classics ( O primeiro sempre sendo o maior e os outros disputando a segunda e terceira classificação ).



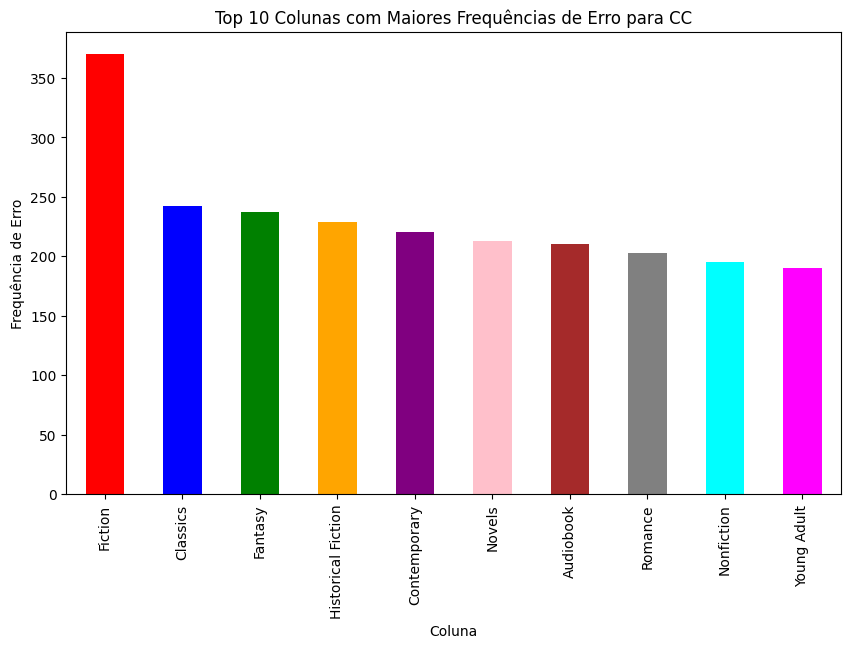
***Imagem 11 -*** *Top10 Gêneros com maiores erros LR*



***Imagem 12 -*** *Top10 Gêneros com maiores erros SVM*

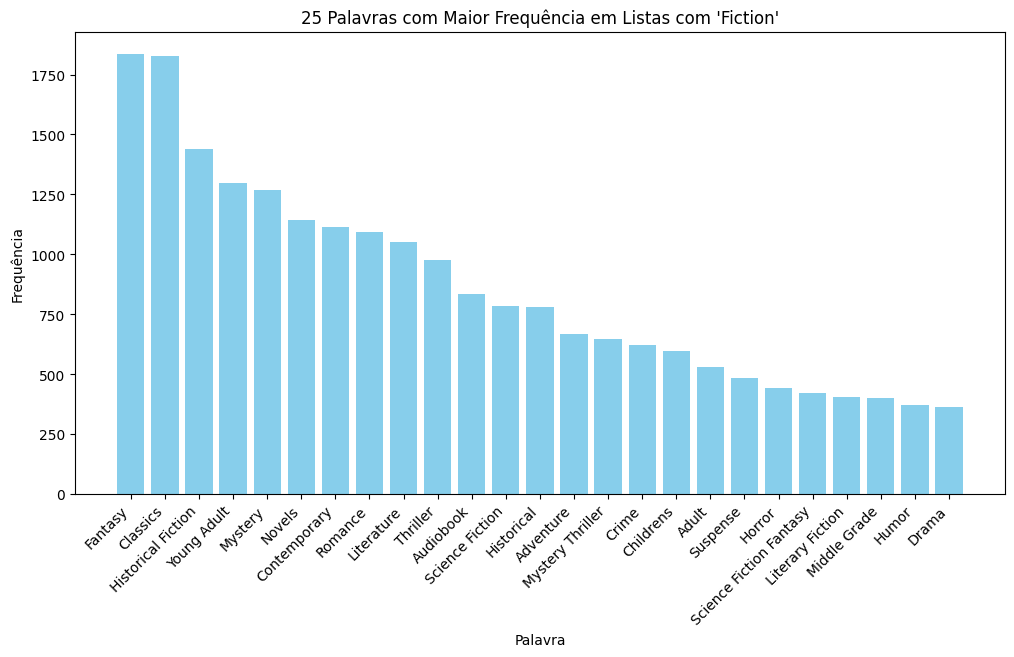


***Imagem 13 -*** *Top10 Gêneros com maiores erros SGD*



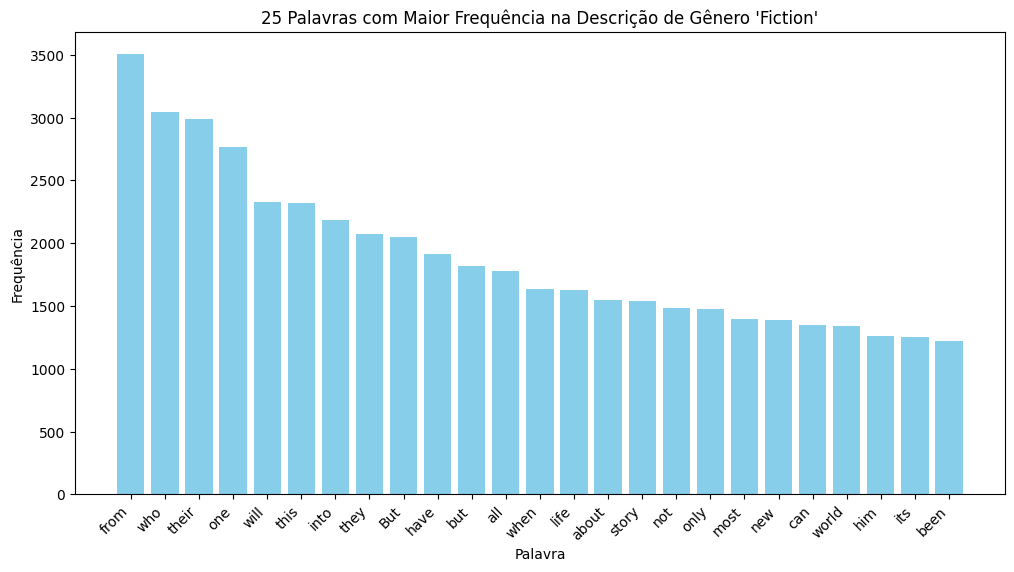
***Imagem 14 -*** *Top10 Gêneros com maiores erros CC com RandomF*

Ademais, ao analisarmos a Classe Fiction, percebe-se que ela aparece em grandes quantidades, mas ao mesmo tempo, junto de outras classes, como as duas que estavam no TOP 2. O que indica que há um certo nível de dependência de dados, mas não muito visto o desempenho demonstrado pelo Classifier chain. Ou seja, há um certo nível em alguns componentes, mas não em todos..

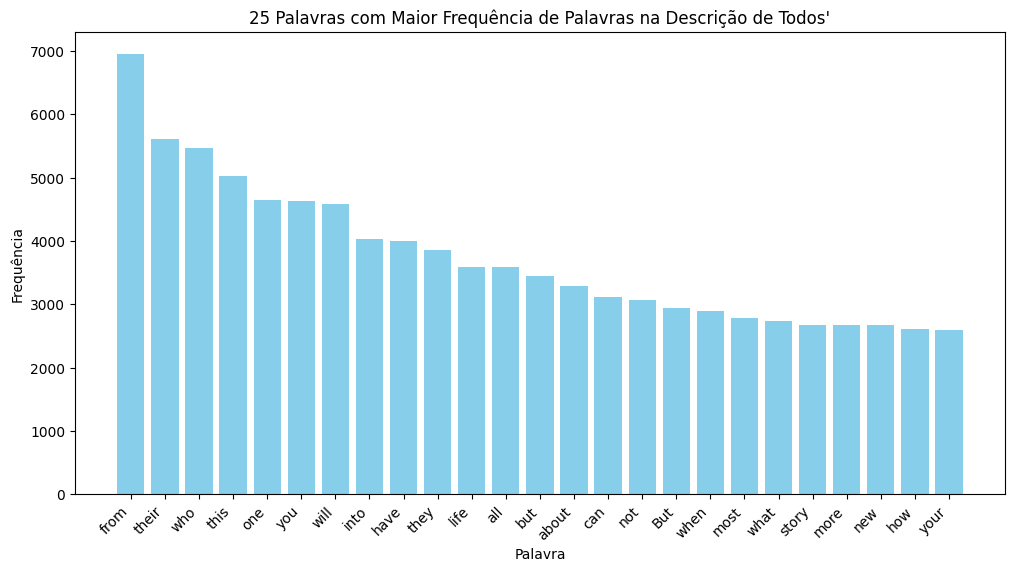


***Imagem 15 -*** *Top25 Gêneros com Maior Frequência junto com “Fiction”*

Explorando ainda mais, ao verificar o nível de aparição de palavras na descrição onde se tem o gênero Fiction. Mas também no geral, obtêm-se os seguintes gráficos.



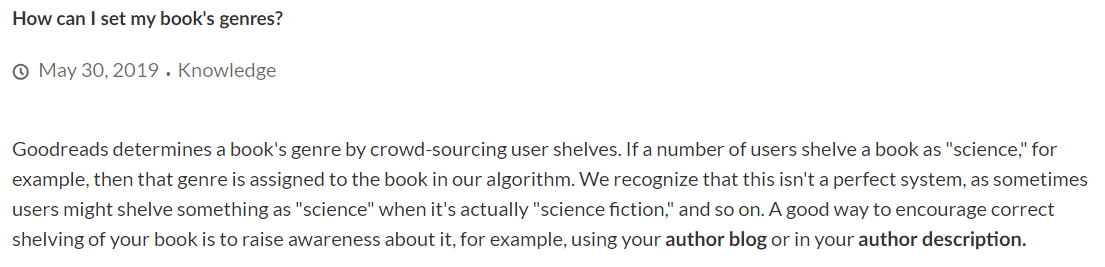
***Imagem 16 -*** *Top25 palavras com Maior Frequência na “Description” de “Fiction”*



***Imagem 17 -*** *Top25 palavras com Maior Frequência na “Description” de Todos*

Ao observar essas imagens, é possível perceber que não há palavras que indicam muito ao gênero de Ficção. Como por exemplo ter uma palavra de “dragons” para Fantasia. Mas também, os termos que usam em fiction, apesar de aparecerem muito, no geral aparecem de forma baixa em relação ao total, não sendo um indício forte da sua ocorrência.

Por fim, foi pesquisado como é feito a seleção de gêneros na plataforma, e que indicou uma razão extremamente forte de terem certas inconsistências. O gênero é uma média de todas as pessoas que classificaram aquele livro. Ou seja, não leva em consideração a opinião dos autores, somente a dos usuários.



***Imagem 18 -*** *Site da Goodreads, como escolhem gêneros*

Dado ao exposto, percebe-se que é um problema muito complexo. Há alguns gêneros, como o Fiction, que possuem certas dependências de outros - aparecem mais com outros. Poucos termos que significam muito para indicar uma classificação mais fácil. Mas também problemas do domínio, onde a própria rotulação é confiável.

## **9. Bibliografia**

[https://www.kaggle.com/datasets/ishikajohari/best-books-10k-multi-genre-data](https://www.kaggle.com/datasets/ishikajohari/best-books-10k-multi-genre-data/data)

[Link da realização do trabalho](https://colab.research.google.com/drive/1Mq1x1XSeP__lWOFhHP7iAt7HTSe8oXwm#scrollTo=wy0Y1dpNbDd7).

<https://help.goodreads.com/s/article/How-can-I-set-my-book-s-genres>

<https://drive.google.com/drive/folders/1E-tKaN6vOOyAQGl5i3Jw33h1GExqCkn->