UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Larissa Vitória Vasconcelos Sousa

1. Introdução

O *Plenty of Fish* (POF) é um popular aplicativo de namoro online, amplamente utilizado em países como Canadá, Reino Unido, Irlanda, Austrália, Nova Zelândia, Espanha, Brasil e Estados Unidos. Como uma das plataformas de namoro mais acessadas, é fundamental para a empresa manter um ambiente positivo e funcional para seus usuários, sendo as avaliações dos usuários um indicador essencial para medir o sucesso dessas iniciativas.

Este trabalho tem como objetivo classificar as avaliações dos usuários com base na pontuação atribuída (estrelas de 1 a 5), utilizando técnicas de Processamento de Linguagem Natural (NLP) e algoritmos de *Machine Learning*. Para a análise, serão consideradas duas variáveis principais, o conteúdo das avaliações e a pontuação atribuída. O *dataset* contém mais de 435 mil registros de avaliações do aplicativo disponíveis na *Google Play Store*.

A Tabela 1 apresenta todas as colunas presentes no dataset de forma abrangente.

Tabela 1 - Base de dados

Nome da coluna	Descrição
Nome do usuário	Nome do usuário que fez a avaliação
Imagem do usuário	Imagem de perfil associada ao usuário
Conteúdo	Comentário deixado pelo usuário sobre o aplicativo
Pontuação	Avaliação do aplicativo, em uma escala de 1 a 5 estrelas
Contagem de polegares para cima	Número de vezes que o comentário foi considerado útil por outros usuários
Versão da revisão	Versão do aplicativo no momento em que a avaliação foi feita
Data de criação da avaliação	Data e hora em que o comentário foi postado
Resposta ao comentário	Resposta da empresa ao comentário do usuário
Data da resposta	Data e hora em que a empresa respondeu ao comentário

ID da avaliação	Identificador único da avaliação

Fonte: Autoria própria (2024).

2. Observações

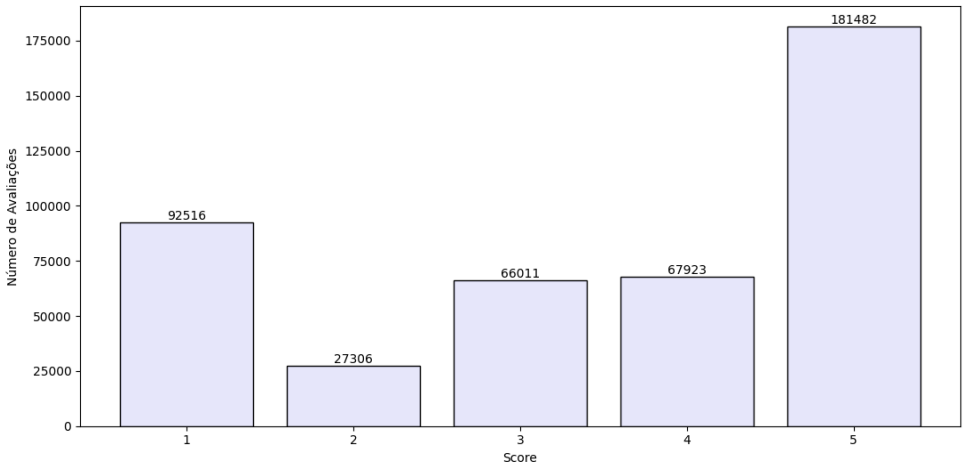
Um contratempo enfrentado foi o tempo prolongado de processamento, causado pelo grande volume de dados e por um erro de memória (`*MemoryError*`). Esse erro provavelmente ocorreu devido ao tamanho do *dataset*, que contém mais de 435 mil linhas, especificamente durante o cálculo manual da matriz *TF-IDF*, que demanda uma quantidade significativa de memória. Para contornar esse problema, optou-se por extrair uma amostra aleatória de 10 mil linhas, o que permitiu continuar o processamento de maneira eficiente. Porém, as acurácias alcançadas não foram satisfatórias, o que indica que o modelo ainda precisa de melhorias para obter resultados melhores.

3. Resultados e discussão

3.1 Questão 1

Antes de selecionar uma amostra de 10 mil linhas e aplicar o balanceamento, foi analisada a distribuição da variável *score* no dataset completo. A Figura 1 a seguir mostra a distribuição, que varia de 1 a 5. Observa-se que a maioria das avaliações (181.482) atribui a nota 5 ao aplicativo. Por outro lado, mais da metade desse número (92.516) deu nota 1.

Figura 1 - Distribuição das notas das avaliações



A tabela abaixo apresenta um exemplo de uma frase do dataset, ilustrando cada etapa do pré-processamento. Embora o pré-processamento tenha sido aplicado a todo o *dataset*, esta linha foi selecionada para visualização como exemplo. As etapas realizadas incluem a remoção de menções, emojis e caracteres especiais, a transformação em minúsculas, a tokenização, a remoção de *stopwords* e a lematização.

Tabela 2 - Frase exemplo do pré processamento

Pré processamento	
Texto original	The best for dating 😍 ❤️
Após remover menções	The best for dating 😍 ❤️
Após remover emojis	The best for dating
Após remover caracteres especiais e transformar em minúsculas	the best for dating
Após tokenizar	['the', 'best', 'for', 'dating']
Após remover stopwords	['best', 'dating']
Após lematização	['best', 'dating']

Fonte: Autoria própria (2024).

3.2 Questão 2

Foram implementadas funções manuais para vetorização utilizando *CountVectorizer* e *TF-IDF*, com o objetivo de serem aplicadas na terceira questão. A Tabela 3 apresenta os resultados obtidos, destacando dois aspectos importantes: a frequência absoluta das palavras em cada documento, calculada pelo *CountVectorizer*, e a importância relativa das palavras em cada documento, medida pelo *TF-IDF*. Embora as funções tenham sido aplicadas ao *dataset*, foram utilizados três documentos para testar o funcionamento dessas funções.

- Doc 1: this is a test
- Doc 2: this test is only a test
- Doc 3: testing is fun

Tabela 3 - *CountVectorizer* e *TF-IDF*

Termo	Count Vectors (Doc 1)	Count Vectors (Doc 2)	Count Vectors (Doc 3)	TF-IDF Vectors (Doc 1)	TF-IDF Vectors (Doc 2)	TF-IDF Vectors (Doc 3)
this	1	1	0	0.25	0.1667	0
is	1	1	1	0.1781	0.1187	0.2374
a	1	1	0	0.25	0.1667	0
test	1	2	0	0.25	0.3333	0
only	0	1	0	0	0.2342	0
testing	0	0	1	0	0	0.4685
fun	0	0	1	0	0	0.4685

Fonte: Autoria própria (2024).

3.3 Questão 3

Utilizando a vetorização por *TF-IDF*, observou-se que a acurácia do classificador foi de 33% com pré-processamento, enquanto sem pré-processamento foi de 20%.

Ao comparar as vetorizações *CountVectorizer* e *TF-IDF*, ambas com pré-processamento, constatou-se que o *TF-IDF* apresentou uma acurácia de 33%, superando o *CountVectorizer*, que obteve 20%. Isso sugere que o *TF-IDF,* com o pré-processamento aplicado, é mais eficaz na representação textual para a tarefa de classificação.

A análise das variações de pré-processamento, especificamente entre lematização e *stemming*, foi realizada utilizando o *TF-IDF*, que demonstrou ser a melhor forma de vetorização no item b. Os resultados mostraram que o *TF-IDF* com *stemming* alcançou uma acurácia de 20%, enquanto o *TF-IDF* com lematização obteve uma pequena redução, com 19%.

4. Conclusões

A análise realizada revelou que as acurácias obtidas foram insatisfatórias. Apesar do uso de técnicas avançadas de Processamento de Linguagem Natural (NLP) e algoritmos de *Machine Learning*, o desempenho dos classificadores não atendeu às expectativas.

O *TF-IDF*, com pré-processamento, alcançou uma acurácia de 33%, enquanto o *CountVectorizer* obteve apenas 20%. Embora o *TF-IDF* tenha superado o *CountVectorizer*, ambos os métodos apresentaram resultados abaixo do esperado. A comparação entre lematização e *stemming* com *TF-IDF* também mostrou pouca diferença na acurácia, com o stemming obtendo 20% e a lematização 19%.

Esses resultados indicam que o código precisa ser revisado e aprimorado para alcançar melhores desempenhos. O baixo nível de acurácia sugere que há espaço para melhorias na configuração dos modelos e no pré-processamento dos dados.

5. Próximos passos

Os próximos passos incluem revisar o pré-processamento, avaliar as técnicas de vetorização e analisar o método de classificação para aprimorar os resultados.