FACULDADE DE TECNOLOGIA
BAIXADA SANTISTA
CIÊNCIA DE DADOS

RELATÓRIO – SIMILARIDADE DE AVALIAÇÕES DE HOTÉIS
ÁLGEBRA LINEAR

Luana Elizabete Oliveira A. Lima

SANTOS – SP
2025

1 INTRODUÇÃO

A análise de avaliações de hotéis em plataformas como o TripAdvisor apresenta um desafio significativo devido ao grande volume de dados e à diversidade de expressões utilizadas pelos usuários. Sistemas de recomendação baseados em similaridade textual tornam-se essenciais para facilitar a navegação e a tomada de decisão dos usuários.

A similaridade de cosseno, um dos métodos mais utilizados para medir a semelhança entre textos, quantifica a similaridade entre dois vetores avaliando o cosseno do ângulo entre eles. No contexto de avaliações de hotéis, cada review pode ser representada como um vetor, onde as dimensões correspondem a diferentes palavras ou frases, e os valores refletem sua importância no texto.

2 OBJETIVO

Este trabalho tem como objetivo desenvolver um sistema de análise de similaridade de avaliações de hotéis utilizando técnicas de Processamento de Linguagem Natural (PLN) e álgebra linear. O sistema permite que o usuário selecione um tópico específico (como localização, limpeza, atendimento) e, com base nessa escolha, receba recomendações de avaliações semelhantes.

3 METODOLOGIA

3.1 Fundamentação Matemática

3.1.1 Similaridade de Cosseno

A similaridade de cosseno é calculada através da fórmula:

$$\text{similaridade}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Onde:

- A e B são vetores de características
- $A \cdot B$ representa o produto escalar entre os vetores
- $\|A\|$ e $\|B\|$ são as magnitudes dos vetores

No código, esta implementação é realizada através da biblioteca `scikit-learn`.

```
vec_base = vectorizer.transform([review_base])
similarities = cosine_similarity(vec_base, tfidf_matrix).flatten()
top_indices = similarities.argsort()[-11:][::-1]
top_indices = [i for i in top_indices if i != random_idx][:10]
```

Figura 1: Autor Próprio

3.1.2 Vetorização TF-IDF

O TF-IDF (Term Frequency-Inverse Document Frequency) transforma o texto em vetores numéricos usando a fórmula:

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \times \log\left(\frac{N}{df(t)}\right)$$

- $\text{tf}(t, d)$ é a frequência do termo t no documento d
- $df(t)$ é o número de documentos contendo o termo t
- N é o total de documentos

A vetorização é implementada com o `TfidfVectorizer` do `scikit-learn`.

```
vectorizer = TfidfVectorizer(
    max_features=5000,
    min_df=2,
    max_df=0.95,
    ngram_range=(1, 2),
    sublinear_tf=True
)
```

Figura 2: Autor Próprio

3.2 Processamento de Dados

3.2.1 Pré-processamento

O pré-processamento envolve várias etapas:

```
def preprocess(text: str) -> str:
    """Limpa, tokeniza e lematiza o texto."""
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    text = ' '.join(text.split())
    tokens = nltk.word_tokenize(text)
    tagged_tokens = nltk.pos_tag(tokens)
    processed_tokens = [
        lemmatizer.lemmatize(word.lower(), get_wordnet_pos(pos))
```

Figura 3: Autor Próprio

- Remoção de caracteres especiais: `re.sub(r'^a-zA-Z', '', text)`
- Conversão para minúsculas: `text.lower()`
- Remoção de números: `re.sub(r'\d+', '', text)`
- Normalização de espaços: `' '.join(text.split())`

Figura 4: Autor Próprio

3.2.2 Definição e Expansão de Tópicos

- Definição dos tópicos: dicionário com palavras-chave por categoria
- Expansão de sinônimos: função `expand_keywords` com auxílio do WordNet
- Pré-processamento das palavras: função `preprocess_words` aplica *stemming* nas palavras expandidas

4 EXTRAÇÃO DOS DADOS

O sistema utiliza o dataset `tripadvisor_hotel_reviews.csv`, que contém avaliações de hotéis com informações como o texto da avaliação, nota atribuída (de 1 a 5), nome do hotel e data da avaliação. Para otimizar o processamento, foi utilizada uma amostra aleatória de 1000 avaliações.

Figura 5: Autor Próprio

5 PROCESSAMENTO DE SIMILARIDADE

5.1 Vetorização e Cálculo de Similaridade

Após o pré-processamento, o sistema realiza a vetorização do texto com TF-IDF e o cálculo da similaridade entre as avaliações por meio da similaridade de cosseno.

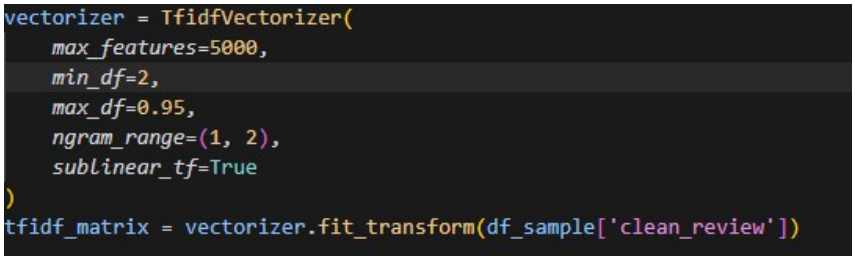

Figura 6: Autor Próprio

5.2 Identificação de Avaliações Similares

O sistema identifica as avaliações mais similares através dos pares com maior grau de similaridade em relação a uma avaliação de referência, escolhida aleatoriamente dentro de um tópico.

```
random_idx = random.choice(df_topico.index)
review_base = df_sample.loc[random_idx, 'clean_review']
review_original = df_sample.loc[random_idx, 'Review']
rating_base = df_sample.loc[random_idx, 'Rating']
tags_base = df_sample.loc[random_idx, 'topics']

vec_base = vectorizer.transform([review_base])
similarities = cosine_similarity(vec_base, tfidf_matrix).flatten()
top_indices = similarities.argsort()[-11:][::-1]
top_indices = [i for i in top_indices if i != random_idx][:10]
```

Figura 7: Autor Próprio

6 RESULTADOS

O sistema desenvolvido apresenta uma interface gráfica intuitiva e eficiente para análise de similaridade de avaliações de hotéis. A interface, construída com `Tkinter`, permite que os usuários selecionem tópicos específicos e visualizem imediatamente as avaliações mais relevantes.

Os resultados são apresentados de forma organizada e clara, com destaque para:

- A avaliação de referência selecionada aleatoriamente
- As 10 avaliações mais similares
- Notas atribuídas (de 1 a 5)
- Tópicos identificados
- Grau de similaridade calculado

A combinação da interface gráfica com o processamento de similaridade de cosseno permite uma análise rápida e precisa das avaliações, facilitando a identificação de padrões e a tomada de decisão dos usuários. O sistema demonstra eficácia tanto na categorização automática das avaliações quanto na apresentação clara dos resultados, tornando a análise de grandes volumes de avaliações mais acessível e eficiente.