

# Lista 4 - Relatório de Busca de Animes

Gabriel Medina, Jonatas Fernandes

Abril 2023

## 1 Introdução

Neste relatório será descrita a busca de animes feita com base em um conjunto de dados em formato CSV e implementada em Python. O objetivo da busca é encontrar os animes mais similares a um anime escolhido pelo usuário. Serão descritas as etapas de codificação, a base de dados utilizada, um exemplo de busca e resultados obtidos.

## 2 Base de Dados

A base de dados utilizada contém informações de cerca de 16000 coletados em 2020, e foi obtida através do site Kaggle.com. O arquivo CSV, contém informações como o nome do anime, gênero e a sinopse do anime.

Para preparar a base de dados para a busca, realizamos algumas modificações no conjunto de dados original. Utilizamos o código a seguir para remover palavras consideradas irrelevantes (Stop words) e também para excluir colunas que não seriam úteis na busca.

```
import pandas as pd
from nltk.corpus import stopwords

df = pd.read_csv('old_anime_with_synopsis.csv')

def remove_stopwords(df):
    stop_words = set(stopwords.words('english'))
    df['synopsis'] = df['synopsis'].fillna('')
    df['synopsis'] = df['synopsis'].apply(lambda x: ' '.join([word for word in x.split()
    if word.lower() not in stop_words]))
    df = df.rename(columns={'synopsis': 'synopsis'})
    df = df.drop(columns=['MAL_ID', 'Score', 'Genres'])
    return df

df = remove_stopwords(df)
df.to_csv('anime_with_synopsis.csv', index=False)
print("Execution successful. The database 'anime_with_synopsis.csv' is now available for use.")
```

### 3 Codificação

Primeiro, importamos as bibliotecas necessárias e definimos as variáveis necessárias, incluindo o caminho do arquivo CSV e o tamanho do chunk. Em seguida, lemos o arquivo CSV e o transformamos em um DataFrame. A seguir, preenchemos as colunas vazias com uma string vazia e criamos uma matriz de token de contagem usando a função `CountVectorizer`. Em seguida, criamos a matriz de similaridade entre as sinopses dos animes com a função `cosine similarity`.

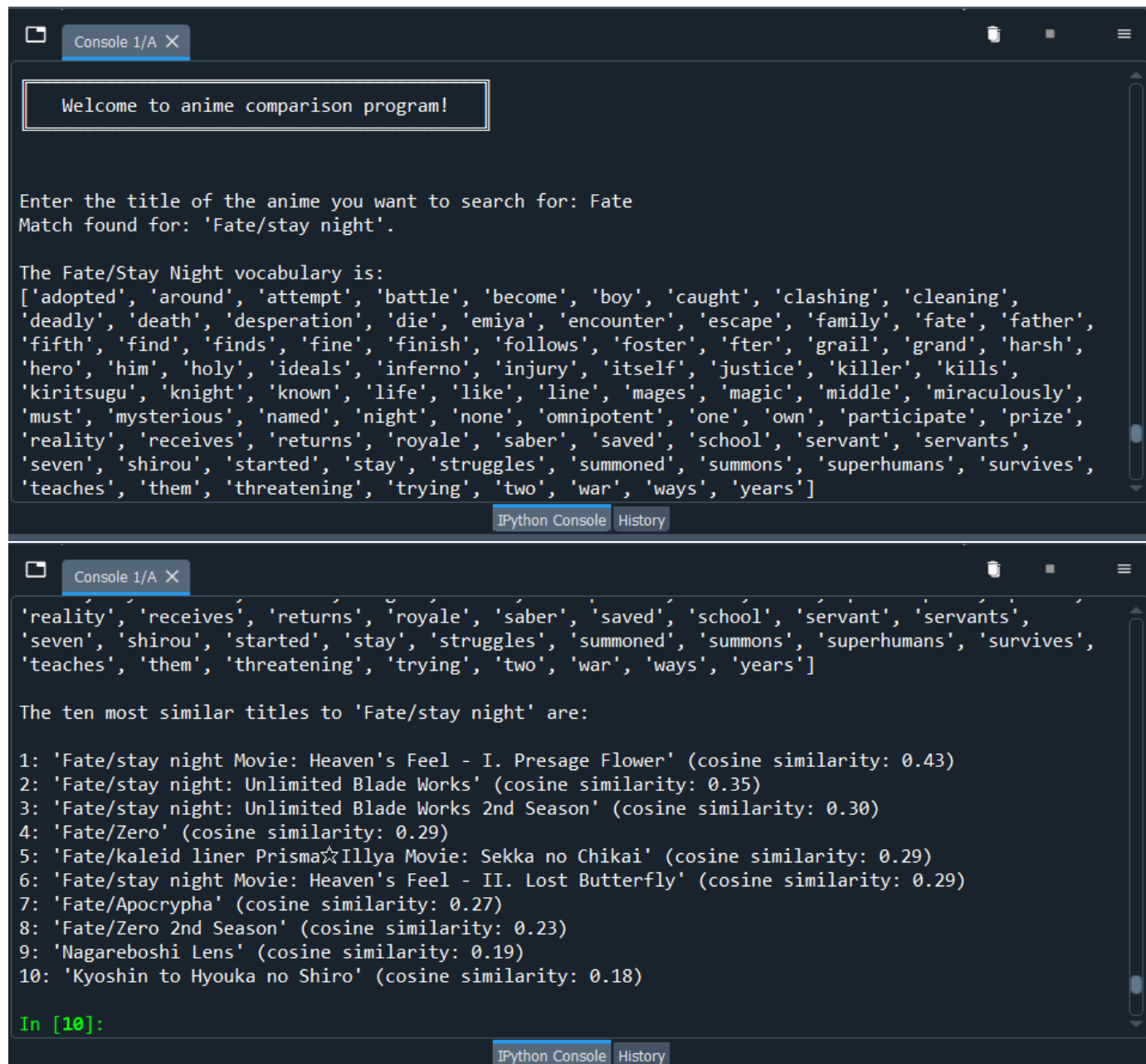
Solicitamos ao usuário o título do anime que deseja buscar. O programa verifica se o título está na lista de animes e, se estiver, seleciona a sinopse do anime para calcular a similaridade com outros animes. Caso contrário, o programa sugere uma pesquisa alternativa, fornecendo títulos semelhantes com base na sinopse. Se não houver títulos semelhantes, o programa solicita uma nova pesquisa.

Após a busca, o programa exibe a vetorização das palavras da sinopse do anime de entrada e os 10 animes mais similares, juntamente com o valor do ângulo de similaridade. A similaridade é calculada usando o cosseno do ângulo entre as duas sinopses.

A similaridade de cosseno é um algoritmo que mede a similaridade entre vetores de alta dimensionalidade. Neste caso, cada palavra na sinopse do anime é tratada como uma dimensão e a similaridade é calculada em um espaço vetorial de alta dimensão. O algoritmo considera a frequência de cada palavra em cada sinopse e, em seguida, calcula o ângulo entre as duas sinopses para determinar a similaridade.

## 4 Exemplo

Utilizamos a palavra "Fate" em uma busca e o programa fez a correção para "Fate/stay night", resultando na localização exata de 85 dimensões, ou seja, 85 palavras presentes no anime "Fate/stay night" e em outras sinopses armazenadas no banco de dados, além disso, o programa nos apresentou os 10 animes mais semelhantes a "Fate/stay night", como ilustrado nas imagens abaixo.



The image shows two screenshots of a Python console application. The top screenshot displays the program's welcome message and the results of a search for the anime "Fate". The search found "Fate/stay night" and listed its vocabulary. The bottom screenshot shows the results of a search for the anime "Fate/stay night", displaying the ten most similar titles to it.

```
Console 1/A X

Welcome to anime comparison program!

Enter the title of the anime you want to search for: Fate
Match found for: 'Fate/stay night'.

The Fate/Stay Night vocabulary is:
['adopted', 'around', 'attempt', 'battle', 'become', 'boy', 'caught', 'clashing', 'cleaning',
'deadly', 'death', 'desperation', 'die', 'emiya', 'encounter', 'escape', 'family', 'fate', 'father',
'fifth', 'find', 'finds', 'fine', 'finish', 'follows', 'foster', 'fter', 'grail', 'grand', 'harsh',
'hero', 'him', 'holy', 'ideals', 'inferno', 'injury', 'itself', 'justice', 'killer', 'kills',
'kiritsugu', 'knight', 'known', 'life', 'like', 'line', 'mages', 'magic', 'middle', 'miraculously',
'must', 'mysterious', 'named', 'night', 'none', 'omnipotent', 'one', 'own', 'participate', 'prize',
'reality', 'receives', 'returns', 'royale', 'saber', 'saved', 'school', 'servant', 'servants',
'seven', 'shirou', 'started', 'stay', 'struggles', 'summoned', 'summons', 'superhumans', 'survives',
'teaches', 'them', 'threatening', 'trying', 'two', 'war', 'ways', 'years']

Python Console History

Console 1/A X

'reality', 'receives', 'returns', 'royale', 'saber', 'saved', 'school', 'servant', 'servants',
'seven', 'shirou', 'started', 'stay', 'struggles', 'summoned', 'summons', 'superhumans', 'survives',
'teaches', 'them', 'threatening', 'trying', 'two', 'war', 'ways', 'years']

The ten most similar titles to 'Fate/stay night' are:

1: 'Fate/stay night Movie: Heaven's Feel - I. Presage Flower' (cosine similarity: 0.43)
2: 'Fate/stay night: Unlimited Blade Works' (cosine similarity: 0.35)
3: 'Fate/stay night: Unlimited Blade Works 2nd Season' (cosine similarity: 0.30)
4: 'Fate/Zero' (cosine similarity: 0.29)
5: 'Fate/kaleid liner Prisma☆Illya Movie: Sekka no Chikai' (cosine similarity: 0.29)
6: 'Fate/stay night Movie: Heaven's Feel - II. Lost Butterfly' (cosine similarity: 0.29)
7: 'Fate/Apocrypha' (cosine similarity: 0.27)
8: 'Fate/Zero 2nd Season' (cosine similarity: 0.23)
9: 'Nagareboshi Lens' (cosine similarity: 0.19)
10: 'Kyoshin to Hyouka no Shiro' (cosine similarity: 0.18)

In [10]:

Python Console History
```