**Relatório**

**Guião PL04**

Métodos Probabilísticos para Engenharia Informática

Departamento de Eletrónica, Telecomunicações e Informática

Prof. Amaro Sousa

Ano letivo 2022/2023

Turma P6

Isaac Moura, 105065

Miguel Gomes, 103826

06/01/2023

# Índice

[Índice 2](#_Toc62760458)

[Introdução 3](#_Toc62760459)

[Estrutura de Dados 4](#_Toc62760460)

[Aplicação 6](#_Toc62760461)

[Menu 6](#_Toc62760462)

[Função yourMovies 8](#_Toc62760463)

[Função SuggestionsOtherUsers 9](#_Toc62760464)

[Função searchTitle 11](#_Toc62760465)

[Anotações 13](#_Toc62760466)

# Introdução

A aplicação tem duas componentes, cada uma num script de MatLab. Uma componente é a estrutura de dados da aplicação, ou seja, vai conter as estruturas e funções usadas para manipular os dados dos utilizadores. O código presente neste script só necessita de ser executado uma vez para gerar todos os dados necessários sendo apenas preciso repetir o processo se quisermos inserir novos dados, como filmes ou utilizadores. O código armazena dados como nomes de filmes, nomes de utilizadores, função MinHash para cada filme e respetivo shingle. O outro script é responsável por executar a aplicação carregando os dados do outro script.

# Estrutura de Dados



Na determinação do valor de k da MinHash dos filmes utilizamos as conclusões obtidas no exercício 4 da parte 4.3 do guião prático. Foi medido o tempo de calculo da MinHash para certos valores de k assim como as distâncias de Jaccard para os utilizadores que deram valores similares. Usando estes dados em conjunto com o real valor da distância de Jaccard chegamos a conclusão de que 100 seria um bom número para k pois junta a precisão dos valores obtidos com um tempo de execução razoável.

Quanto ao tamanho dos shingles o enunciado recomenda um valor entre 2 e 5 e , após alguns testes, a diferença do tempo de execução não foi muito relevante pelo que optamos por shingles com tamanho 3.

Aumentando os valores de k na MinHash conseguimos sempre aumentar a precisão do valor das distâncias de Jaccard. Aumentar estes valores tem como consequência o aumento do tempo de execução do código pelo que é necessário encontrar um ponto de equilíbrio conforme cada caso de MinHash.

Na decisão do valor de k para a MinHash dos shingles foi escolhido um valor intermédio que fornece uma precisão aceitável para a funcionalidade tendo em conta o tempo de execução. Não é benéfico ter um nível muito elevado de precisão se a aplicação demora um tempo descabido a correr.

# Aplicação



Como temos os dados pré carregados basta fazer load. A função menu é chamada uma vez que serve de base a todo o programa.

## **Menu**

****

Temos uma função menu que vai fornecer ao utilizador todas as operações que pode efetuar na aplicação. Depois de efetuar o “login”, ou seja, de inserir o seu numero(id), o programa vai listar as operações disponíveis. As opções vão do 1 ao 5 e utilizam como dados os conjuntos da estrutura de dados como a MinHash, lista de filmes, lista de utilizadores e lista de filmes vistos e avaliados.

Tivemos em atenção possíveis inputs errados por parte do utilizador e por isso a aplicação ignora-os e volta a pedir o número de utilizador quando este é inválido.

## **Função yourMovies**



Esta função lista todos os filmes que o utilizador que está a usar o sistema já viu. Tem como parâmetros o Id do utilizador, a lista de todos os filmes vistos por cada utilizador (YourMoviesTable) e ainda a lista que tem todos os filmes e respetivas categorias(uFilms).

## **Função SuggestionsOtherUsers**

****

Esta função sugere títulos de filmes que ainda não foram avaliados pelo utilizador atual, provenientes dos dois utilizadores que são mais similares a ele.

Para encontrar os utilizadores semelhantes é usado a MinHashValue com os valores previamente calculados. Usando estes valores é possível calcular a distância de Jaccard. Foram mantidos os valores de K usados na estrutura de dados. A função ismember é responsável por filtrar aqueles filmes recomendados que o nosso utilizador já avaliou.

## **Função searchTitle**

****

A função searchTitle tem como objetivo fornecer uma lista de filmes baseada numa string inserida pelo utilizador. Esta lista contém os 5 filmes cujo título é mais parecido com a string inserida. Usamos uma estrutura semelhante da função SuggestionsCategories para criar os shingles e formação da MinHash para cada um. As distâncias de Jaccard entre os shingles da string introduzida e dos filmes foram comparadas para encontrar um valor mínimo.

Caso a nossa string não esteja dentro do limite de pesquisa o programa responde que não foi possível encontrar uma correspondência.

A lista de filmes aparece ordenada pelas strings que mais são parecidas com a string inserida pelo utilizador.

# Anotações

A função de hash usada na elaboração deste trabalho foi a “DJB31MA.m” que foi fornecida no guião prático 4. O valor de k da função foi sendo alterado conforme as necessidades o que permitiu obter várias funções hash dentro de uma só, variando um parâmetro.

