

**Conceção e implementação de um sistema de recomendação.**

Grupo 9

Sistemas Baseados em similaridades

Autores:

João Sá

Tiago Silva

**Índice**

**1. Introdução1**

**2. Data Treatment4**

2.1 Datasets Iniciais4

2.2 Criação de géneros nas piadas5

2.3 Categorização da variável Rating6

2.4 Ordenação de piadas por género e rating7

2.5 Sumário8

**3. Implementação do sistema de recomendação9**

3.1 Tecnologias utilizadas9

3.2 Dataset Final9

3.3 Filtragem por conteúdo10

3.4 Filtragem Colaborativa11

3.4.1 Métricas utilizadas11

3.4.2 Seleção de piadas …………………………………………………………………………………………………………….12

3.4.3 Feedback Sistema…………………………………………………………………………………………………………….13

3.5 Futuro Sistema ……………..…………………………………………………………………………………………………………….13

**4. Tutorial de utilização ……………………………………………………………………………………………………………………14**

**5. Conclusão15**

1. **Introdução**

Este trabalho tem como finalidade a Conceção e implementação de um Sistema de Recomendação.

Foi escolhida uma base de dados constituída por:

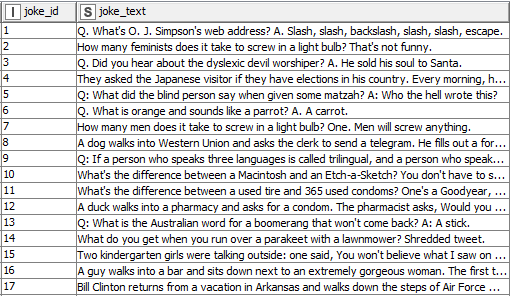
* 139 Anedotas
* 40863 Utilizadores

Cada utilizador avaliou diferentes anedotas e o objetivo deste trabalho é criar um sistema de recomendação que consiga recomendar piadas para um novo utilizador. O sistema irá ser implementado usando técnicas de filtragem colaborativa e filtragem por conteúdo.

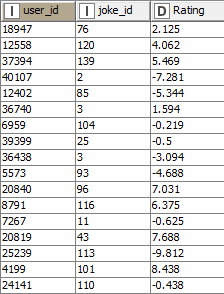
O sistema de recomendação irá estar integrado em uma plataforma web de modo a facilitar a interação do utilizador com o sistema tornando a experiência também mais apelativa.

1. **Data treatment**

2.1. Datasets Iniciais

O tratamento dos dados foi realizado na plataforma ”KNIME”. O dataset estava dividido em dois ficheiros com formato “CSV”, no primeiro “CSV” tínhamos o Id das piadas e o texto das piadas como mostra a figura 1.

*Fig1. Dataset com as piadas*

 No segundo “CSV” tínhamos o Identificador do Usuário, Identificador da piada e o Rating da piada desse Usuário como mostra a figura 2.

*Fig2. Dataset com o rating dos usuários*

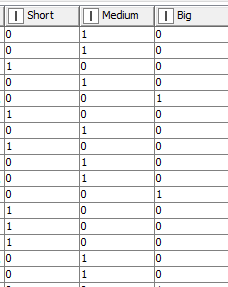
2.2. Criação de géneros nas piadas

Começamos por juntar ambos os “CSV”, para fazermos isso utilizamos o Node ”Joiner”. Após juntar ambos os “CSV” dividiu-se as piadas em três categorias:

* + Grande
  + Média
  + Pequena

 Para isso utilizamos o node “Java Snippet” onde determinamos o tamanho de cada uma das piadas como mostra a figura 3.

*Fig.3 java snippet para obter o número de caracteres da piada*

 Após termos o tamanho de cada piada decidimos que o Length abaixo de 200 as piadas eram curtas, Length entre 200 e 700 as piadas eram médias e acima de 700 as piadas eram grandes. Para isso utilizamos três Rule Engines, com isto ficamos com três colunas (Short, Medium, Big) para avaliar cada uma das piadas.

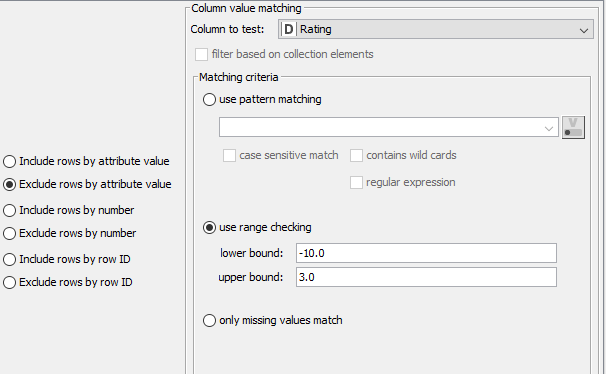
*Fig4. Binary Enconding no género das piadas*

Aplicamos uma técnica denominada de “Binary Encoding” como mostra a fig.4. Depois de categorizarmos cada umas das piadas como pequena, média ou grande tínhamos outro objetivo para tratar.

2.3. Categorização da variável Rating

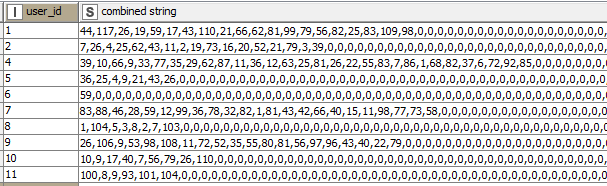
A variável Rating tinha valores entre -10 e 10, para simplificarmos o nosso dataset decidimos categorizar a nossa variável em três valores diferente (gostou, mais ou menos e não gostou) e finalmente determinarmos a opinião de cada utilizador para cada uma das piadas.

Para isso utilizamos diferentes nodes, começamos por determinar que todas as piadas com Rating acima de 3 o utilizador gostou, Rating [-3, 3] considerou razoável e Rating abaixo de -3 não gostou onde no final tínhamos o id de cada usuário e a opinião de cada piada.

 Para dividirmos o dataset em três grupo utilizamos o node “Row Filter”. Na figura 5 temos o “Row Filter” utilizado para determinar as piadas que cada utilizador gostou, o processo foi similar para todas as categorias.

*Fig.5 Obtenção de piadas gostadas por cada utilizador*

Para cada um dos “CSV” deixamos como 0 todas as anedotas que não correspondiam, por exemplo se o utilizador não tinha gostada de uma piada o valor ficava a 0. Na figura abaixo conseguimos obter uma melhor perceção do resultado.



*Fig6. Dataset tratado com as piadas gostadas para cada utilizador*

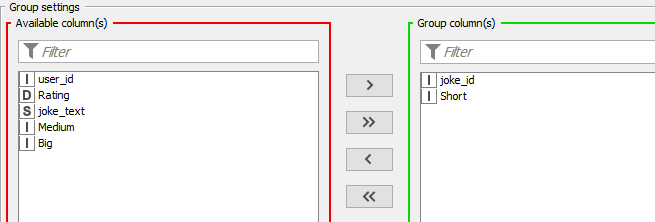
Tal como mostra na figura, temos o id de cada utilizador e o id de cada piada que o utilizador gostou, todas as outras piadas deixamos com o valor igual a zero. Repetimos o processo para as piadas que cada utilizador achou mais ou menos e para as piadas que cada utilizador não gostou.

Com isto já organizamos os nossos dados conforme a opinião de cada utilizador e categorizámos cada piada pelo tamanho. O atributo tamanho será utilizado no início do nosso sistema de recomendação quando não tivermos qualquer opinião do novo utilizador, vamos deixar o utilizador escolher se pretende piadas longas, medias, curtas ou diferentes combinações entre eles. Esta técnica permite evitar o chamado “Cold Start”.

2.4. Ordenação de piadas por género e rating

Para terminarmos o nosso trabalho no KNIME e o tratamento dos dados decidimos também obter a média de avaliações para cada uma das piadas e assim conseguimos descobrir quais as piadas curtas, médias ou grandes mais bem pontuadas.  
 O objetivo deste processo é na escolha das primeiras piadas o nosso sistema de recomendação escolher aleatoriamente entre as melhores piadas.

Para a realização deste processo utilizados o node” Group By” onde fizemos a média do Rating para cada piada como mostram as figuras abaixo.



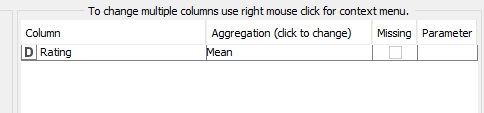
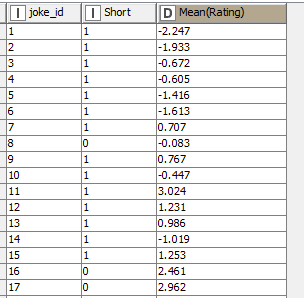


Fig.9 Resultado final

Fig.8. Média de Rating para cada Joke\_Id

Fig.1. Calcular para cada Joke\_Id

Para terminarmos utilizamos o node “Sorter” para ordenar a média de Rating de cada uma das piadas.

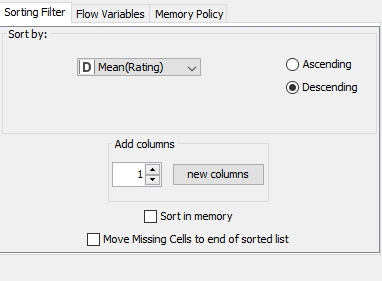


Fig.2 Ordenação da média de Rating

Realizamos este processo para os três tipos de piada (curta, média ou grande) e no final exportamos os três “CSV” para utilizarmos no python na realização do nosso sistema de recomendação.

* 1. Sumário

Recapitulando o tratamento de dados realizado no KNIME:

* Categorizar cada uma das piadas como pequena, média ou grande;
* Categorizar o Rating como gostou, mais ou menos, não gostou;
* Obtermos a média de Rating de cada uma das piadas e ordenarmos da melhor para a pior;

No final ficámos com 6 “CSV”:

* Piadas que cada utilizador gostou;
* Piadas que cada utilizador achou mais ou menos;
* Piadas que cada utilizador não gostou;
* Top piadas Pequenas;
* Top piadas Médias;
* Top piadas Grandes;

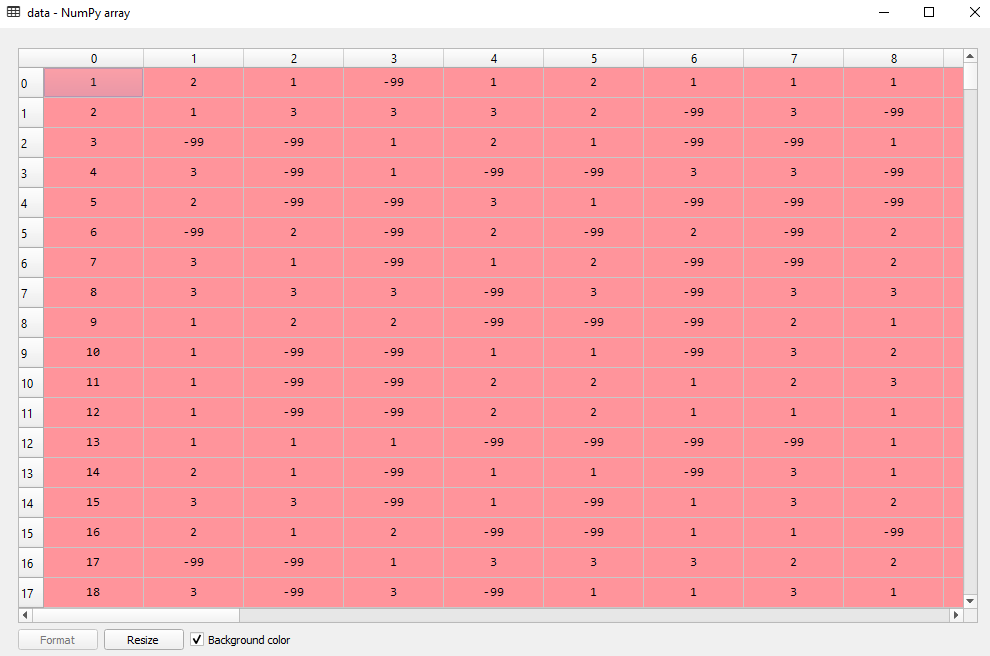
1. **Implementação do sistema de recomendação**
   1. Tecnologias utilizadas

Para implementar o sistema de recomendação utilizou-se a linguagem de programação Python e o IDE “Spyder”. Foram usadas as librarias comuns a projetos desta categoria como numpy, pandas, scipy e matplotlib.

Para implementação da plataforma web utilizou-se a framework “FLASK” devido a sua simplicidade e facilidade de uso que no entanto permite atingir os nossos objetivos.

* 1. Dataset final

Usando os datasets preparados do “Knime” que continham as piadas gostadas por cada utilizador criou-se uma matriz [40863, 140] sendo que cada linha representa um utilizador e cada coluna representa o rating desse utilizador para cada uma das 139 piadas, com exceção da primeira coluna que contem o id de cada utilizador.

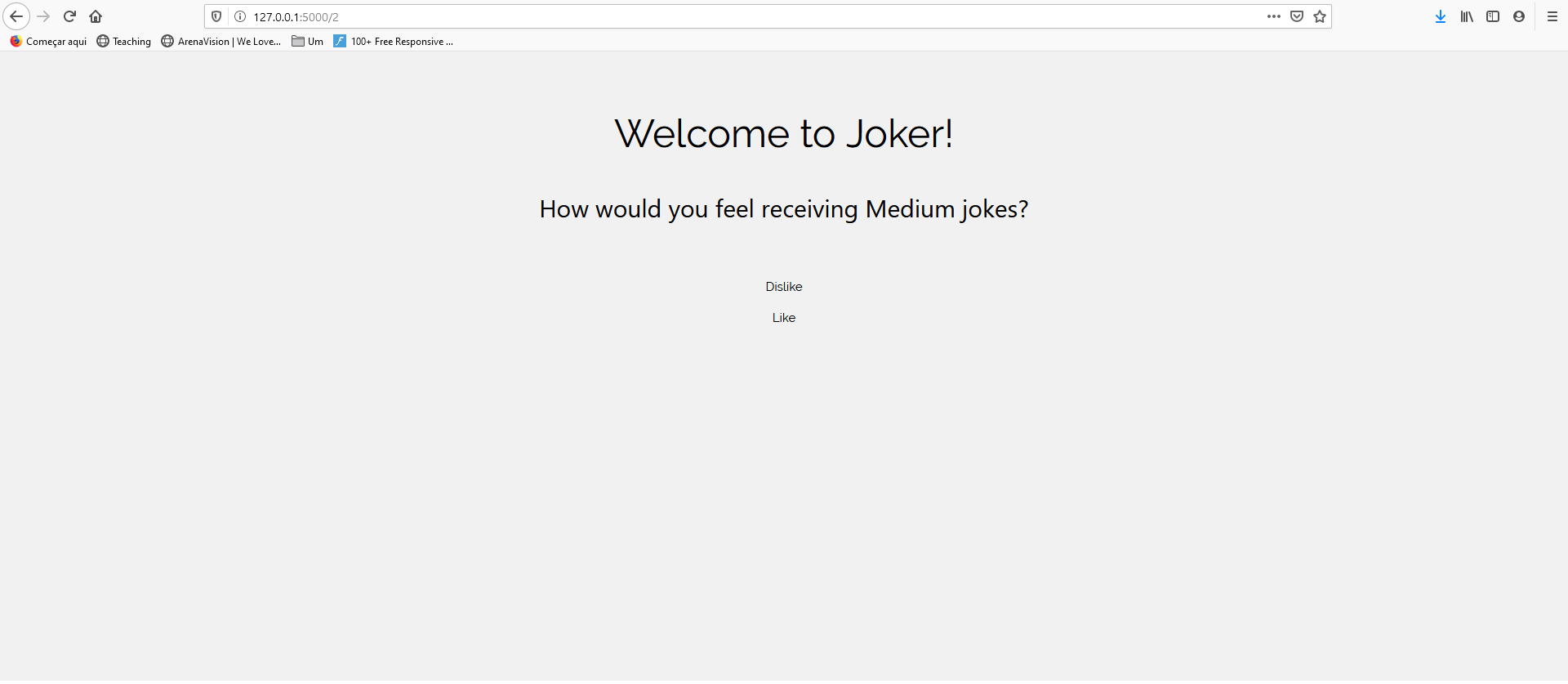


*Fig.11 Dataset Final*

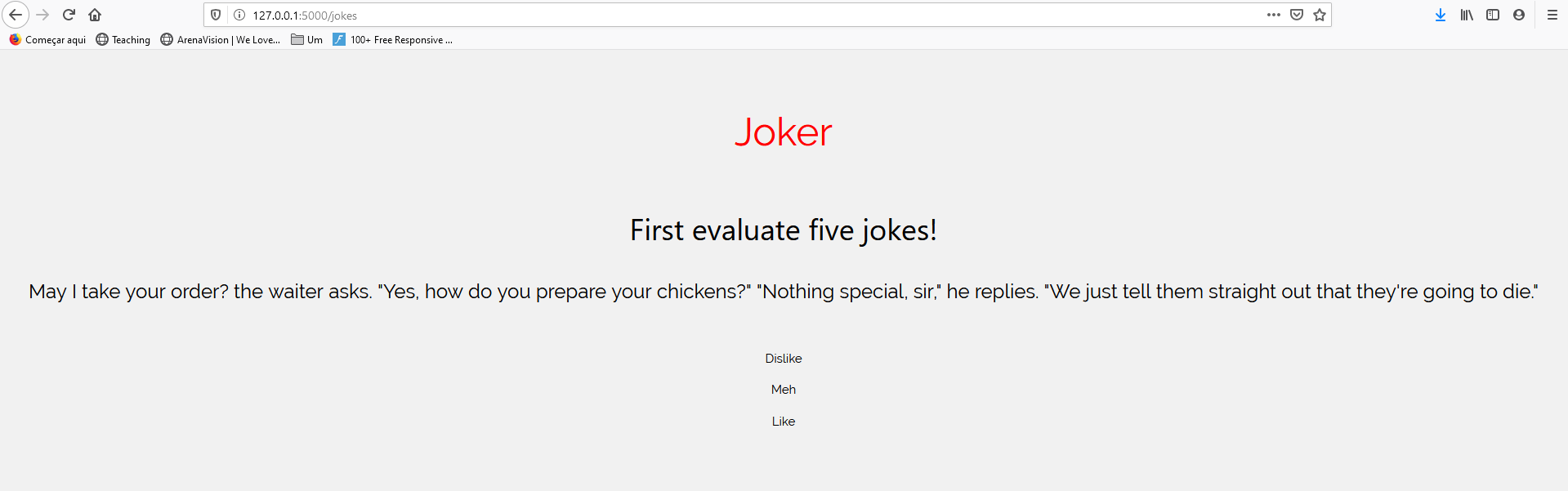
* 1. Filtragem por conteúdo

Filtragem por conteúdo baseia-se na premissa de utilizar os meta dados do conteúdo de modo a fazer recomendações ao utilizador. No nosso sistema esta técnica é utilizada no nosso sistema de modo a evitar um “cold start”. Quando o utilizador inicialmente abre o nosso website são feitas três questões de modo a perceber de que tipo de piadas o utilizador gosta, Curtas, médias ou longas, fig12. As opções escolhidas serão o tipo de piadas inicialmente apresentadas ao utilizador.

As piadas fornecidas são escolhidas de entre as mais bem cotadas pelos utilizadores em média de modo a tentar fornecer as piadas que mais agradam no geral, fig 13.



*Fig12. Exemplo de questão ao utilizador*



*Fig13.Primeiras cinco piadas utilizando filtragem por conteúdo*

* 1. Filtragem colaborativa

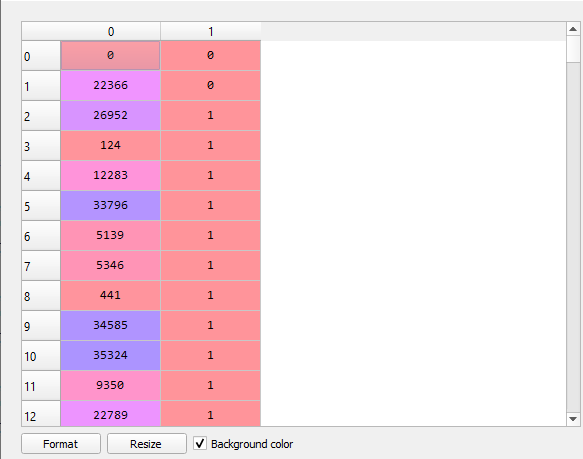
Filtragem colaborativa têm com premissa que utilizadores semelhantes tem gostos semelhantes e, portanto, é encontrado um ou vários utilizadores semelhantes ao utilizador e este recebe as recomendações baseadas nos gostos dos utilizadores semelhantes

No nosso sistema esta técnica começa a ser implementada após o utilizador avaliar as primeiras cinco piadas pois consideramos que com esta informação já se consegue encontrar utilizadores relativamente semelhantes

3.4.1. Métricas utilizadas

Para medir a distância entre dois utilizadores foi utilizada a distância euclidiana, apenas piadas que foram avaliadas pelos dois utilizadores contam para a distância pois só deste modo podemos avaliar a diferença de gostos.

Como o nosso dataset de utilizadores tem cerca de 40000 utilizadores diferentes consegue-se medir a distância do utilizador atual e cada um dos utilizadores do datasets com um desempenho bastante satisfatório, daqui retorna uma lista ordenada contendo todas as distâncias, figura 14.

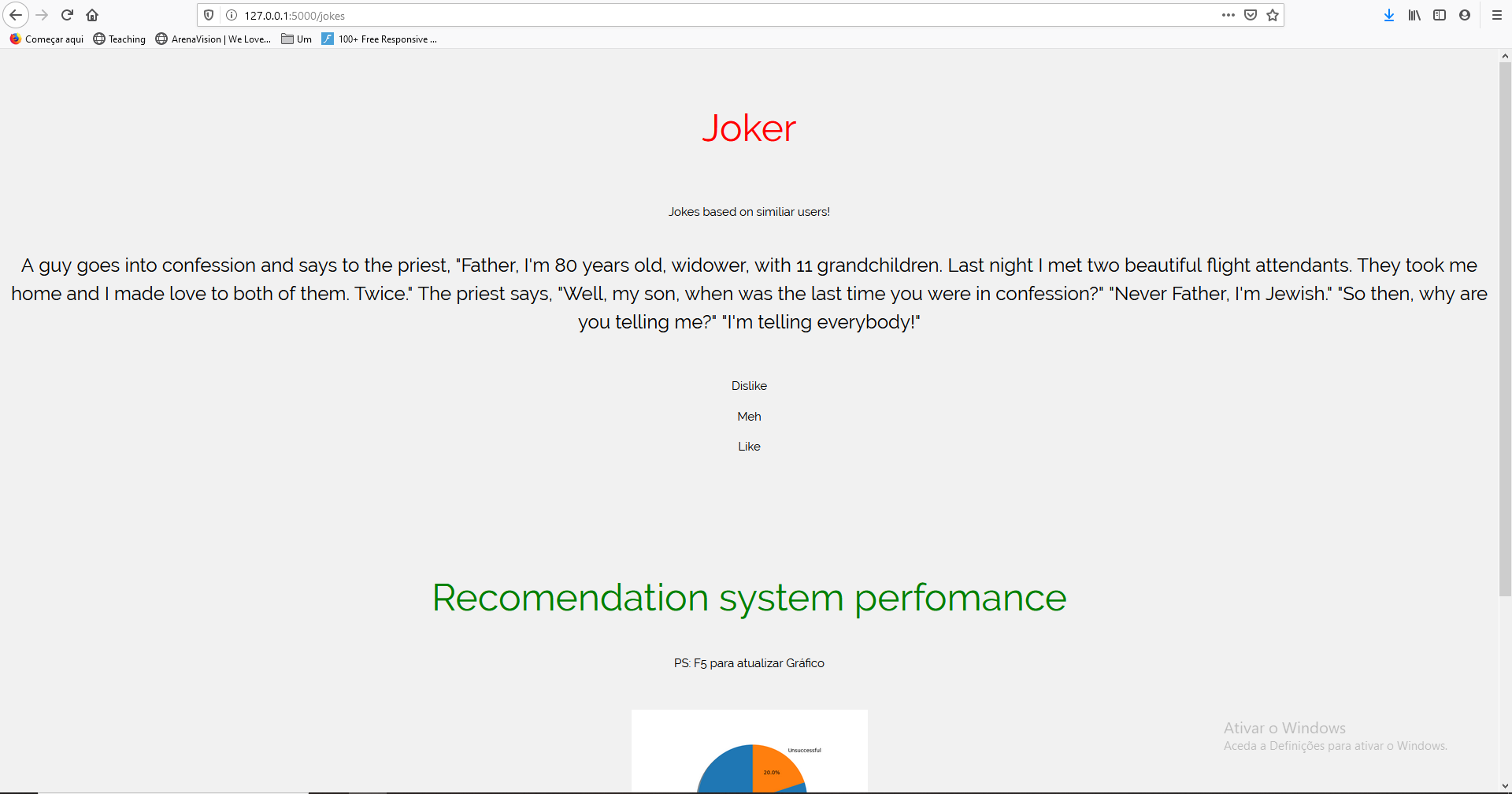


*Fig.14 Lista de distâncias*

Neste caso em específico pode-se observar que o utilizador da base de dados mais próximo do utilizador atual é o 22366 que tiveram exatamente os mesmos gostos. Este processo é repetido para cada nova piada que o utilizador avalie e a distância entre dois utilizadores apenas é validade se tiverem pelo menos cinco piadas em comum para evitar que utilizadores que só tem uma piada em comum tenham uma distância de zero.

* + 1. Seleção de piada

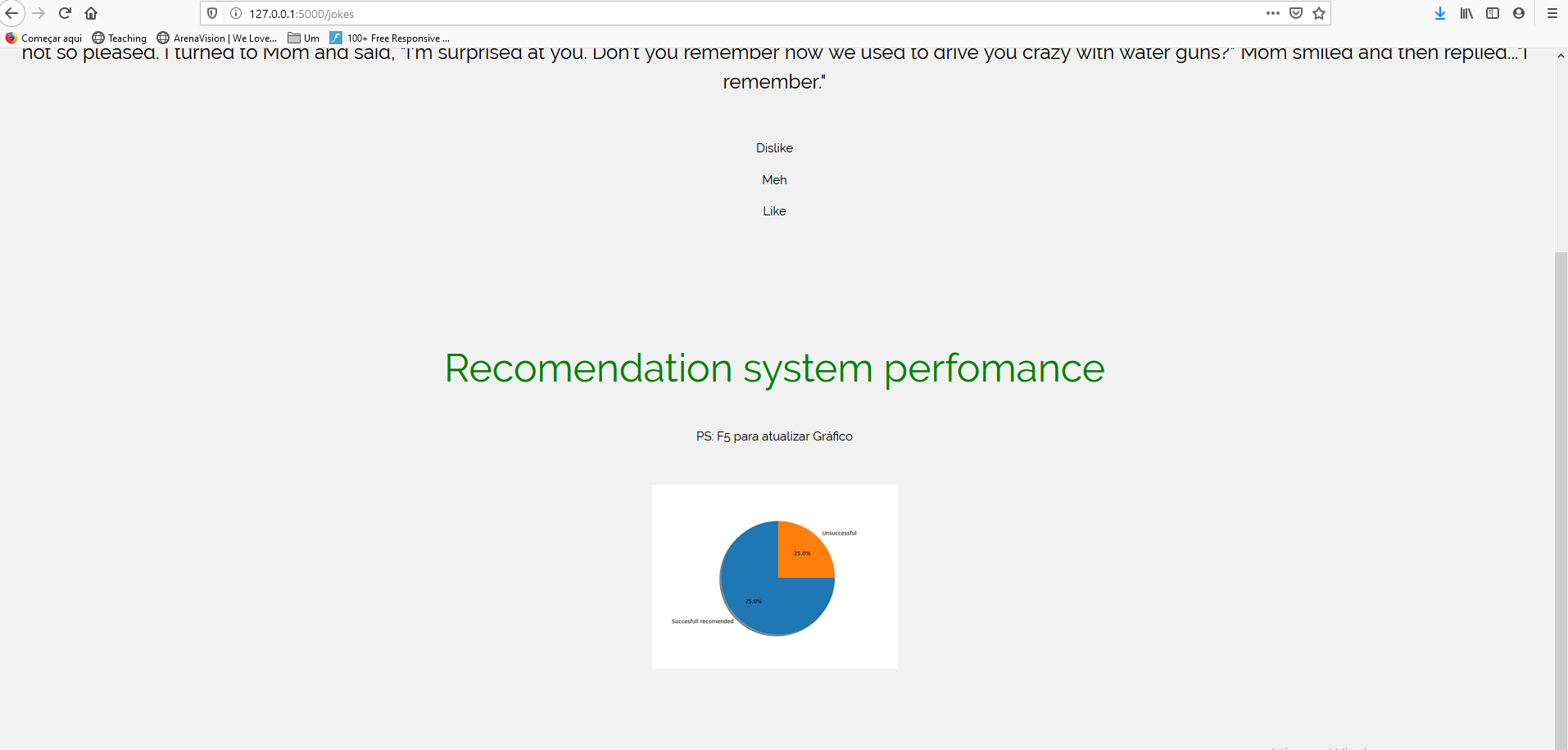
Tendo encontrado o utilizador mais próximo o sistema escolhe aleatoriamente uma das piadas com rating positivo avaliadas pelo utilizador encontrado e recomenda-a. Esta é avaliada e o processo repete-se



*Fig.15 Filtragem Colaborativa*

* + 1. Feedback Sistema

De modo a dar feedback das recomendações dadas ao utilizador está incluído no website uma secção que mostra um gráfico com a taxa de sucesso de recomendações. Deste modo tornamos a experiência mais dinâmica e interessante para o utilizador e estes têm noção da qualidade das recomendações que recebem, figura 16.



*Fig16. System Perfomance*

* 1. Futuro do sistema

A cada utilização por um novo utilizador, este é acrescentado á base de dados. Isto permite que o sistema se torne continuamente mais robusto e adaptável a um maior número de utilizadores, no entanto, no futuro seria necessário otimizar a filtragem colaborativa pois torna-se inviável comparar o utilizador atual com o crescente número de registos na base de dados.

1. **Tutorial de utilização**

4.1. Linha de Comandos

1º. Instalar todas as dependências

* Pip install numpy
* Pip install matplotlib
* Pip install os
* Pip install scipy
* Pip install pandas
* Pip install random
* pip install Flask

2º. Adicionar o projeto ao Flask

C:\path\to\app>set FLASK\_APP=KNNProjectV-1.py

3º. Correr a aplicação

C:\path\to\app>python -m flask run

4º-Abrir a aplicação

Depois de correr a aplicação temos de abrir o browser e colocar o seguinte endereço:

* <http://127.0.0.1:5000/>

1. **Conclusão**

Em suma, neste trabalho o grupo percebeu os fundamentos para a criação de um sistema de recomendação e as dificuldades inerentes para a concretização do mesmo. Assim como perceber a importância de contornar o “cold start”.

Descobrir e moldar um dataset existente de modo a permitir aplicar um sistema de recomendação no mesmo provou-se um dos maiores desafios para a concretização deste trabalho. Aplicar várias técnicas de recomendação em conjunto foi também um desafio.

No futuro seria importante testar outros algoritmos para a filtragem colaborativa e aumentar a performance dos mesmos.