

Escola de Engenharia  
**Universidade do Minho**

---

**Trabalho Prático de Grupo Nº2**

---

Realizado por:

Benjamim Oliveira PG42815  
Gonçalo Almeida A84610  
Nuno Pereira PG42846  
Rui Reis A84930

No âmbito do Perfil  
Machine Learning : Fundamentos e Aplicações  
dos cursos:  
MEIE/MEI/MMC

Unidade Curricular:  
SBS (Sistemas Baseados em Similaridade)

## Conteúdo

<b>1</b>	<b>Introdução &amp; Contextualização</b>	<b>2</b>
<b>2</b>	<b>Dados Utilizados</b>	<b>2</b>
<b>3</b>	<b>Técnicas</b>	<b>2</b>
3.1	Filtragem Colaborativa . . . . .	2
3.1.1	Top-N Não Personalizado . . . . .	3
3.1.2	Memory Based : User-based Nearest-Neighbour . . . . .	4
3.1.3	Workflow . . . . .	5
3.2	Filtragem baseada em Conteúdo . . . . .	6
3.2.1	Baseada em Modelo . . . . .	6
3.2.2	Simple Keyword Representation . . . . .	7
3.2.3	Term Frequency - Inverse Document Frequency . . . . .	8
3.3	Recomendações Baseadas em Conhecimento . . . . .	9
3.3.1	Baseado em Restrições . . . . .	9
<b>4</b>	<b>Interface Gráfica</b>	<b>10</b>
4.1	Web Scraping . . . . .	10
4.2	Web View . . . . .	10
<b>5</b>	<b>Conclusão</b>	<b>13</b>

## 1 Introdução & Contextualização

Modelar os gostos de determinados grupos de pessoas tem um impacto económico notável, tal modelação permite não só identificar os gostos atuais do utilizador bem como gostos futuros ou outras estatísticas determinantes. Por exemplo, a Netflix, uma das maiores plataformas de *streaming*, possui como grande vantagem um ótimo sistema de modelação de gostos. Desde a sua criação, a Netflix consegue alcançar público com interesses muito dispersos. Isto deve-se ao facto de eles conseguirem, com um elevado grau de certeza, modelar os gostos do seu público, e, dessa forma, perceber que tipo de séries podem trazer mais vantagens no mercado atual.

Como isso em mente, o presente trabalho prático visa aplicar essa mesma lógica a uma plataforma de partilha de receitas de cozinha, a Food. Partindo de dados sobre as receitas e avaliações dos vários utilizadores, pretendemos desenvolver um sistema capaz de aplicar um amplo leque de metodologias para o desenvolvimento de sistemas de recomendação. Nomeadamente, são empreendidas metodologias do tipo: filtragem colaborativa, filtragem baseada no conteúdo e no conhecimento.

Neste documento, começamos por fazer uma análise geral aos dados considerados. Passando a uma apresentação de cada uma das metodologias e sub-métodos de aplicação no contexto. Por fim, apresentamos como foi efetuada a avaliação do nosso modelo e como as respectivas recomendações são apresentadas ao utilizador. A metodologia utilizada consiste numa análise aprofundada dos métodos em questão, uma vista geral sobre como estes foram aplicados em KNIME que culmina, finalmente, na apresentação dos resultados obtidos e outros dados importantes.

## 2 Dados Utilizados

O dados utilizados são provenientes do website Food<sup>1</sup> e contemplam mais de 18 anos de atividade, agregando um total de mais de 180 mil receitas e 700 mil revisões das mesmas. O data set original pode ser encontrado através da plataforma Kaggle<sup>2</sup>.obre o conjunto de dados fornecidos, damos especial atenção aos seguintes ficheiros:

- RAW\_recipes.csv
- RAW\_interactions.csv

Que encapsulam os dados das receitas e interações entre utilizadores e receitas, respectivamente. Através destes dados, conseguimos ter uma noção do conteúdo objetivo de cada receita, bem como a avaliação subjetiva de cada utilizador. Os restantes ficheiros são também utilizados para validar os modelos desenvolvidos.

## 3 Técnicas

### 3.1 Filtragem Colaborativa

Filtragem colaborativa consiste, de forma superficial, em utilizar os gostos dos utilizadores, baseado na semelhança entre estes, de forma a recomendar novos itens. No entanto, esta metodologia requer a existência de uma comunidade e oferecem uma explicabilidade reduzida. Nesta metodologia, aplicamos duas técnicas distintas: Top-N não personalizado e vizinhos mais próximos.

<sup>1</sup><https://www.food.com/>

<sup>2</sup><https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions>

### 3.1.1 Top-N Não Personalizado

Rankings não personalizados destinam-se a abrangir os itens, neste caso receitas, mais relevantes para toda a comunidade. Na prática, pretendemos obter o Top-N de receitas com melhores pontuações de acordo com um determinado critério de atuação. No nosso caso, decidimos utilizar 4 modelos distintos, apresentados abaixo, de forma a encapsular comportamentos de interesse. Nos resultados desta técnica, os resultados são ordenados pela ordem decrescente da classificação médias, desde que a receita tenha sido classificada mais de 30 vezes.

- **Geral:** O modelo geral visa indicar as receitas mais bem classificadas de toda a comunidade. Como tal, é possível obtermos os resultados da figura 1.

Row ID	S name	I id	I minutes	I contrib...	S submitted	S tags
Row500_Row...	yeast biscuits	2677	132	163272	1999-08-19	['time-to-make', 'co...
Row9415_Ro...	berry cream cheese coffee cake	24768	85	2586	2002-04-08	['weeknight', 'time...
Row11368_R...	no bake hershey's bar pie	29084	20	37305	2002-05-21	['30-minutes-or-less...
Row64874_R...	perfectly chocolate hershey's hot cocoa	153877	4	232669	2006-01-30	['15-minutes-or-less...
Row69595_R...	kittchen's caesar tortellini salad	166669	5	89831	2006-05-01	['15-minutes-or-less...
Row46057_R...	substitution for pumpkin pie spice	107059	5	96436	2004-12-28	['15-minutes-or-less...
Row23105_R...	caprese salad tomatoes italian marinated to...	55309	10	63098	2003-03-01	['15-minutes-or-less...
Row76927_R...	the best creole cajun seasoning mix	186029	5	89831	2006-09-13	['15-minutes-or-less...
Row2100_Ro...	mozzarella tomato and basil salad	8507	5	3288	2000-11-11	['15-minutes-or-less...
Row26124_R...	mom's caramel rolls	61932	140	62191	2003-05-09	['weeknight', 'time...

Figura 1: Resultados para o TOP-10 Geral.

- **Por Tag:** Aplicamos um filtro ao modelo geral, de forma a que este só considere resultados que contenham nas suas tags uma tag específica. Na figura 2 podemos observar um exemplo deste resultado.

Row ID	I recipe_id	D Mean(r...)	S name	I minutes	I contrib...	S submitted	S tags
Row500_Row...	2677		yeast biscuits	132	163272	1999-08-19	['time-to-make', 'course', 'preparation', 'for-a...
Row9415_Ro...	24768	5	berry crese...	85	2586	2002-04-08	['weeknight', 'time-to-make', 'course', 'prepar...
Row11368_R...	29084	5	no bake her...	20	37305	2002-05-21	['30-minutes-or-less...', 'time-to-make', 'course...
Row64874_R...	153877	5	perfectly ch...	4	232669	2006-01-30	['15-minutes-or-less...', 'time-to-make', 'course...
Row69595_R...	166669	5	kittchen's c...	5	89831	2006-05-01	['15-minutes-or-less...', 'time-to-make', 'course...
Row46057_R...	107059	5	substitution ...	5	96436	2004-12-28	['15-minutes-or-less...', 'time-to-make', 'course...
Row23105_R...	55309	5	caprese sala...	10	63098	2003-03-01	['15-minutes-or-less...', 'time-to-make', 'course...
Row76927_R...	186029	5	the best cre...	5	89831	2006-09-13	['15-minutes-or-less...', 'time-to-make', 'course...
Row2100_Ro...	8507	4.974	mozzarella t...	5	3288	2000-11-11	['15-minutes-or-less...', 'time-to-make', 'course...
Row26124_R...	61932	4.974	mom's cara...	140	62191	2003-05-09	['weeknight', 'time-to-make', 'course', 'prepar...

Figura 2: Resultados para o TOP-10 por Tag.

- **Por Tipo Nutritivo:** Na prática, os utilizadores de uma determinada plataforma estão interessados em obter receitas que superem um determinado critério nutritivo, o que acontece neste modelo. Por exemplo, na figura 3 podemos observar um caso em que é retornado ao utilizador as receitas mais bem classificadas e com mais proteína. No entanto, isto podia facilmente ser expandido para captar as restantes características das receitas.

The screenshot shows a data analysis interface with a top navigation bar (File, Edit, Help, Navigation, View) and a title "Filtered - 3:84:83 - Rule-based Row Filter (get only 10 first)". Below the title is a table titled "Table 'default' - Rows: 10 Spec - Columns: 13 Properties Flow Variables". The table contains 10 rows of data with columns: Row ID, recipe\_id, Mean(r...), name, minutes, contrib..., submitted, and tags. The data includes various recipes like "pernil puerto rican pork shoulder", "seasoned goldfish crackers", etc. A "String Widget" dialog is open below the table, with the placeholder "Indicate nutritional type" and a text input field containing "protein". At the bottom right of the dialog are "Reset", "Apply", and "Close" buttons.

Figura 3: Resultados para o TOP-10 por tipo nutritivo.

- **Por Ingrediente:** Aplicamos um filtro ao modelo geral, de forma a que este só considere resultados que contenham nas sua lista de ingredientes um ingrediente específico. Na figura 4 podemos observar um exemplo deste resultado.

The screenshot shows a data analysis interface with a top navigation bar (File, Edit, Help, Navigation, View) and a title "Appended table - 3:129:125 - String Manipulation (restore trailing)". Below the title is a table titled "Table 'default' - Rows: 10 Spec - Columns: 13 Properties Flow Variables". The table contains 10 rows of data with columns: Row ID, recipe\_id, Mean(r...), name, minutes, contrib..., submitted, and tags. The data includes various recipes like "greek yoghurt and fruit salad", "flank steak with lime choped...", etc. A "String Widget" dialog is open below the table, with the placeholder "Indicate ingredient name" and a text input field containing "honey". At the bottom right of the dialog are "Reset", "Apply", and "Close" buttons.

Figura 4: Resultados para o TOP-10 por Ingrediente.

### 3.1.2 Memory Based : User-based Nearest-Neighbour

Esta técnica permite comparar um determinado utilizador com o seu grupo de vizinhos. Define-se vizinhos como sendo todos os utilizadores com os quais o cliente possui receitas avaliadas em comum. Sendo  $P$  o conjunto de receitas avaliadas por ambos os utilizadores  $a$  e  $b$ , e  $(\bar{r}_a, \bar{r}_b)$  as respectivas avaliações médias desses utilizadores. Com isto em mente, podemos calcular o coeficiente de correlação de Pearson, ou semelhança, da seguinte forma:

$$\text{sim}(a,b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a) \cdot (r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \cdot \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}} \quad (1)$$

Com este conceito de semelhança, podemos prever a classificação que o cliente  $a$  daria à receita  $y$ , que

ainda não avaliou anteriormente. Baseando na semelhança ao conjunto  $N$  de vizinhos que avaliaram o item  $y$ . Este mecanismos pode ser caracterizado da seguinte forma:

$$\text{pred}(a,y) = \bar{r}_a + \frac{\sum_{b \in N} \text{sim}(a,b) \cdot (r_{b,y} - \bar{r}_b)}{\sum_{b \in N} \text{sim}(a,b)} \quad (2)$$

Com isto, e dado um cliente e item de interesse, somos capazes de indicar o interesse daquele cliente no item. Assim, para abrangir um maior conjunto de itens, podemos seleccionar os vizinhos mais próximos, através de *clustering* e prever a afinididade do cliente aos itens melhor classificados do mesmo *cluster*.

### 3.1.3 Workflow

Para a aplicação desta metodologia foram desenvolvidos os *workflows* das figuras 5 e 6, que representam as diferentes técnicas aplicadas.

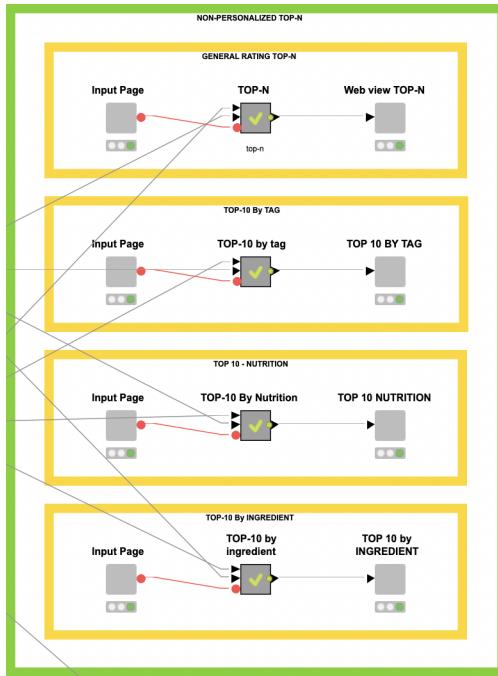


Figura 5: Workflow para a técnica de Top-N não personalizado.

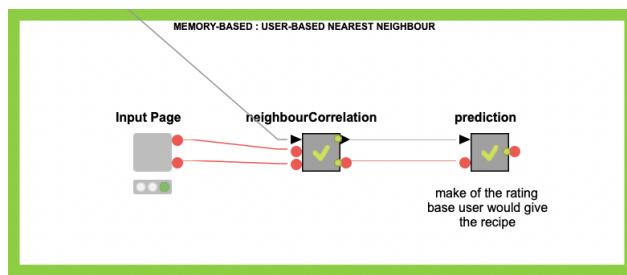


Figura 6: Workflow para a técnica de *Memory Based - User Based*.

### 3.2 Filtragem baseada em Conteúdo

Ao contrário da filtragem colaborativa, a filtragem *Content-Based*, como o nome indica, corresponde a olhar para documentos semelhantes e termos de conteúdo e recomendar de acordo. Se o utilizador  $A$  comprou um determinado item  $X$  e existe um item  $Y$  que esse mesmo utilizador ainda não comprou, então  $Y$  será recomendado se for semelhante a  $X$ .

Porém, a semelhança entre dois determinados itens é subjetiva. Como tal, diferentes técnicas abordam o contexto de semelhança entre itens de forma distinta. Neste projeto, são abordados as técnicas de: filtragem baseada em modelo, representação de palavras-chave simples e TF-IDF.

#### 3.2.1 Baseada em Modelo

Cada receita tem a si associada um conjuntos de *tags* que identifica as principais características da receita. No contexto deste estudo, abordamos a filtragem baseada em modelo com a técnica de *one-hot encoding* das *tags* de cada receita.

No entanto, depois de análise de dados, foi possível concluir que o número de *tags* distintas existentes é na ordem das centenas, o que nitidamente torna o método de *one-hot encoding* complexo. De forma a ultrapassar este problema, decidimos utilizar apenas as 20 *tags* mais comuns no universo de receitas, o que simplifica drasticamente a nossa análise.

De seguida, utilizando *k-Means*, as receitas são repartidas em  $N$  clusters. Ao considerar clusters em vez de toda a base de dados, reduzimos significativamente a complexidade do nosso sistema. Sendo que agora será apenas necessário comparar com os itens mais semelhantes, ou seja, do mesmo cluster. Com isto, podemos produzir algoritmos mais complexos que possuem em consideração diferentes prespetivas, desde que no mesmo cluster. Nomeadamente, aplicamos o seguinte tipo de recomendações:

- Top-10 Receitas do mesmo cluster que o utilizador.
- Top-5 Receitas de acordo com as *tags* mais consumidas pelo utilizador.

Ambas as recomendações funcionam de forma semelhante e utilizam as preferências do utilizador de forma a filtrar o conteúdo apresentado.

Na figura 12 podemos ver apresentado o primeiro destes temas, a recomendação de top-10 do mesmo cluster que o utilizador.

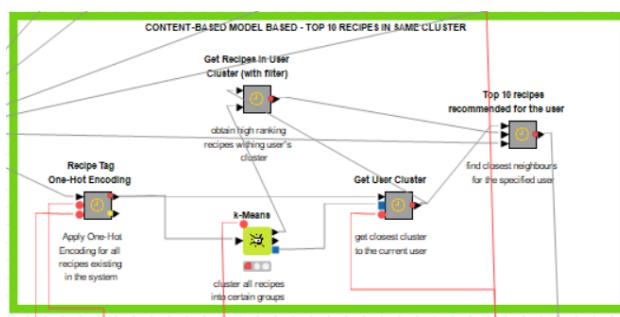


Figura 7: Workflow para a recomendação de Top-10 do mesmo cluster.

Primeiramente, fazemos *one-hot encoding* para todas as receitas da nossa base de conhecimento, treinamos um modelo de clustering com estes dados e, de seguida, obtemos as tags mais comuns dos itens mais bem classificados pelo utilizador em questão. Com estes dados, conseguimos obter um *cluster* ao utilizador, que corresponde ao conjunto de receitas que se encontram mais próximas dos gostos daquele utilizador.

Por fim, dentro do cluster correspondente do utilizador, encontramos as receitas mais próximas e com melhor classificação. No final, são essas que são apresentadas ao utilizador.

Na figura 8 podemos ver aplicada uma tática semelhante. Porém, no caso do top-5 de receitas de acordo com as 3 tags mais consumidas pelo utilizador, as receitas do mesmo cluster apresentadas são filtradas de forma a refletir apenas receitas que, de facto, possuam aquelas tags.

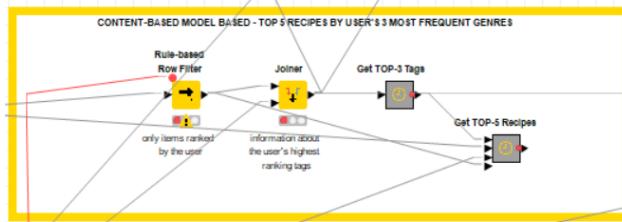


Figura 8: Workflow para a recomendação de Top-5 de acordo com tags do utilizador.

### 3.2.2 Simple Keyword Representation

Na falta de existência de tags, a técnica de *simple keyword representation* pode ser uma boa opção pois baseia-se na descrição da própria receita.

Dado um conjunto de receitas, a descrição de cada uma das receitas é convertida para o formato de lista. Com esta lista, e de acordo com os itens classificados pelo utilizador, conseguimos determinar o perfil de utilizador, ou seja, as palavras-chave que mais surgem nas receitas que o interveniente classifica com uma boa classificação.

De seguida, e utilizando a lista de palavras-chave de todos os documentos que o utilizador não classificou, conseguimos utilizar o coeficiente de *Dice* de forma a aferir quais as receitas que mais se aproximam do conjunto de palavras-chave preferidas pelo utilizador.

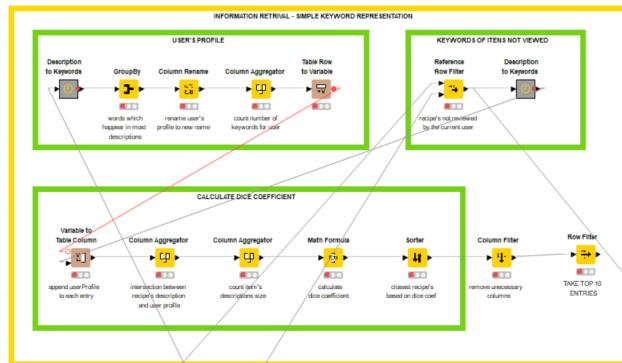


Figura 9: Workflow para a aplicação da técnica de *simple keyword representation*.

Na figura 9 podemos observar a aplicação desta Técnica em Knime. Primeiramente, é traçado o perfil de utilizador, bem como obtida a lista de keywords para todas as receitas que o utilizador ainda não classificou. De seguida, calculamos o coeficiente de *Dice* de forma a obter as receitas que mais se aproximam do perfil traçado para aquele utilizador.

### 3.2.3 Term Frequency - Inverse Document Frequency

A técnica de *simple keyword representation* possui graves deficiências principalmente por ter em consideração o número de vezes que uma dada palavra-chave ocorre no documento. Isto é um comportamento de extremo interesse pois, no fundo, estamos interessados em valorizar palavras-chave que surgem de forma mais densa em todos os documentos.

Como tal, a técnica de *Term Frequency - Inverse Document Frequency* (TF-IDF) apresenta-se como uma boa alternativa. Por um lado, a componente de *Term Frequency* (TF) mede a densidade de um dado termo ao longo do documento, palavras mais densas são consideradas de maior interesse. Por outro lado, a parte de *Inverse Document Frequency* (IDF) permite reduzir o peso de palavras-chave que surgem em todos os documentos, enaltecedo palavras-chave mais raras.

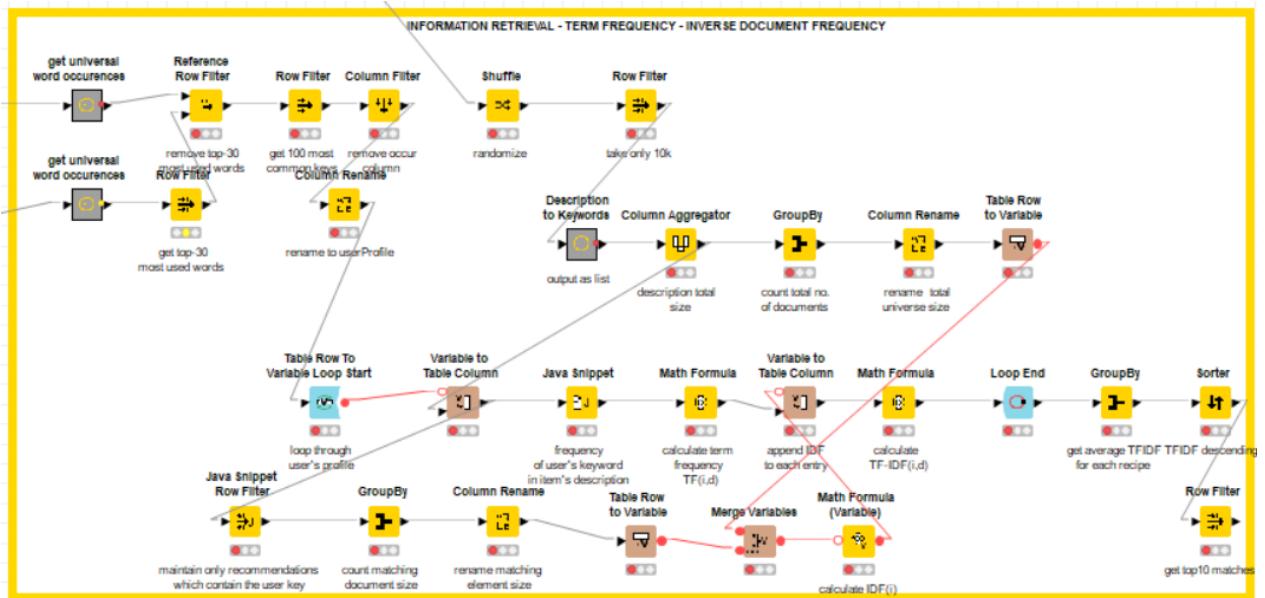


Figura 10: Workflow para a aplicação da técnica de *term-frequency - inverse document frequency*.

Na figura 10 podemos observar a nossa implementação desta técnica. Inicialmente é calculada a IDF de todas as palavras-chave do perfil do utilizador. De seguida, para cada receita de interesse (as que o utilizador ainda não classificou), classificamos a frequência de cada palavra-chave do perfil do utilizador na descrição das receitas.

Posto isto, conseguimos obter o coeficiente de TF-IDF para todas as palavras-chave e documentos de interesse do utilizador. De seguida, recomendamos ao utilizador as receitas que possuem uma TF-IDF média mais elevada, que evidencia uma importância de todas as palavras-chave, de forma geral, na receita.

### 3.3 Recomendações Baseadas em Conhecimento

#### 3.3.1 Baseado em Restrições

Recomendações baseadas em conhecimento são utilizadas muitas vezes para contornar *cold starts*. Esta técnica é também aplicada com frequência em casos onde não se conseguem registar dados do utilizador com frequência suficiente, como é por exemplo o caso do mercado mobiliário.

A base deste modelo passa por converter *constraints* de alto nível em *queries* que possam ser aplicadas aos dados que nos servem de base.

One word that describes what you're looking for!

Pick some ingredients you like!

salt  butter  sugar  onion  water  eggs  olive oil  flour  garlic cloves  milk  pepper  
 brown sugar  garlic  all-purpose flour  baking powder  egg  salt and pepper  parmesan cheese  lemon juice  
 baking soda  vegetable oil  black pepper  vanilla  cinnamon  tomatoes  sour cream  garlic powder  
 vanilla extract  honey  onions  oil  garlic clove  cream cheese  celery  cheddar cheese  unsalted butter  
 soy sauce  mayonnaise  chicken broth  paprika  worcestershire sauce

How are you feeling?

Are you in a rush?

Yes  No  Oh, not at all...

Figura 11: Webview para Recomendações Baseadas em Conhecimento.

Na figura acima pode ser observada a interface criada para recolher estas *constraints* de alto nível, como por exemplo, como a pessoa se está a sentir (How are you feeling?) ou se a pessoa está com pressa ou não (Are you in a rush?). São recolhidos ainda outros dados utilizados como auxiliares na escolha das receitas a sugerir, sendo estes dados uma *keyword* e alguns ingredientes da preferência do utilizador.

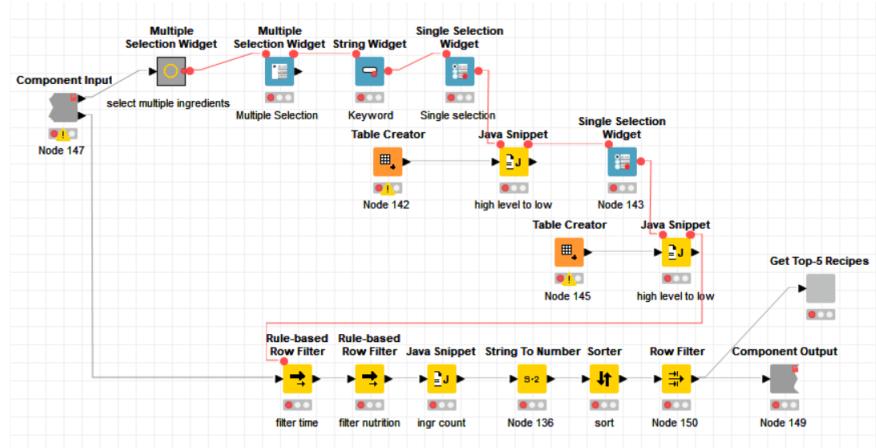


Figura 12: Workflow para a recomendação de Top-10 do mesmo cluster.

Estas *constraints* são trabalhadas, após serem recolhidas, para que a partir delas se consigam gerar

*queries*, como previamente foi referido.

Como exemplo, a partir da pergunta "Are you in a rush?" será extraída uma *querie* relativa à duração das receitas. Com a pergunta "How are you feeling?", que tem como respostas possíveis:

- "Healthy!"
- "Craving Calories..."

filtramos o nível calórico da refeição a ser sugerida. Utilizamos ainda os ingredientes para obter as receitas que possam ser mais apetecíveis ao utilizador.

## 4 Interface Gráfica

### 4.1 Web Scraping

Por forma a poder mostrar as imagens das receitas ao utilizador foi necessário obter o *url* de cada receita através dos respetivos valores dos atributos *name* e *id*. Após gerar o *url*, através do nodo *HTTP Retriever*, foram realizados pedidos para obter o código HTML da página de cada receita, código este filtrado da resposta do nodo com a aplicação de um outro nodo *HTML Parser*. Por fim, com um nodo *XPath*, foi encontrado o *url* e este associado a cada receita.

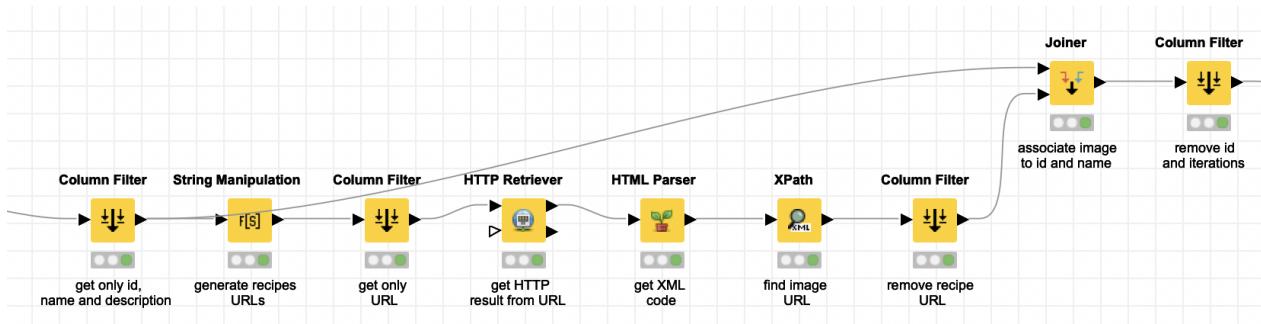


Figura 13: Workflow do web scraping

### 4.2 Web View

Para desenvolver as *web views* começamos por criar componentes de entrada e de saída. Os componentes de entrada recebem inputs, por exemplo tipos de nutrição para que depois os dados sejam tratados. Os nodos utilizados nesses componentes foram *string input*, *text output* e *component output* exemplo da figura 12, que pertence ao *Collaborative Base Recommender* do *top 10 nutrition*.

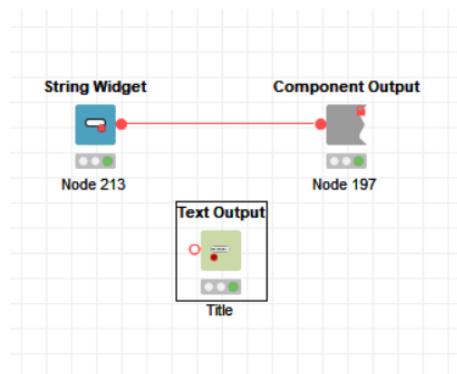


Figura 14: Input Components

Na figura 13 temos a representação da *web view* do input da *top 10 nutrition*. Para aplicar o valor do input temos de selecionar no canto inferior direito da janela da *web view* o *close and apply as new default* e executar os nodos da métrica em uso.

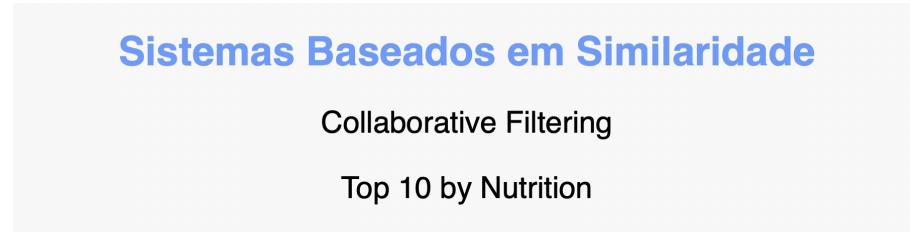


Figura 15: Input Web View

Depois dos dados serem tratados, são apresentados nos *component* de *output* numa vista web. Este *component* recebe uma tabela de id das receitas, descrições das receitas e nomes das receitas que são lidas e apresentadas no nodo *generic JavaScript view* em forma de uma lista de cartões. Os títulos das vistas são apresentados nos *text output*. Exemplo da figura 14, que pertence ao *Collaborative Base Recommender* do *top 10 nutrition*.

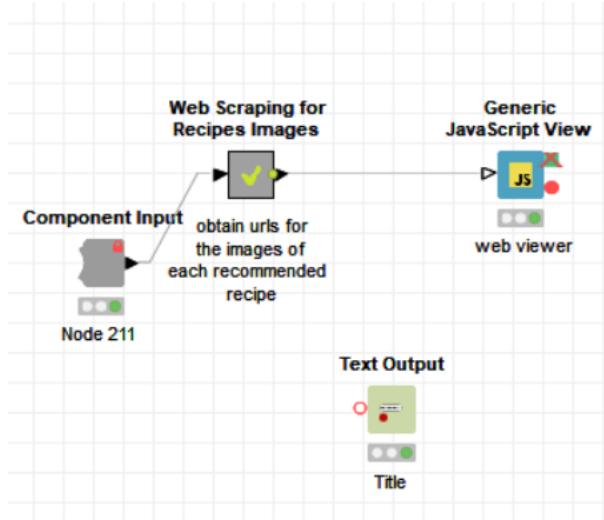


Figura 16: Output Components

Na figura 15 temos a representação *web view* do output da *top 10 nutrition*, em alguns casos não existe imagem da receita sendo apresentada uma imagem predefinida, para atualizar a *web view* temos que a fechar e voltar a abrir.



Figura 17: Output Web View

## 5 Conclusão

Com a realização deste trabalho prático o grupo aumentou e pôs em prática os conhecimentos adquiridos sobre sistemas de recomendações, sobretudo sobre as técnicas e os algoritmos que utilizam. Foram aplicados diferentes paradigmas, cada um com diferentes estratégias, tornando-se um sistema de recomendações híbrido.

Há que salientar a importância da ferramenta KNIME no desenvolvimento do projeto pois, sendo que como grupo já se encontrava familiarizado devido à sua constante utilização durante o semestre, facilitou a elaboração do sistema.

Também deve-se referir o *dataset* utilizado, pois um bom sistema de recomendações depende de um bom conjunto de dados e o conjunto escolhido demonstrou-se relevante e rico em informação para este tipo de trabalho. Ainda, a existência de um *dataset* com interações dos utilizadores ajudou a contornar o problema do *cold start*.

Contudo, apesar do grupo considerar o sistema desenvolvido satisfatório, não se encontra completo nem ótimo. A execução dos seus processos é bastante demorada, tendo em conta que são utilizadas técnicas bastante complexas. Apesar de ter havido intenções de aplicar a abordagem *Constraint-based* do paradigma *Knowledge-based*, os fatores tempo e carga de trabalho não o permitiram.

De uma forma geral, considera-se que foram atingidas as expectativas, conseguindo-se ultrapassar vários obstáculos que foram surgindo ao longo do projeto.