

**Mestrado em Engenharia Informática**

**Perfil de Machine Learning: Fundamentos e Aplicações**

**Sistemas Baseados em Similaridade**

Trabalho Prático de Grupo – 2ª Parte

4º Ano, 1º Semestre

Ano letivo 2020/2021

Diogo Alexandre Rodrigues Lopes

PG42823

Fábio Gonçalves

PG42827

Joel Costa Carvalho

PG42837

Conteúdo

[**Introdução** 3](#_Toc61364733)

[**Tarefa 1.** Análise, Tratamento e Exploração de dados do *dataset* (*2019 Iowa Liquor Sales*) 4](#_Toc61364734)

[**a.** Sobre o *dataset* selecionado 4](#_Toc61364735)

[**b.** Carregar, no *Knime*, o *dataset* selecionado 7](#_Toc61364736)

[**c.** Aplicar de nodos, de modo a fazer Tratamento de Dados 10](#_Toc61364737)

[**d.** Aplicar de nodos, de modo a fazer Análise de Dados 12](#_Toc61364738)

[***1.*** Metanode *Data Listing Information* 14](#_Toc61364739)

[**2.** Metanode *Data Sales Information* 16](#_Toc61364740)

[**3.** Metanode *Data TOP’s* 18](#_Toc61364741)

[**e.** Aplicar de nodos, de modo a fazer a Otimização do Modelo 22](#_Toc61364742)

[**4.** Solução de Otimização Ótima do Modelo 22](#_Toc61364743)

[**5.** Evidência da Implementação de outras Soluções de Otimização do Modelo 25](#_Toc61364744)

[**Tarefa 2.** Conceção e Implementação de um Sistema de Recomendação 27](#_Toc61364745)

[**a.** Aplicar de nodos, baseados em Regras de Associação no Modelo 27](#_Toc61364746)

[**1.** Regras de Associação Implementadas baseadas na *feature* ‘*Product Name’* 27](#_Toc61364747)

[**2.** Regras de Associação Implementadas baseadas na *feature* ‘*Category Name’* 33](#_Toc61364748)

[**b.** Aplicar de nodos, baseados em *Clusters* no Modelo 36](#_Toc61364749)

[**1.** *Clusters* baseados na *feature* ‘*Category Name’* 39](#_Toc61364750)

[**2.** *Clusters* baseados na *feature* ‘*Product Name’* 40](#_Toc61364751)

[**3.** *Clusters* baseados na *feature* ‘*Price’* 41](#_Toc61364752)

[**4.** *Clusters* baseados na *feature* ‘*Month (name)’* 42](#_Toc61364753)

[**5.** *Clusters* baseados na *feature* ‘*City’* 43](#_Toc61364754)

[**c.** Avaliação do Sistema de Recomendação 44](#_Toc61364755)

[a. Regras de Associação 44](#_Toc61364756)

[**Conclusão** 46](#_Toc61364757)

# **Introdução**

Para a elaboração desta 2ª Parte do Trabalho Prático de Grupo, que consistia na conceção e implementação de um Sistema de Recomendação, tendo o mesmo como principal objetivo indicar ao utilizador, da maneira mais precisa e robusta, os diversos produtos que se encontram de acordo com as suas preferências. Para a implementação deste sistema, escolhemos um *dataset* da plataforma *Iowa Data (Internal)* ([https://mydata.iowa.gov](https://mydata.iowa.gov/login)), mais concretamente, o *dataset* ***2019 Iowa Liquor Sales***. Com o intuito de tornar de mais fácil perceção todo o trabalho elaborado, resolvemos dividir a implementação do mesmo em duas partes. Inicialmente será feita uma análise, tratamento e exploração de dados do *dataset* selecionado. Depois disto, numa segunda fase, será elaborado o sistema de recomendação.

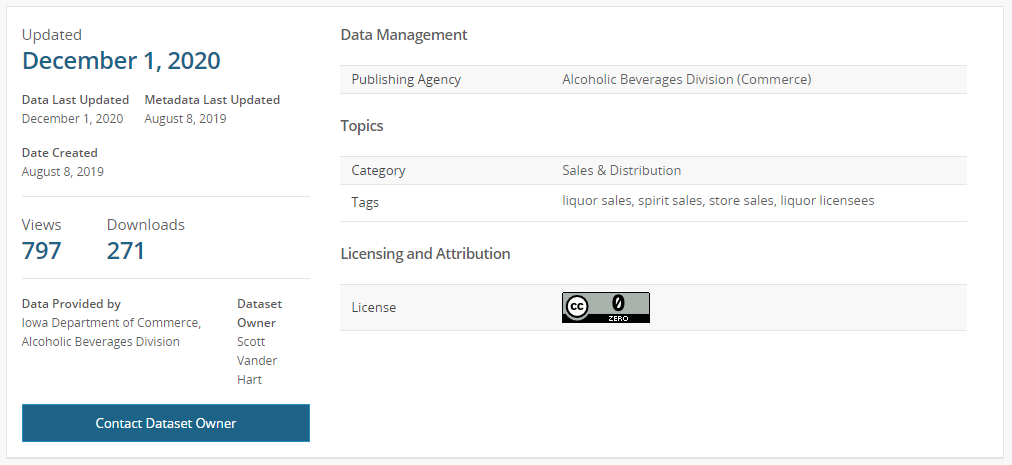
A primeira parte do trabalho, resumiu-se à estruturação e “limpeza” do *dataset*, desde remoção de colunas, *casting* e eliminação de valores nulos. Após o *dataset* estar devidamente estruturado, algumas análises e cálculos foram realizados sobre o mesmo. Relativamente à última fase e a mais significativa do projeto, o Sistema de Recomendação, foi dividido em dois tipos, sendo eles baseados em *Clusters* e em *Regras* *Associativas*. De salientar que este relatório contém todas estas etapas descritas, assim como os aspetos importantes relacionados com todo este processo. Os principais objetivos com a implementação/resolução deste trabalho são:

1. Consultar, Analisar e Selecionar um *dataset* sobre os quais seja possível desenvolver um Sistema de Recomendação;
2. Utilizar a plataforma *KNIME* para desenvolver um, ou vários, *workflows* para Exploração, Análise e Tratamento dos Dados assim como para Extração de Informação dos mesmos;
3. Desenvolver um Sistema de Recomendação seguindo uma abordagem híbrida implementando paradigmas como top-N, filtragem colaborativa, baseada em conteúdo, baseada em conhecimento, entre outros;
4. Implementar Métodos para Controlo/Avaliação da Qualidade das Recomendações;
5. O sistema desenvolvido deverá ser capaz de receber e tratar novos *inputs* de um utilizador, devolvendo uma, ou mais, recomendações. Deverão também ser implementados métodos que permitam ao utilizador perceber o porquê da recomendação.

# **Tarefa 1.** Análise, Tratamento e Exploração de dados do *dataset* (*2019 Iowa Liquor Sales*)

## Sobre o *dataset* selecionado

O *dataset* selecionado, disponível em [https://t.ly/0nWe](https://mydata.iowa.gov/Sales-Distribution/2019-Iowa-Liquor-Sales/38x4-vs5h), contém uma visualização filtrada de dados com informações relativas à compra e venda de bebidas alcoólicas dos vendedores licenciados da Classe E por produto e data de compra para o ano civil de 2019. Esta licença de bebidas (Classe E), para supermercados, lojas de bebidas, lojas de conveniência, etc., permite a estes estabelecimentos comerciais a venda de bebidas para consumo externo em recipientes originais fechados, de modo a que no *dataset*, é possível observar por loja o volume de vendas de um determinado artigo/produto.



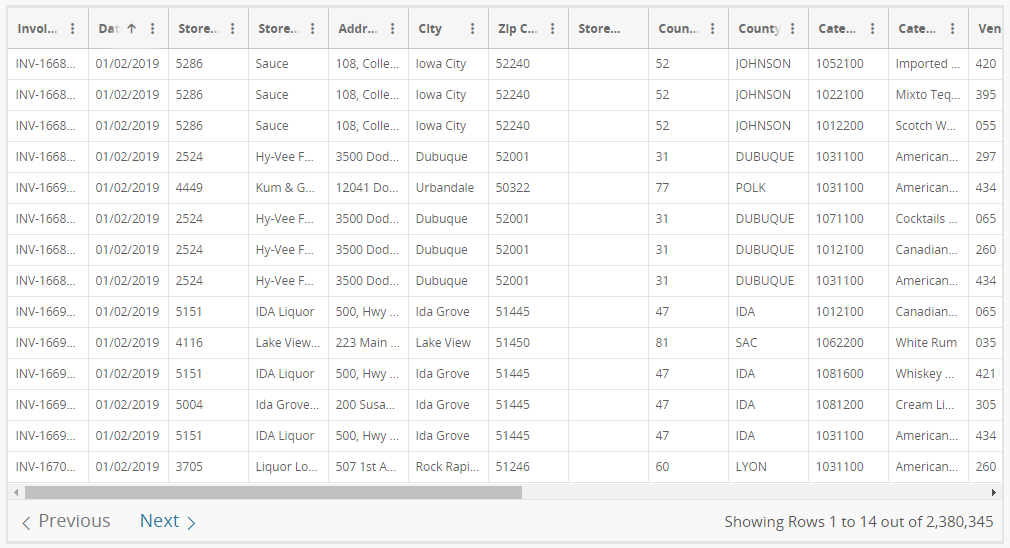
**Figura nº1 – Informação sobre o *dataset***

Tal como é possível observar na imagem acima e como já foi referido, este conjunto de dados é fornecido pelo ***Departamento de Comércio de Iowa*** e a ***Divisão de Bebidas Alcoólicas***. Os principais tópicos do mesmo são as vendas e distribuição de bebidas alcoólicas feitas por parte de estabelecimentos e vendedores devidamente licenciados. Salientamos também que o *dataset* foi criado em Agosto de 2019 e a última atualização feita foi em Dezembro de 2020.

Este *dataset* é composto por 24 colunas (*features*) com cerca de 2 380 000 registos, todos referentes ao ano de 2019.

|  |  |  |
| --- | --- | --- |
| Nome da Coluna | Descrição | Tipo de Dados |
| Invoice/Item Number | Fatura Concatenada e Número da Linha associado ao pedido. Isto fornece um identificador único para os produtos incluídos no pedido | Plain Text |
| Date | Data do Pedido | Date & Time |
| Store Number | Número único atribuído à Loja que efetuou o pedido do Licor. | Plain Text |
| Store Name | Nome da Loja que encomendou o Licor | Plain Text |
| Address | Morada da Loja que encomendou o Licor | Plain Text |
| City | Cidade onde a Loja está localizada | Plain Text |
| Zip Code | Código Postal da loja que encomendou o Licor | Plain Text |
| Store Location | Local da Loja que efetuou o pedido do Licor. O endereço, cidade, estado e código postal são *geocodificados* para fornecer coordenadas geográficas | Point |
| County Number | Número do Município de Iowa para o Município onde a Loja que pediu a bebida está localizada | Plain Text |
| County | Município onde está localizada a Loja que fez o pedido | Plain Text |
| Category | Código da Categoria associado à bebida pedida | Plain Text |
| Category Name | Categoria da bebida pedida. | Plain Text |
| Vendor Number | Número do Fornecedor da empresa para a marca de bebida alcoólica solicitada | Plain Text |
| Vendor Name | Nome do Fornecedor da empresa para a marca de Licor pedido | Plain Text |
| Item Number | Número do Item para o produto de Licor individual pedido. | Plain Text |
| Item Description | Descrição do Licor individual pedido. | Plain Text |
| Pack | Número de Garrafas numa caixa da bebida pedida | Number |
| Bottle Volume (ml) | Volume de cada Garrafa de bebida solicitada em mililitros. | Number |
| State Bottle Cost | Valor que a Divisão de Bebidas Alcoólicas pagou por cada garrafa de Bebida Alcoólica pedida | Number |
| State Bottle Retail | Valor que a Loja pagou para cada Garrafa de Bebida Alcoólica pedida | Number |
| Bottles Sold | Número de Garrafas de Bebidas Alcoólicas encomendadas pela Loja | Number |
| Sale(Dollars) | Custo total do pedido de bebidas (número de garrafas multiplicado pelo custo de garrafas em cada estado) | Number |
| Volume Sold (Liters) | Volume total de Licor pedido em litros. (ou seja, (Volume da garrafa (ml) x garrafas vendidas) / 1.000) | Number |
| Volume Sold (Gallons) | Volume total de Licor pedido em gallons. (ou seja, (Volume da garrafa (ml) x garrafas vendidas) /3785.411784) | Number |

**Tabela nº1 – Colunas do *dataset***

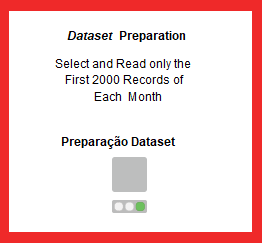


**Figura nº2 – Visão Geral do *dataset***

## Carregar, no *Knime*, o *dataset* selecionado

Tendo em conta que o *dataset* tem cerca de 2 380 000 registos, e uma vez que para o processamento de dados no *KNIME* esta quantidade de registos tornava todo o processamento bastante demoroso, optámos por fazer uma a preparação no conjunto de dados. Com o objetivo de reduzir a quantidade de dados e aumentar a performance do modelo, decidimos extrair do *dataset* 2 000 registos mensais referentes aos meses de Janeiro, Fevereiro, Março, Abril, Maio e Junho do *dataset* original e assim construir um *dataset* novo para ser utilizado na construção do nosso modelo com 12 000 registos.

Para este processo de construção do *dataset*, criámos um componente (***Preparação Dataset***) onde se foi implementado o respetivo *workflow* para extração dos 2 000 registos mensais pretendidos. De acordo com a Figura nº 4 é possível visualizar que para fazer a extração dos registos pretendidos de cada mês, recorremos ao nodo ***Excel Reader (XLS)***. De modo a construir o novo conjunto de dados, aplicamos também o nodo ***Excel Writter (XLS)*** para extrair os registos mensais intencionados. Posteriormente fizemos uma junção manual, destes dados num único ficheiro do tipo ***.xlsx***.

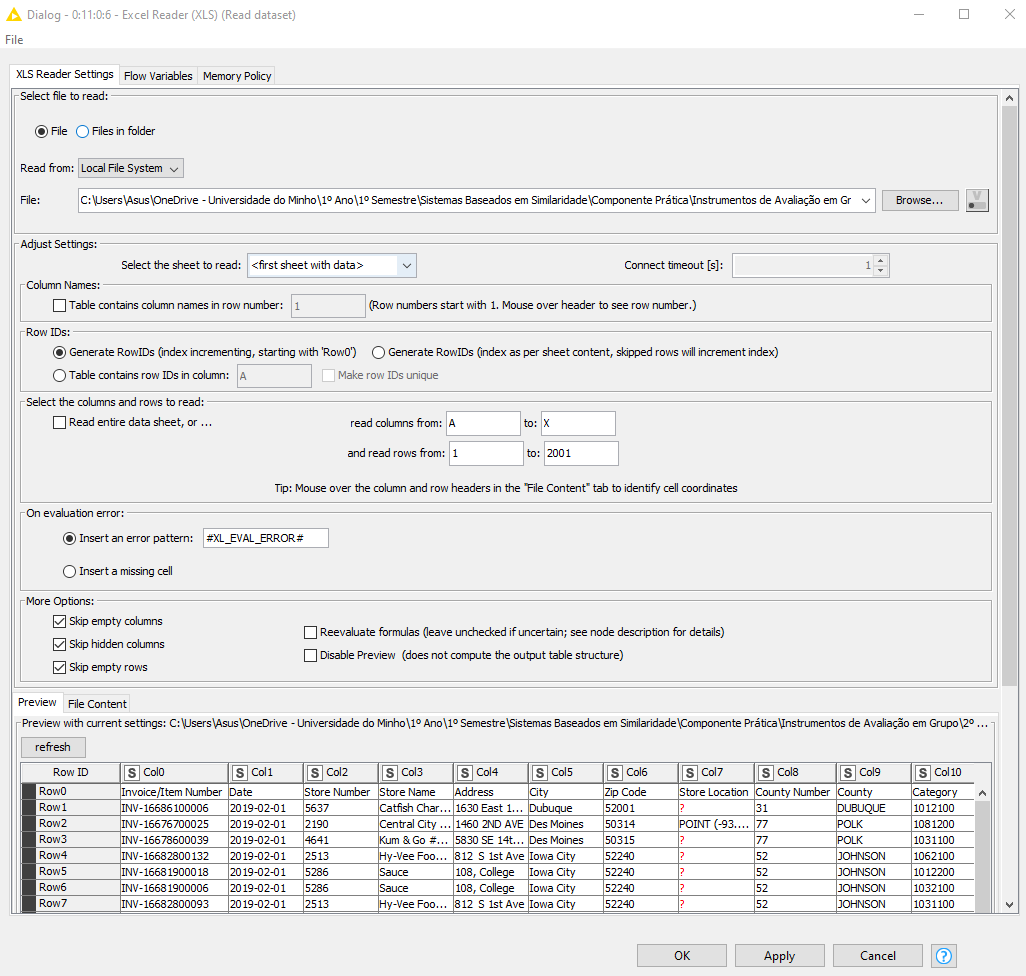


**Figura nº3 – Componente criado para a Preparação do *dataset***

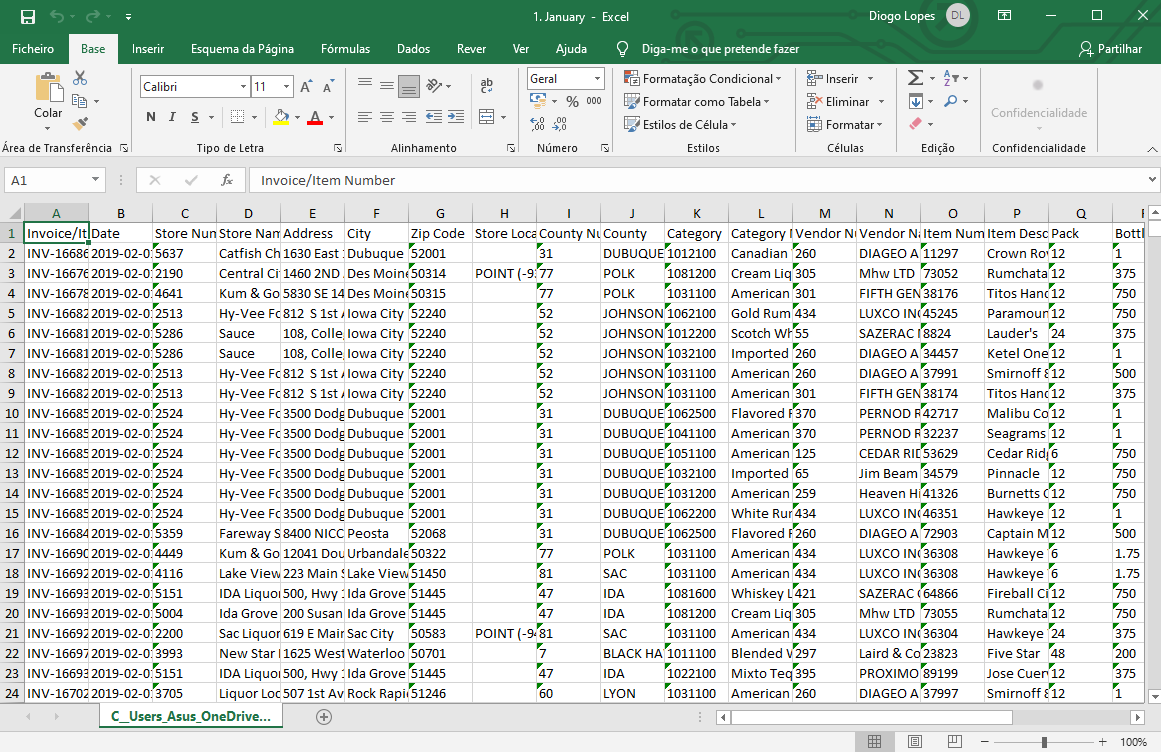


**Figura nº4 – *Workflow* implementado no Componente**

À semelhança da configuração anterior, na Figura nº 5 em todos os nodos de leitura do *dataset*, foram lidos os 2 000 primeiros registos referentes a um determinado mês presentes no conjunto de dados original. De salientar que a exceção dos outros meses, quando efetuamos a extração dos dados referentes ao mês de Janeiro tivemos de ler 2001 registos (ver o destacado na imagem), uma vez que na primeira linha do *dataset* estão presentes os nomes das colunas que identificam os dados.

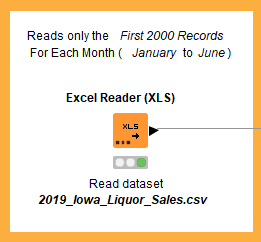


**Figura nº5 – Configuração do nodo *Excel Reader (XLS)* (Mês de Janeiro)**

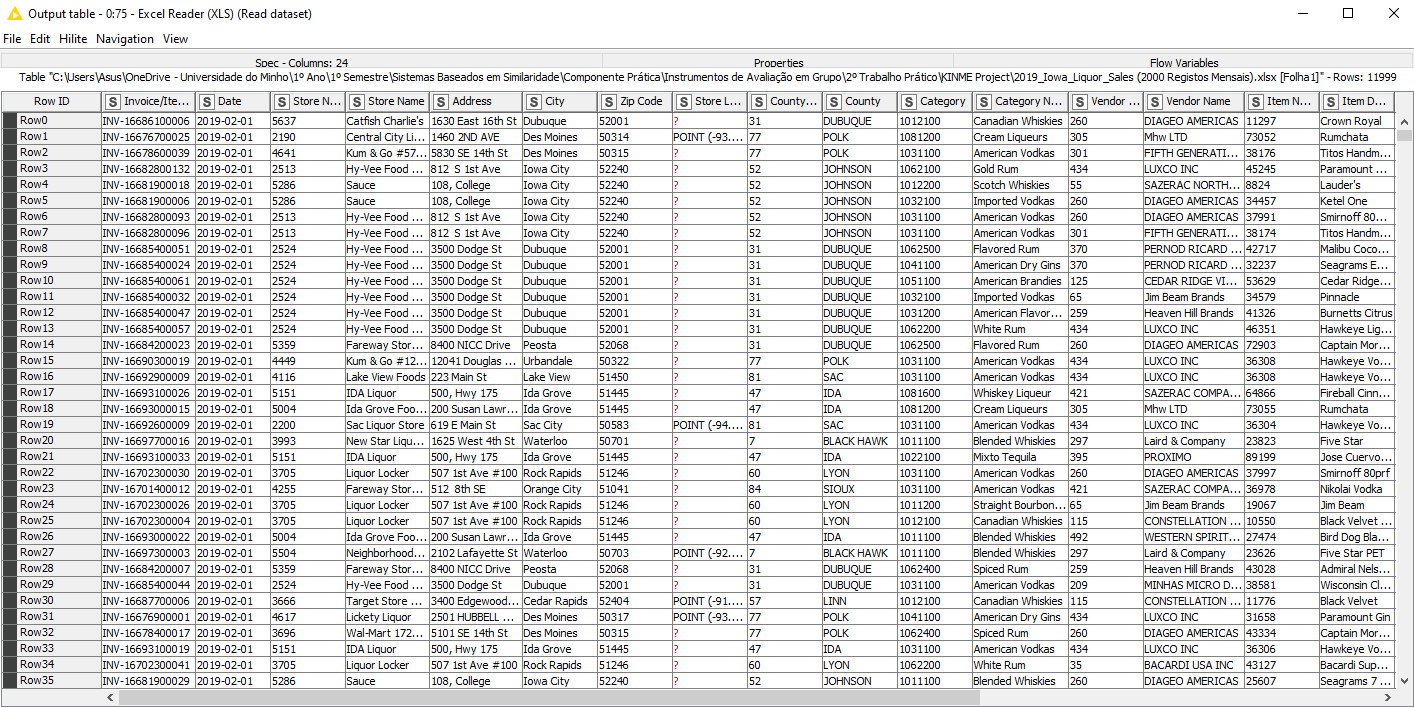


**Figura nº6 – Conjunto de dados extraído (Mês de Janeiro)**

Posteriormente ao preparo do *dataset* que servirá para a implementação do nosso sistema, foi necessário carregar o mesmo no *KNIME*. Para isso, recorremos mais uma vez ao nodo ***Excel Reader (XLS)*** para ler os 12001 registos. Por fim, obtivemos o *output* pretendido que se encontra evidenciado na Figura nº8



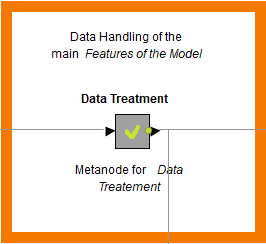
**Figura nº7 – Leitura do *dataset***



**Figura nº8 – Conjunto de Dados**

## Aplicar de nodos, de modo a fazer Tratamento de Dados

Após visualização do conjunto de dados, ficou percetível que era necessário efetuar o tratamento de algumas *features* presentes no *dataset*. Assim implementamos no nosso *workflow* um metanodo (***Data Treatment***) onde efetuamos toda a parte de tratamento de dados.



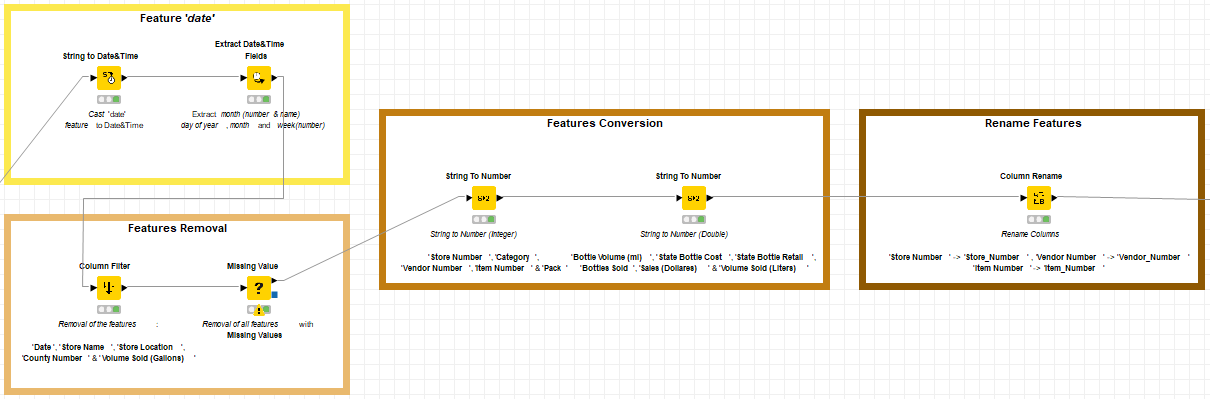
**Figura nº9 – Metanodo criado para o Tratamento de Dados do *dataset***

Ao nível de tratamento de dados, e no que a *feature* ‘*Date*’ diz respeito, fizemos o *cast* da mesma para o formato *Date&Time* recorrendo ao nodo ***String to Date&Time***. Posteriormente, e uma vez que achamos que seria relevante, extraímos da data o mês (número e nome), o dia do ano, do mês e da semana.

Dado o extenso número de *features* que o nosso conjunto de dados possuía, e com o objetivo de melhorar a performance do mesmo, resolvemos eliminar algumas *features*. Recorrendo ao nodo ***Column Filter*** e aplicando-o no *workflow*, as features ‘*Date*’, ‘*Store Name*’, ‘*Store Location*’, ‘*County Number*’ e ‘*Volume Sold (Gallons)*’ foram removidas dos conjuntos de dados. É importante salientar que o que esteve na génese da remoção destas *features* foi o facto de as mesmas replicarem informação já presente, por exemplo, mesmo removendo o nome da loja conseguimos identificar a mesma através do seu número identificador único. Depois de efetuado o processo anterior, e tendo verificado que existiam alguns registos com valores em falta, fizemos a remoção destes recorrendo ao nodo ***Missing Value***.

Tendo em conta que para a otimização dos dados para a implementação do sistema de recomendação é necessário que os atributos sejam do tipo *number*, foi necessário fazer a conversão de algumas *features*. Deste modo, convertemos as features ‘*Store Number*’, ‘*Category*’, ‘*Vendor Number*’, ‘*Item Number*’ e ‘*Pack*’ para o tipo de dados *integer*. As *features* ‘*Bottle Volume (ml)*’, ‘*State Bottle Cost*’, ‘*State Bottle Retail*’, ‘*Bottles Sold*’, ‘*Sales (Dollares)*’ e ‘*Volume Sold (Liters)*’ foram convertidas para o formato do tipo *double*, devido ao facto de possuírem casas decimais.

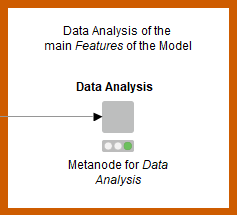
Ainda no que ao tratamento de dados diz respeito, numa fase mais adiantada do trabalho, nomeadamente na parte do cálculo do *MAE* e *MSE*, apercebemo-nos que o *KNIME* entrava em conflito com o nome de algumas *features* devido ao facto de estas possuírem ‘…*Number*’. Deste modo, recorrendo ao nodo ***Column Rename***, solucionámos este problema renomeando as colunas em questão colocando um ‘*\_*’ antes da palavra ‘*Number*’ de acordo com o exemplo seguinte ‘*Store\_Number*’.



**Figura nº10 – *Workflow* implementado para o Tratamento de Dados**

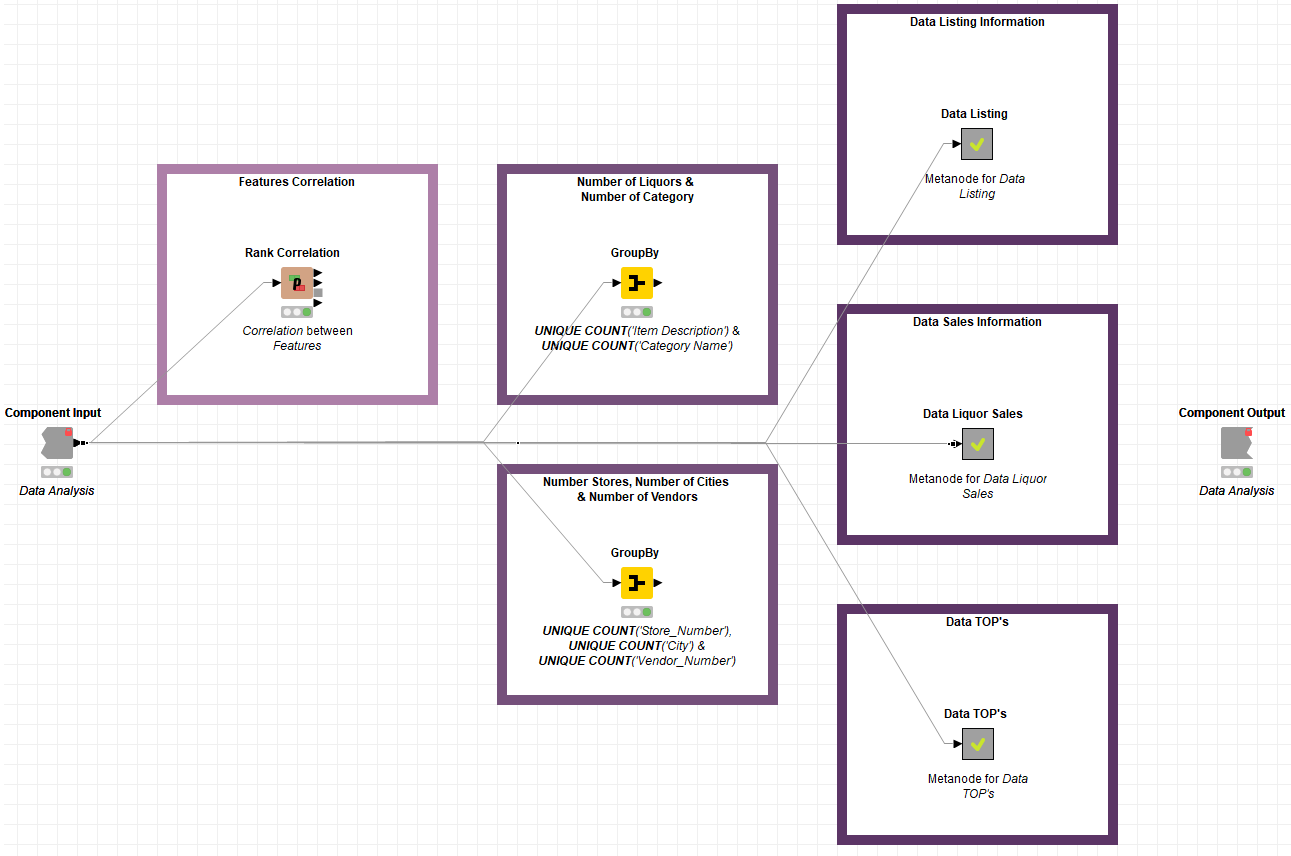
## Aplicar de nodos, de modo a fazer Análise de Dados

Depois de feito o tratamento de dados, foi necessário fazer uma análise ao *dataset*, com o objetivo de perceber algumas das particularidades do mesmo, de modo a facilitar na implementação do sistema de recomendação. Assim implementamos no nosso *workflow* um componente (***Data Analysis***) onde efetuamos toda a parte de tratamento de dados.

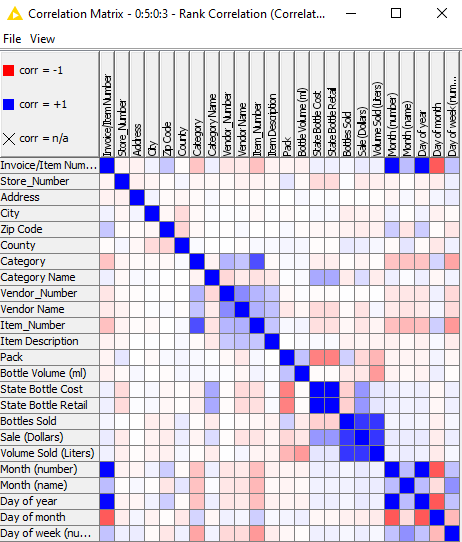


**Figura nº11 – Componente criado para o Análise de Dados do *dataset***

Ao nível de análise de dados, numa primeira abordagem decidimos tentar perceber um pouco mais sobre a relação entre as diversas *features* recorrendo ao nodo ***Rank Correlation***. De seguida, recorrendo a dois nodos ***GroupBy***, onde fizemos 5 *Unique Count* com o objetivo de saber o número de Licores, número de Categorias de Licores, número de Lojas, Cidades e Vendedores presentes no *dataset*. Com isto ficámos a perceber que existem 1161 Licores distribuídos por 46 Categorias diferentes. Sobre as Lojas existem no total 845, distribuídas por 270 Cidades e 86 Vendedores, o que permite concluir que o mesmo vendedor pode ter diversa lojas. Ao nível da relação entre *features* (**Figura nº13**), conseguimos perceber que existem *features*, algumas já esperadas, que se relacionam bem umas com as outras, como é o caso da *feature* ***Bottles Sold*** *e a feature* ***Sale.*** Por outro lado, existem *features* que não se relacionam tão bem, como é o caso das *features Pack* e *State Bottle Cost.*



**Figura nº12 – *Workflow* implementado para o Análise de Dados**



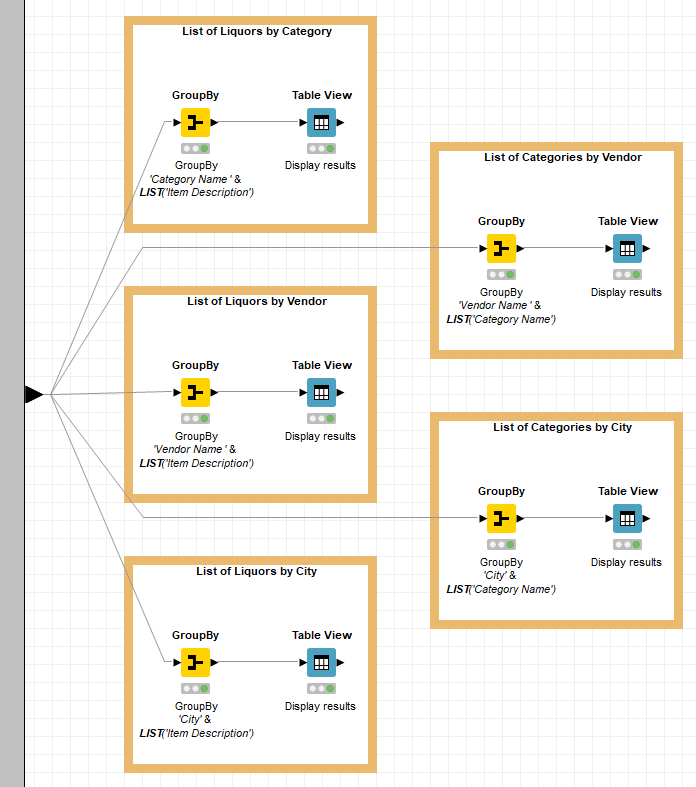
**Figura nº13 – Relação entre *features* do *dataset***

De acordo com o apresentado na **Figura nº12**, é percetível que para além dos já enunciados nodos, foram também criados mais 3 metanodes com o objetivo de fazer uma análise de dados mais detalhada, que permitirá tirar conclusões sobre como poderá ser implementado o sistema de recomendação. Cada um destes, e os seus respetivos resultados encontram-se devidamente explicados nos tópicos seguintes.

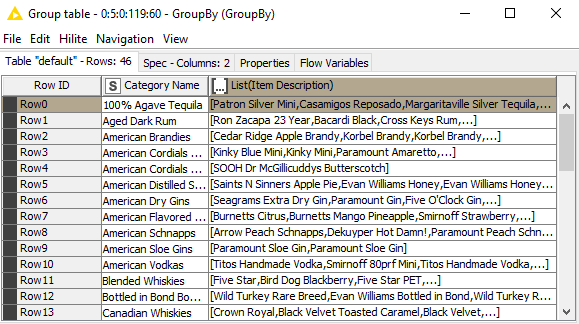
### Metanode *Data Listing Information*

O objetivo da implementação deste metanode, é como o próprio nome indica fazer a listagem da informação contida no *dataset*. Com a implementação de nodos feita neste *workflow*, conseguimos perceber visualizar as seguintes listagens de dados:

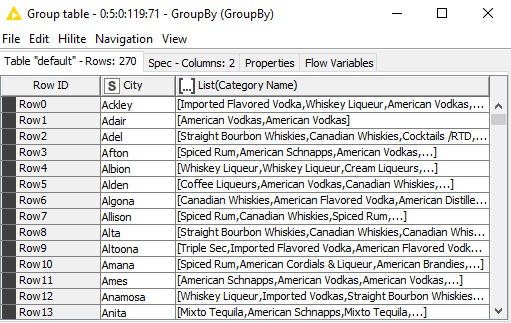
* Listagem de Licores por Categoria
* Listagem de Licores por Vendedor
* Listagem de Licores por Cidade
* Listagem de Categorias por Vendedor
* Listagem de Categorias por Cidade



**Figura nº14 – *Workflow* implementado no Metanode *Data Listing Information***



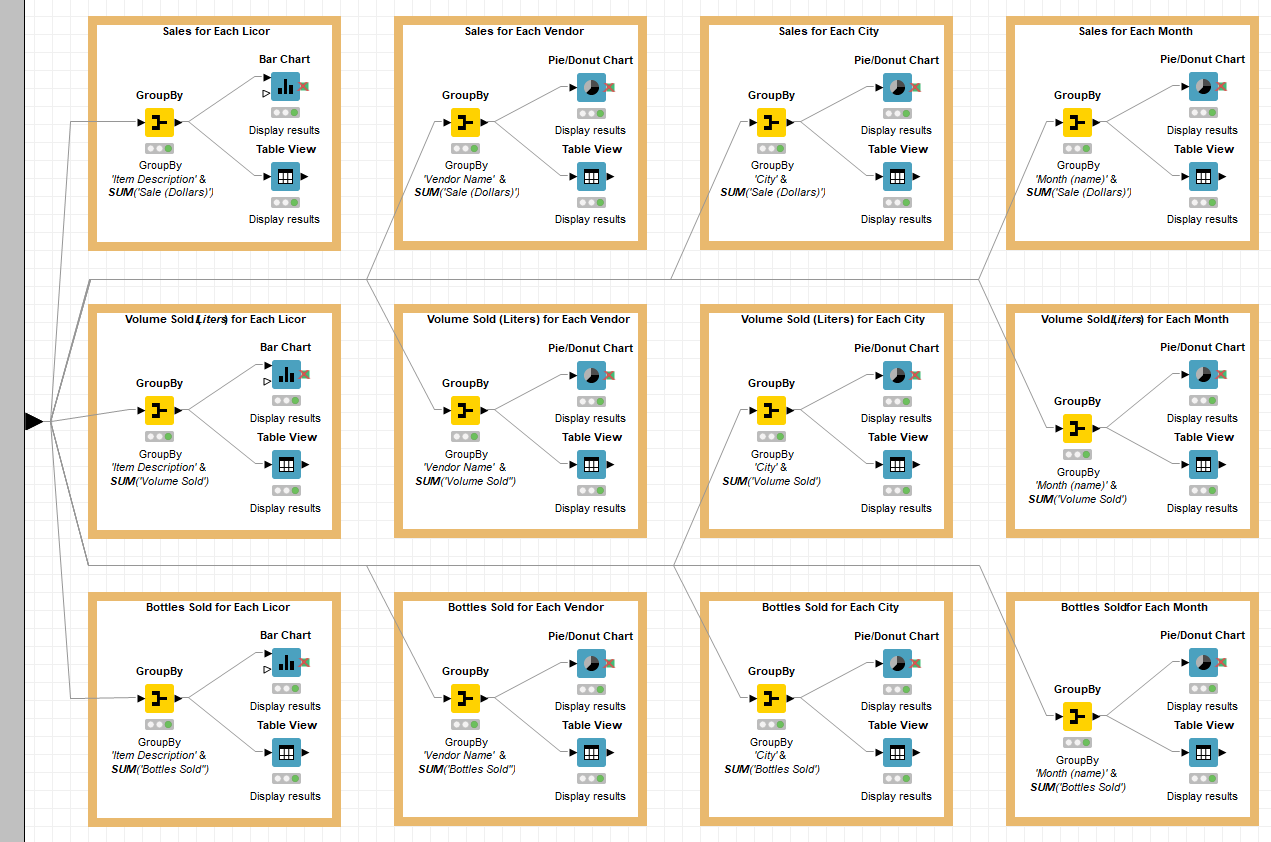
**Figura nº15 – Listagem de Licores por Categoria**



**Figura nº16 – Listagem de Licores por Cidade**

Com esta análise de dados efetuada, e após observação dos resultados obtidos, destacamos a Listagem de Licores por Categoria (**Figura nº15**) e a Listagem de Licores por Cidade (**Figura nº16**), pois achamos interessante para a conceção do sistema de recomendação o utilizador introduzir uma categoria de licores, e mediante as regras implementadas, o sistema sugerir um *TOP* de licores de acordo com a categoria inserida. Por outro lado, também pensamos que seja interessante, o utilizador introduzir o nome da cidade, e o sistema recomendar por exemplo, os mais vendidos nessa determinada cidade.

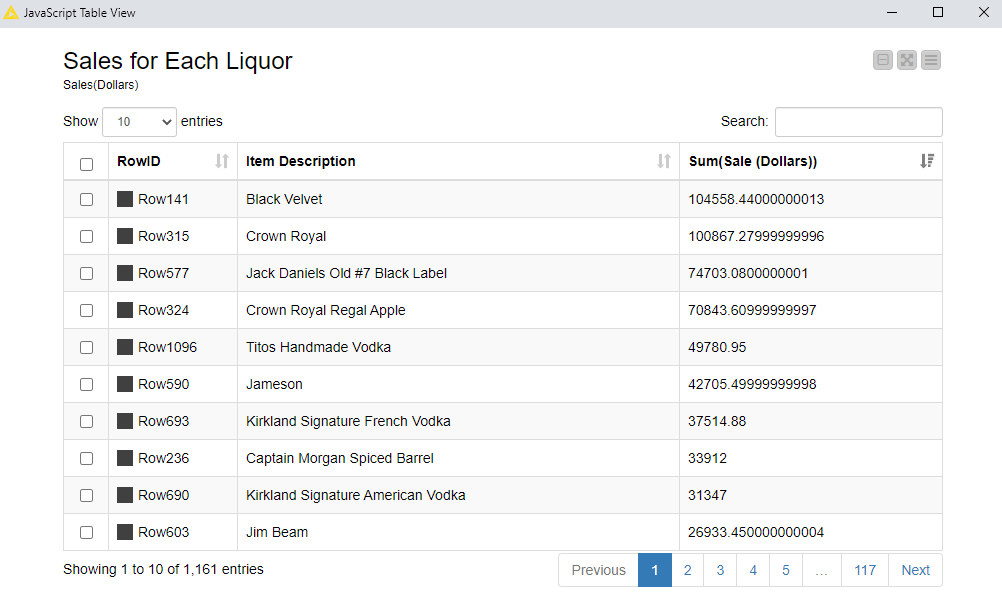
### Metanode *Data Sales Information*



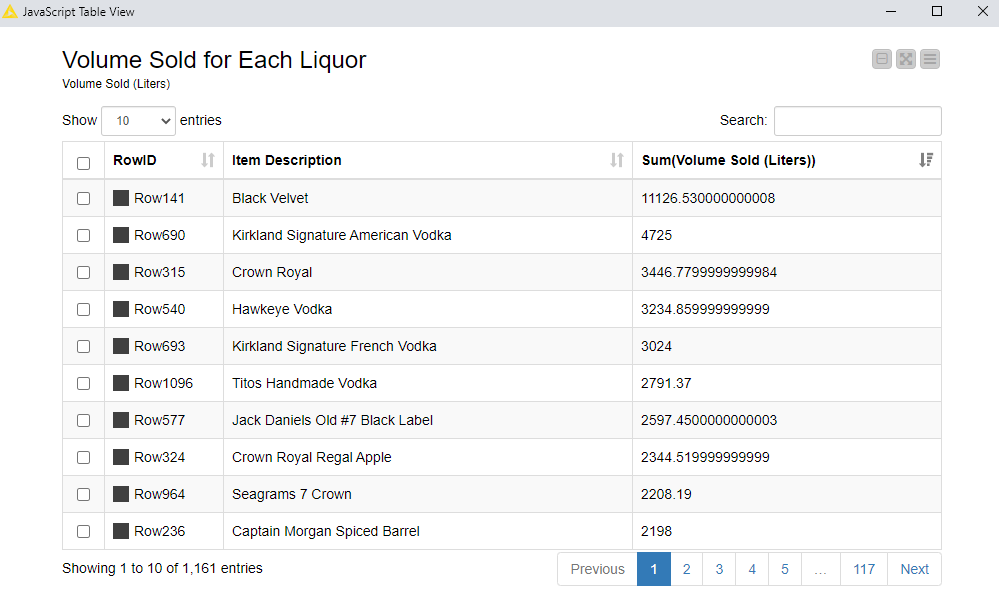
**Figura nº17 – *Workflow* implementado no Metanode *Data Sales Information***

Como o próprio nome indica, este *metanode* foi implementado com o objetivo de perceber um pouco mais sobre as vendas dos licores, pois alguns dos aspetos aqui observados poderão estar na génese das regras do sistema de recomendação. Com a implementação de nodos feita neste *workflow*, conseguimos visualizar as seguintes informações:

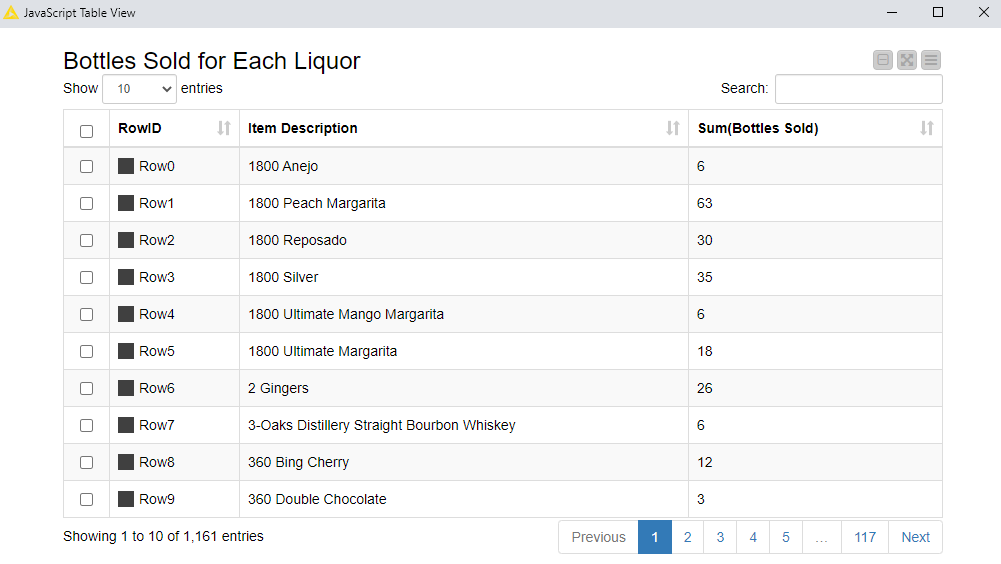
* Vendas de cada Licor
* Vendas de cada Vendedor
* Vendas de cada Cidade
* Vendas de cada Mês
* Volume Vendido em Litros de cada Licor
* Volume Vendido em Litros por cada Vendedor
* Volume Vendido em Litros por cada Cidade
* Volume Vendido em Litros por cada Mês
* Garrafas Vendidas de cada Licor
* Garrafas Vendidas por cada Vendedor
* Garrafas Vendidas por cada Cidade
* Garrafas Vendidas por cada Mês



**Figura nº18 – Vendas de cada Licor**

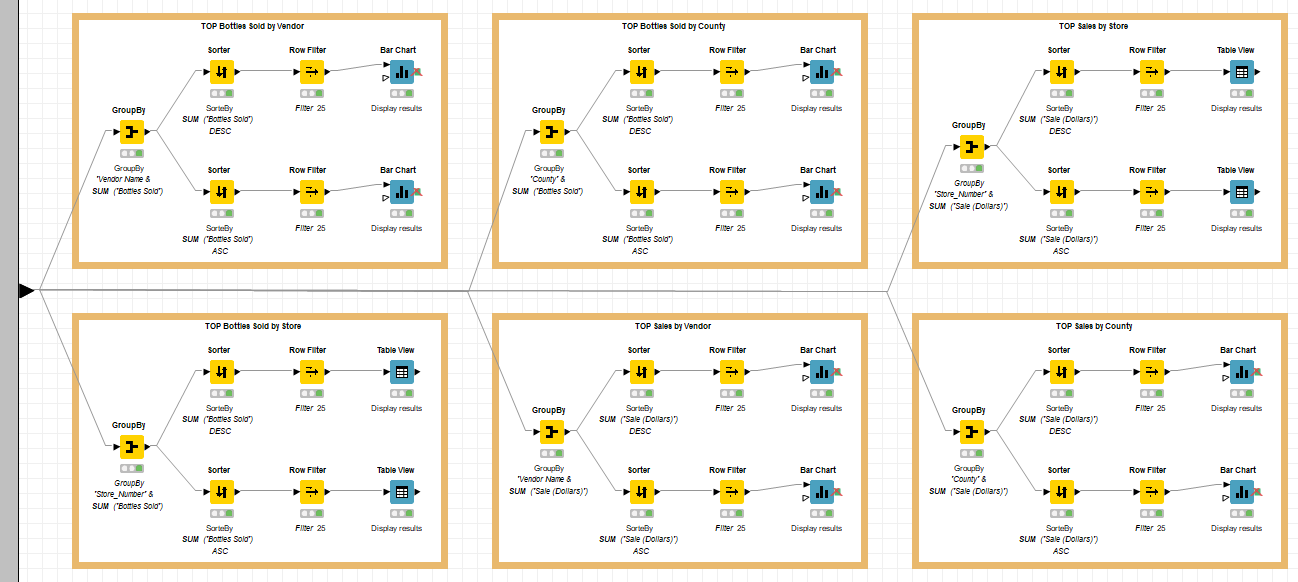


**Figura nº19 – Volume Vendido de cada Licor**



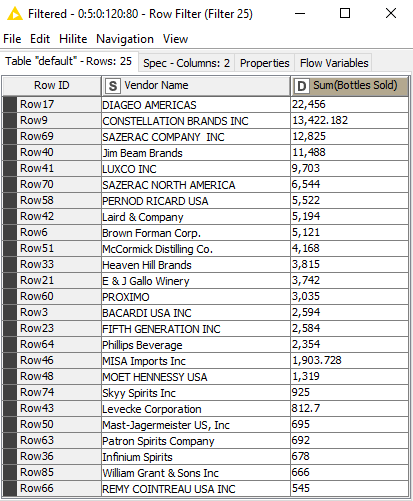
**Figura nº20 – Garrafas Vendidas de cada Licor**

### Metanode *Data TOP’s*

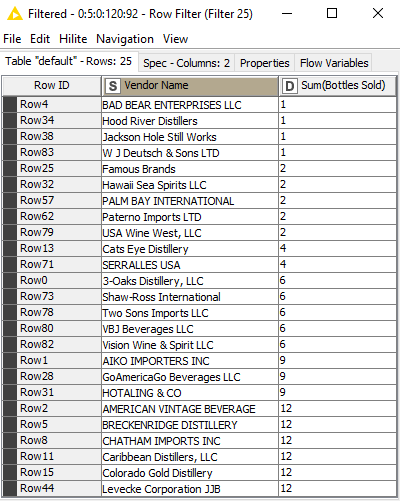


**Figura nº21 – *Workflow* implementado no Metanode *Data TOP’s***

Ao contrário dos outros 2 metanodes implementados, cujo seu principal objetivo era ajudar a estudar o *dataset* de modo a perceber alguns aspetos relevantes que possam estar na base do sistema de recomendação, este centra-se essencialmente na análise de dados, uma vez com a implementação destes *TOP’s*, conseguimos visualizar e perceber alguns dados estatísticos com o volume de vendas e número de garrafas vendidas de cada vendedor, destacando-se pela positiva o ***DIAGEO AMERICAS*** ao nível de garrafas de licor vendidas, e pela negativa de acordo com o volume de vendas, o vendedor ***BAD BEAR ENTERPRISES LLC***. Ao nível das vendas destaca-se pela positiva o vendedor ***DIAGEO AMERICAS*** com um total de vendas de 442 056,31$, e pela negativa o vendedor ***Jackson Hole Still Works*** com um valor de vendas de 26,25$.

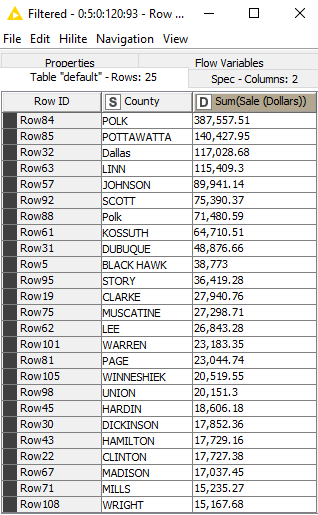


**Figura nº22 – Vendedores com mais Garrafas Vendidas**

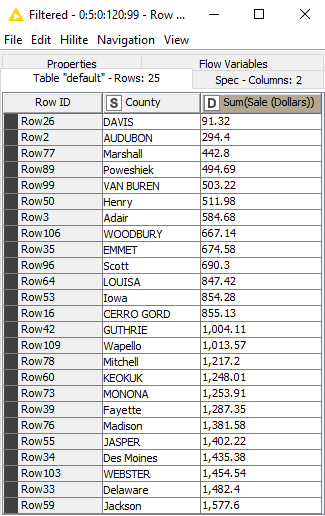


**Figura nº23 – Vendedores com menos Garrafas Vendidas**

Sobre os Municípios com mais vendas destacamos o ***POLK*** com um valor de 387 557,51$ e o com menos vendas é o ***DAVIS*** com um valor de 91,32$. Ao nível de garrafas vendidas, o município que vende mais é novamente o ***POLK*** com um total de 27 120 garrafas, e o que vende menos é o novamente o ***DAVIS*** com 6 garrafas vendidas.

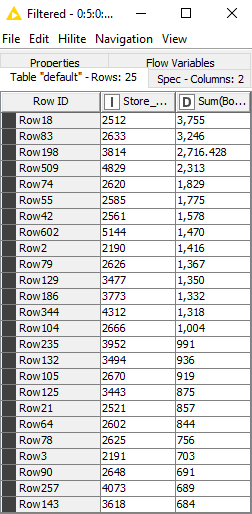


**Figura nº24 – Municípios com mais Vendas**

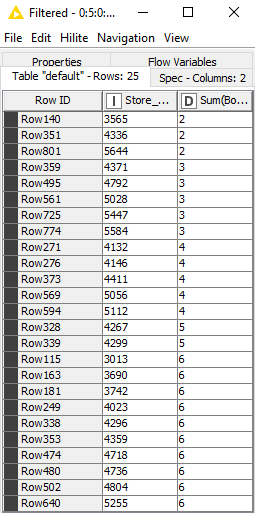


**Figura nº25 – Municípios com menos Vendas**

Sobre as lojas com mais garrafas vendidas destacamos a com o identificador 2512 (***Hy-Vee Wine and Spirits / Iowa City***) com um total de 3 755 garrafas vendidas, e a com o indentificador 3565 (***Hartig Drug Store #10 / Iowa City***) com menos garrafas vendidas



**Figura nº26 – Lojas com mais Garrafas Vendidas**

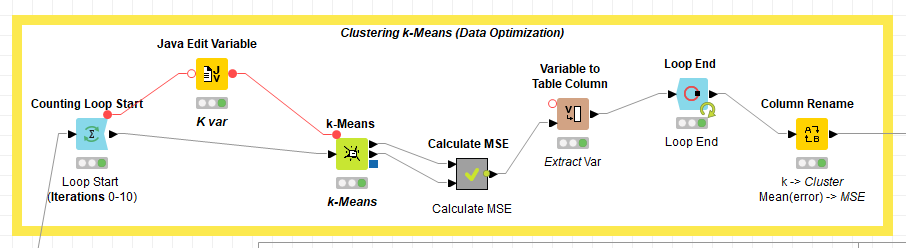


**Figura nº27 – Lojas com menos Garrafas Vendidas**

## Aplicar de nodos, de modo a fazer a Otimização do Modelo

No que a otimização de dados diz respeito, implementamos 4 abordagens diferentes, 2 com o **Método de Cotovelo** (*Elbow Method*), onde aplicamos diversos nodos com o objetivo de calcular o *MAE* e o *MSE*. Nas outras abordagens recorremos ao ***Clustering k-Means***, onde à semelhança do anterior, calculamos o *MAE* e *MSE*. O objetivo destas implementações, foi conseguir determinar qual a otimização ótima para o nosso modelo, ou seja, aquela que apresenta um menor valor de erro tendo todas por base configurações semelhantes, como por exemplo o número de *loops*. Nos tópicos seguintes apresentamos a solução de otimização ótima escolhida, e evidenciamos a implementação e teste de outras otimizações efetuadas.

### Solução de Otimização Ótima do Modelo



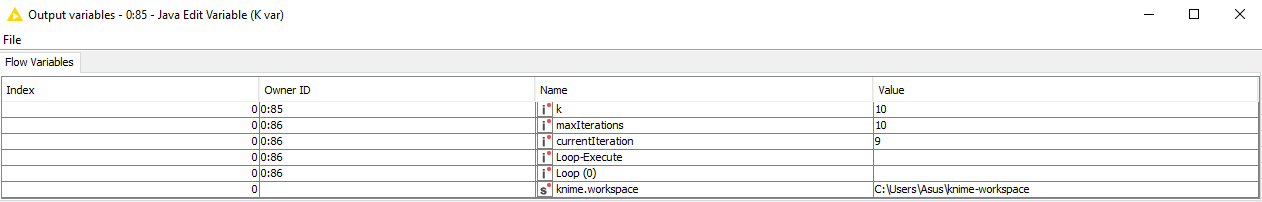
**Figura nº28 – Nodos implementados para a Otimização de Dados**

De acordo com a Figura apresentada, é possível verificar que aquela que consideramos como solução ótima de otimização, é recorrendo ao *Clustering k-Means* e efetuando o cálculo do *MSE*, uma vez que é aquela que apresenta um menor valor para o erro, como podemos verificar na Tabela nº 2.

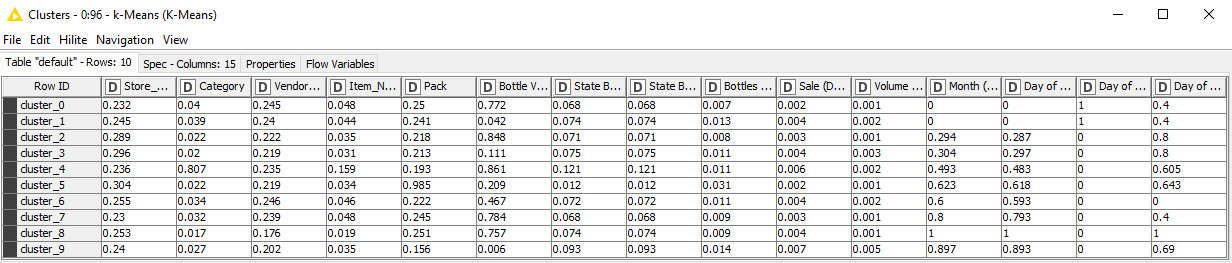
Ao nível de configurações implementadas, foram efetuados 10 *loops*. Recorrendo ao nodo ***Java Edit Variable*** definimos o nosso k, que posteriormente é utilizado como uma *flow variable* no nodo ***k-Means*** para definir o número de *clusters*. Ainda na configuração deste nodo filtramos as *features* que não são relevantes para o nosso Sistema de Recomendação baseado em *clusters*, tais como a ‘*Store\_Number*’, ‘*Vendor\_Number*’, ‘*Item\_Number*’, ‘*State Bottle Cost*’, ‘*Bottles Sold*’, ‘*Sale (Dollars)*’, ‘*Volume Sold (Liters)*’, ‘*Month (number)*’, ‘*Day of year*’, ‘*Day of month*’ e ‘*Day of week (number)*’.

|  |  |  |
| --- | --- | --- |
| Tipo de Otimização | Cálculo do Erro | Erro Obtido |
| *Elbow Method* | *MAE* | 0.064 |
| *Elbow Method* | *MSE* | 0.001 |
| *Clustering k-Means* | *MAE* | 0.072 |
| *Clustering k-Means* | *MSE* | 0.001 |

**Tabela nº2 – Erro Obtido na Otimização do Modelo**

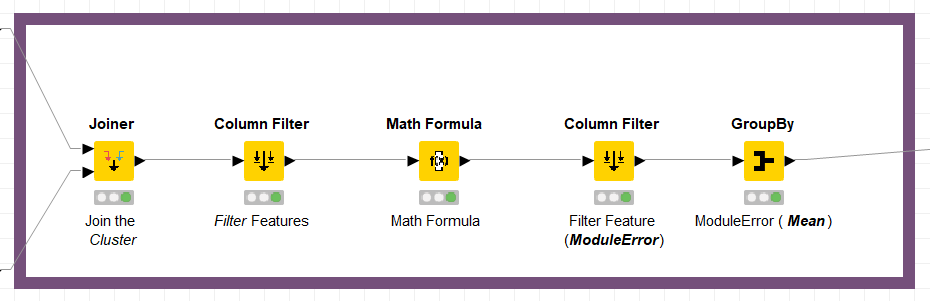


**Figura nº29 – *Output* obtido com o nodo *Java Edit Variable***



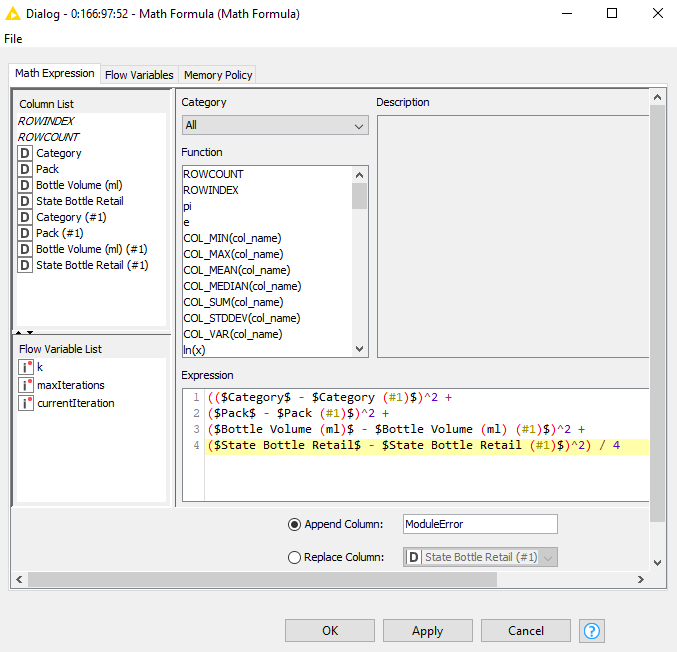
**Figura nº30 – *Clusters* obtidos (Nodo *k-Means*)**

Para o cálculo do *MSE*, implementamos o *workflow* apresentado na imagem seguinte. O nodo ***Joiner***, efetua a junção dos elementos presentes no conjunto de dados com os *clusters* anteriormente definidos. Com o nodo ***Column Filter***, filtramos as *features* que consideramos irrelevantes para o cálculo do erro, tais como o ‘*Invoice/Item Number*’, ‘*Store\_Number*’, ‘*Address*’, ‘*City*’, ‘*Zip Code*’, ‘*County*’, ‘*Category Name*’, ‘*Vendor\_Number*’, ‘*Vendor Name*’, ‘*Item\_Number*’, ‘*Item Description*’, ‘*State Bottle Cost*’, ‘*Bottles Sold*’, ‘*Sale (Dolars)*’, ‘*Volume Sold (Liters)*’, ‘*Month (number)*’, ‘*Month (name)*’, ‘*Day of year*’, ‘*Day of month*’ e ‘*Day of week (number)*’. É importante salientar que no cálculo do erro efetuado em todas as otimizações testadas, removemos as mesmas *features* de modo a não existirem discrepâncias no cálculo da função matemática que nos permite obter o erro.

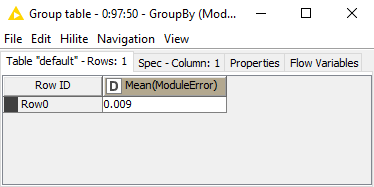


**Figura nº31 – Nodos implementados para o Cálculo do *MSE***

Recorrendo ao nodo ***Math Formula*** aplicamos a respetiva fórmula para o cálculo do *MSE*, de acordo com o evidenciado na imagem seguinte. De seguida, fizemos uma filtragem de *features* de modo a ficarmos apenas com o ***ModuleError***, que é o valor calculado e definido no nodo anterior, para cada um dos registos presentes no *dataset*. Por fim, recorrendo ao nodo ***GroupBy*** fizemos uma agregação manual da coluna ***ModuleError***, onde o tipo de agregação foi a média, e obtivemos o valor de 0.009 já enunciado na Tabela nº2.



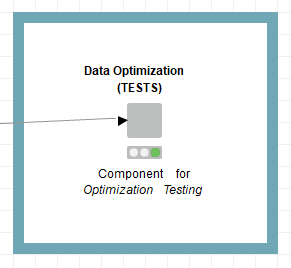
**Figura nº32 – Configurações aplicadas no nodo *Math Formula***



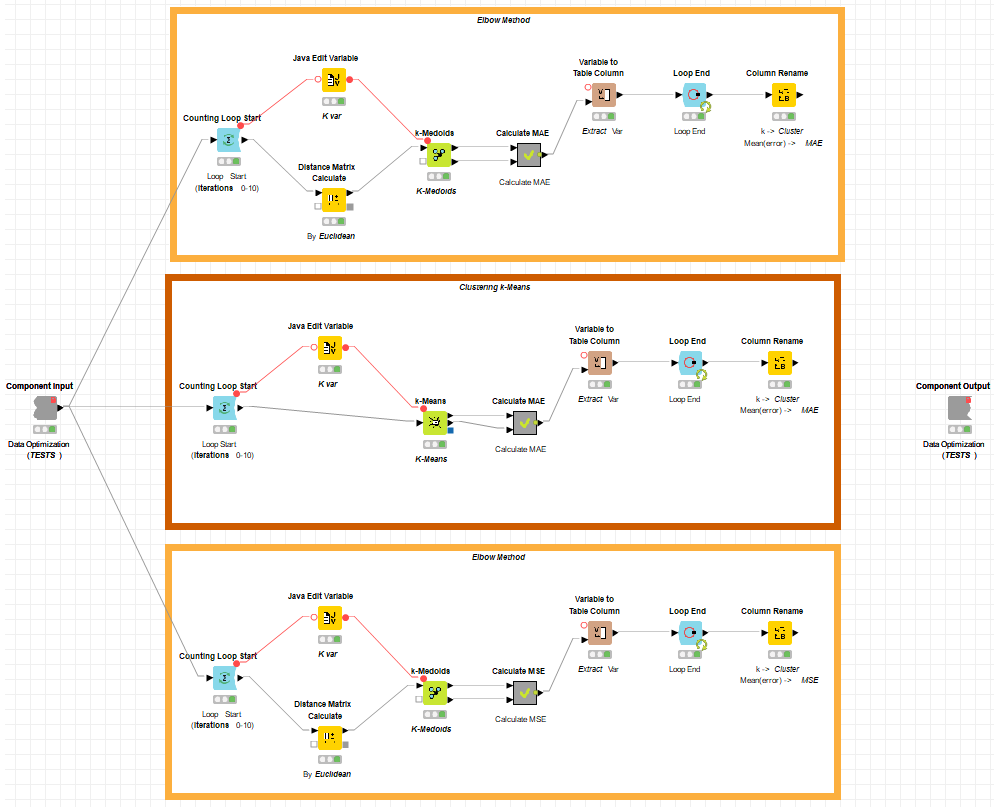
**Figura nº33 – Média do Erro Obtido**

### Evidência da Implementação de outras Soluções de Otimização do Modelo

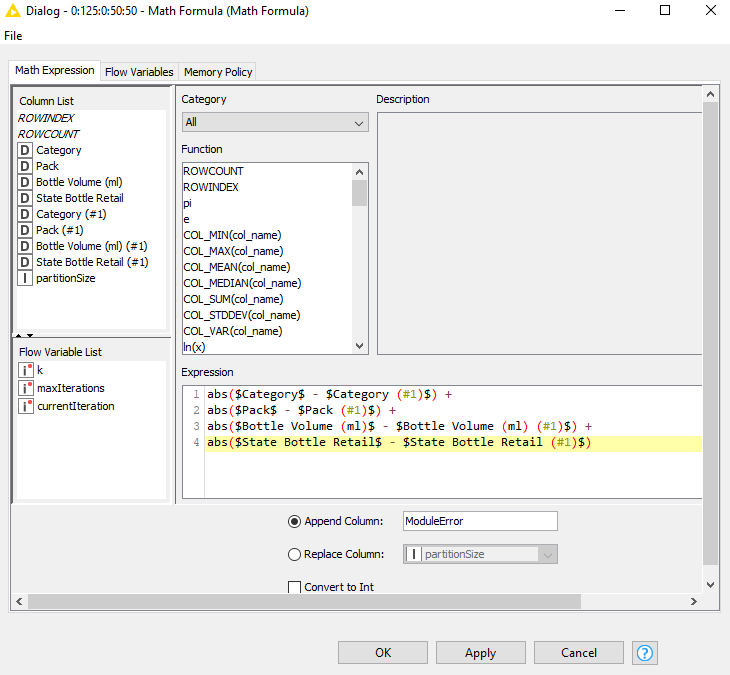
De acordo com o que já foi referido, implementamos e testamos várias formas de otimização do conjunto de dados com o objetivo de identificar a otimização ótima. Depois de termos escolhido o nosso método de otimização, armazenamos os restantes num componente (***Data Optimization (TESTS)***). Ao nível de configurações destas outras otimizações, as mesmas são semelhantes à ótima, uma vez que possuem o mesmo número de *loops*, e as suas únicas diferenças residem no método aplicado (*Elbow Method* ou *Clustering k-Means*) e nos casos do cálculo do *MAE*, uma vez que a fórmula matemática é diferente da do cálculo do *MSE*.



**Figura nº34 – Componente criado para Teste de outras Otimizações**



**Figura nº35 – *Workflow* implementado para Teste de outras Otimizações**

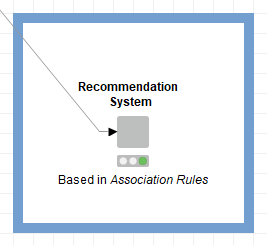


**Figura nº36 – Configurações aplicadas no nodo *Math Formula* (Cálculo *MAE*)**

# **Tarefa 2.** Conceção e Implementação de um Sistema de Recomendação

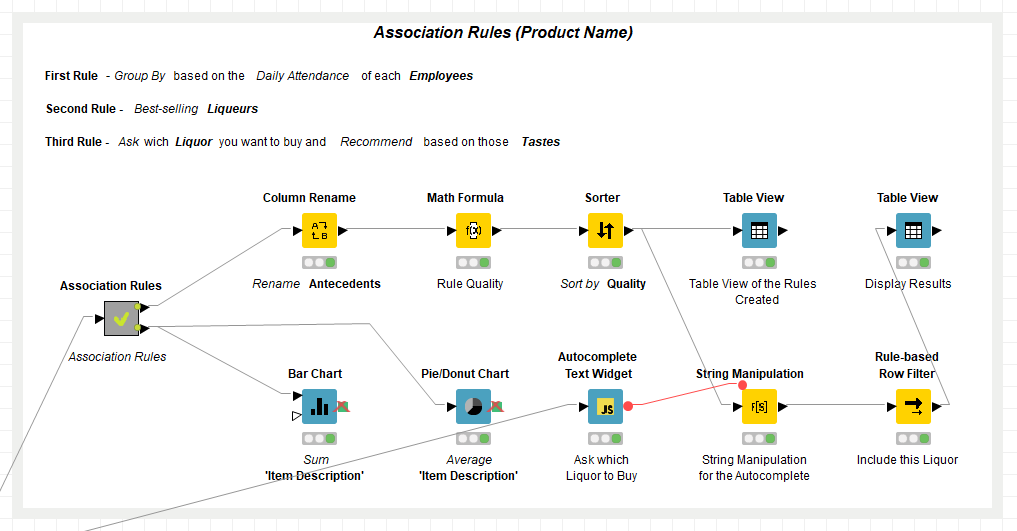
## Aplicar de nodos, baseados em Regras de Associação no Modelo

Para este processo de conceção e implementação de um Sistema de Recomendação baseado em Regras de Associação, criámos um metanode (***Recommendation System***) onde foram implementados dois *workflows*, um baseado na *feature* ‘*Product Name*’ e outro baseado na *feature* ‘*Category Name*’. Cada um destes *workflows* implementados e as suas particularidades, encontram-se devidamente explicados nos tópicos seguintes.



**Figura nº37 – Componente criado para o Sistema de Recomendação Baseado em Regras de Associação**

### Regras de Associação Implementadas baseadas na *feature* ‘*Product Name’*

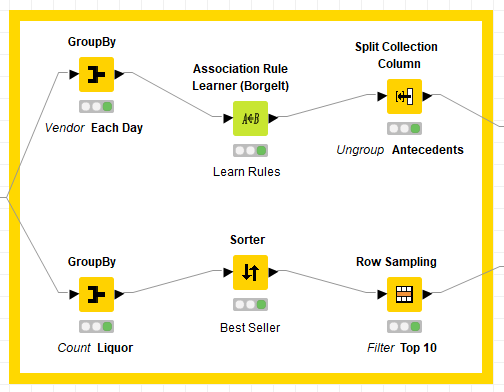


**Figura nº38 – *Workflow* implementado para a Definição de Regras de Associação**

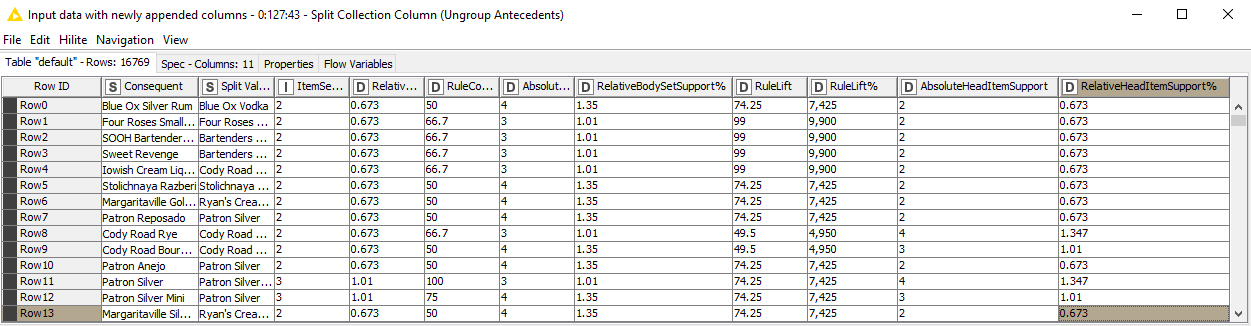
De acordo com a imagem apresentada, podemos observar que implementamos no *workflow*, as referidas Regras de Associação. O objetivo da 1ª Regra (‘*Group By* based on the *Daily Attendance* of each *Employees*’) era conseguir agrupar dados, baseado no atendimento diário de cada funcionário. A 2ª Regra (‘*Best-selling Liqueurs*’) permite-nos saber quais os licores e bebidas alcoólicas mais vendidas. Já a nossa 3ª Regra (‘*Ask* wich *Liquor* you want to buy and *Recommend* based on those *Tastes*’) foi desenvolvida com o propósito de saber qual o produto que o utilizador pretende adquirir, e com base nesse produto, fazer recomendações de acordo com os seus gostos.

Ao nível de implementações realizadas no *KNIME*, visando a implementação das regras de associação enunciadas, criámos um metanode (***Association Rules***), onde trabalhamos a parte a parte do vendedor de cada dia recorrendo ao nodo ***Group By***, agrupando as colunas ‘*Vendor\_Number*’, ‘*Month (number)*’ e ‘*Day of month*’, fazendo uma agregação manual de uma listagem dos itens da feature ‘*Item Description*’, para que depois aplicando o ***Association Rule Learner (Borgelt)*** seja possível aprender as regras definidas.

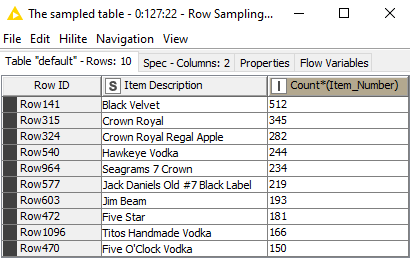
Na parte da definição e tratamento da 2ª regra, começamos por fazer um count dos diversos licores, agrupando a coluna ‘*Item Description*’, fazendo uma agregação manual com uma contagem dos itens da feature ‘*Item\_Number*’. Com o objetivo de saber qual o mais vendido, recorrendo ao ***Sorter***, fizemos uma ordenação pela contagem do atributo ‘*Item\_Number*’. Para finalizar, fizemos uma filtragem do Top 10.



**Figura nº39 – Nodos implementados no Metanode *Association Rules***

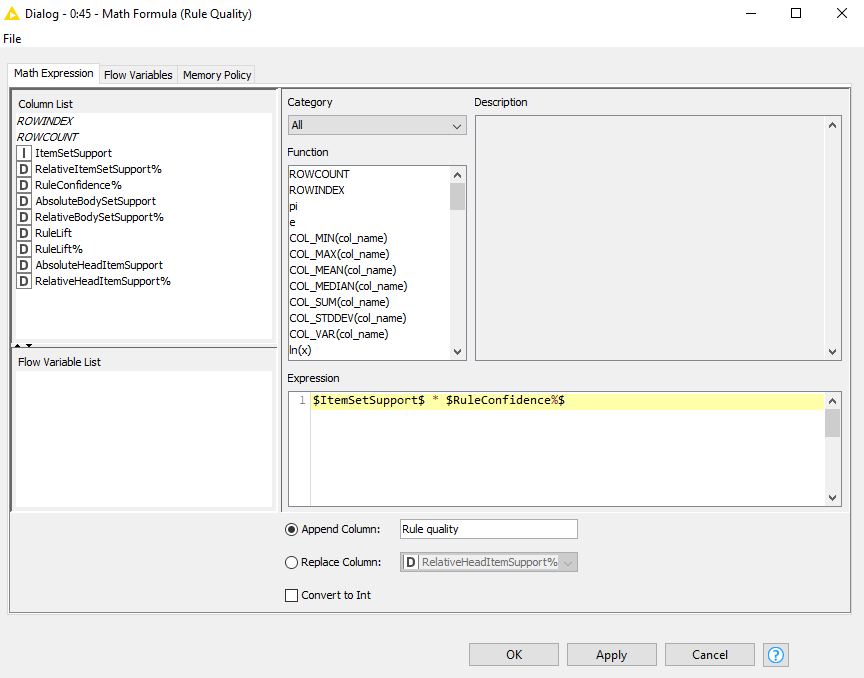


**Figura nº40 – *Output* Obtido após Definição da 1ª Regra (*Split Collection Column*)**



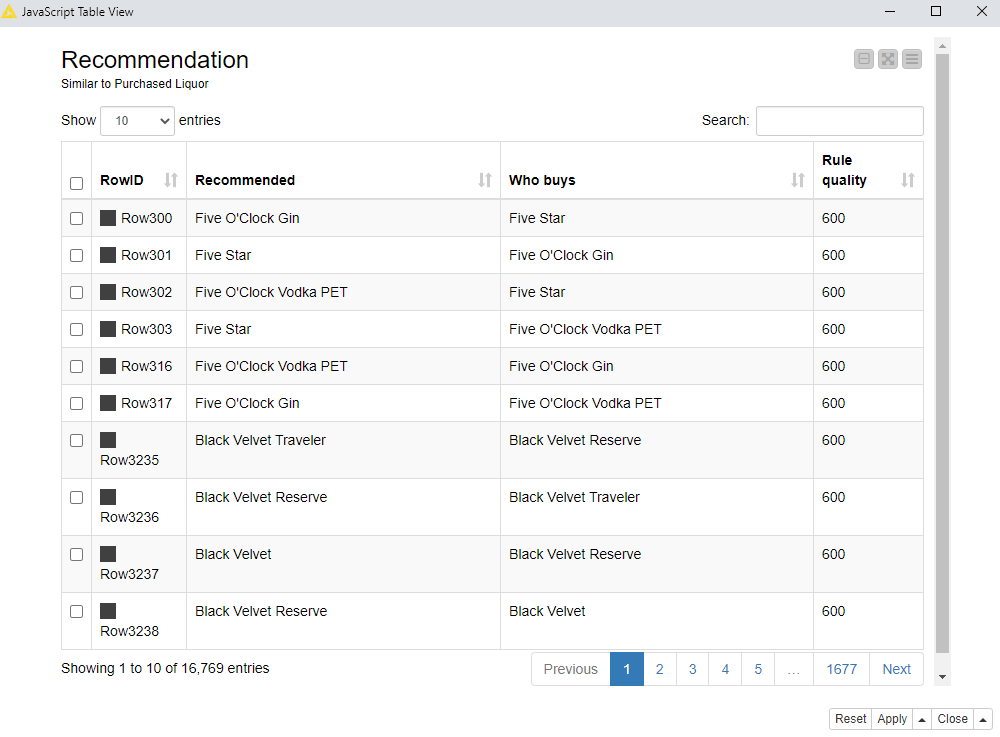
**Figura nº41 – *Output* Obtido após Definição da 2ª Regra (*Row Sampling*)**

Após a saída dos dados proveniente do componente implementado, uma das saídas está ligada ao nodo ***Column Rename***, com o objetivo renomear os antecedentes. De modo a calcularmos a qualidade da relação, recorremos à fórmula ‘*$ItemSetSupport$ \* $RuleConfidence%$*’, de acordo com o evidenciado na imagem seguinte que diz respeito ao nodo ***Math Formula***. De seguida é feita uma ordenação descendente de registos, mediante a *feature* ‘*Rule Quality*’, implementada anteriormente.



**Figura nº42 – Configurações aplicadas no nodo *Math Formula***

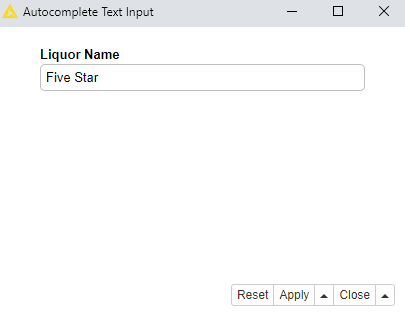
O ***Table View*** apresentado e implementado depois do nodo ***Sorter***, apenas serviu como como teste, de modo a conseguir visualizar se mediante as regras implementadas no nosso sistema, o mesmo recomendava outros licores mediante o licor comprado. De salientar que para a visualização de dados deste ***Table View***, não são necessário quaisquer *Input* de dados por parte do utilizador, uma vez que são mostrados todos os dados de produtos relacionados.



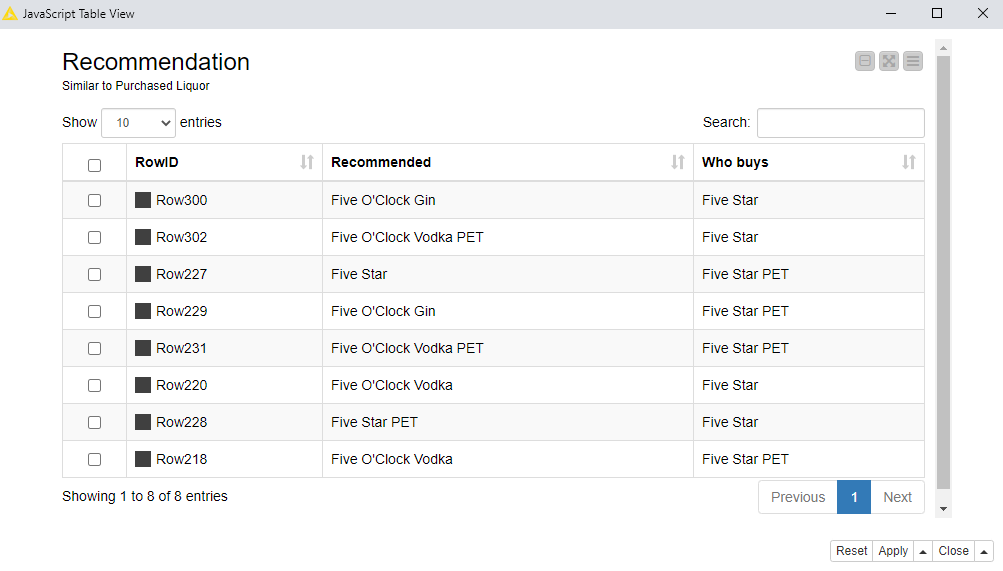
**Figura nº43 – *Table View* do *Output* das Regras de Associação**

Tendo em conta que o principal objetivo do sistema de recomendação é mediante um *Input* do utilizador, que neste caso concreto será o nome do licor/bebida comprado (‘*Product Name*’), serem feitas recomendações de outros licores mediante as regras de associação implementadas, foi necessário implementar no *workflow* o nodo ***Autocomplete Text Widget*** que irá receber os pretendidos *Input’s*. Ao nível de configurações relevantes deste nodo, selecionamos a *feature* do *dataset* que nos dava o nome dos licores (‘*Item Description*’) e definimos com valor *default* o licor ‘*Five Start*’.

Levando em consideração que uma determinada pessoa pode não saber o nome completo, ou até mesmo não saber escrever o nome de um determinado licor, implementamos o nodo ***String Manipulation***, com o objetivo de fazer um *join* com o que o utilizador escreve, e com o que é guardado na variável do *Autocomplete* (‘*join("\*", $${SAutocomplete string input}$$, "\*")*’) Por fim, com a implementação do nodo ***Roled-based Row Filter***, implementamos a regra que nos permite fazer a associação entre o produto comprado e o produto a recomendar. Isto só é possível, mediante uma operação *LIKE* entre as variáveis ‘*who buys*’, que armazena o nome do licor comprado, e ‘*iWant*’, que armazena o nome dos licores a recomendar (‘*$Who buys$ LIKE $iWant$ => TRUE*’).

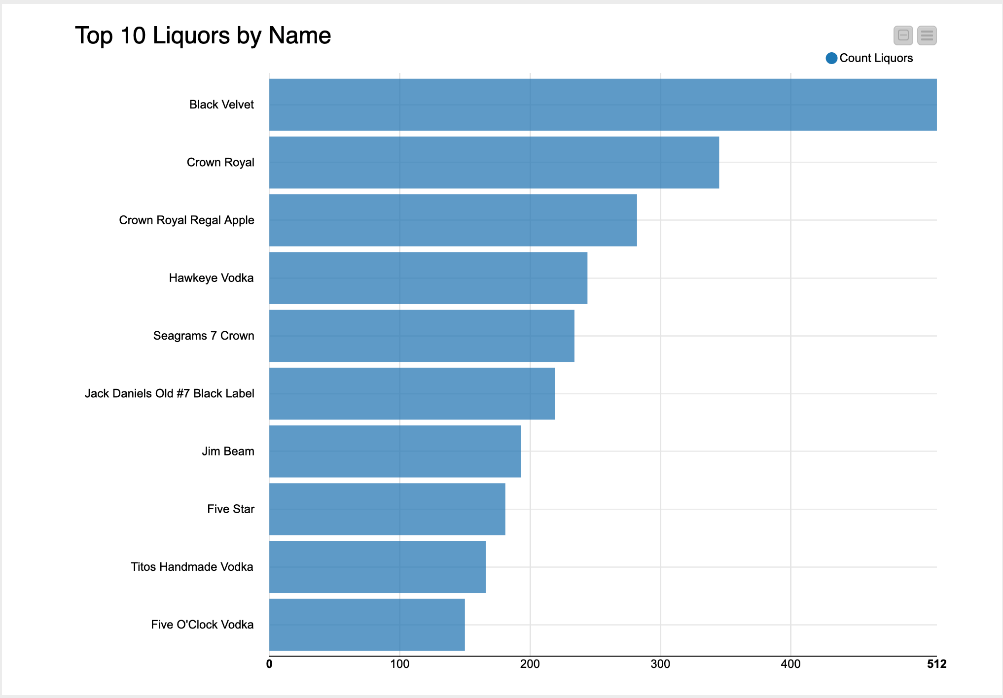


**Figura nº44 – *Input* para o Nome do Licor**

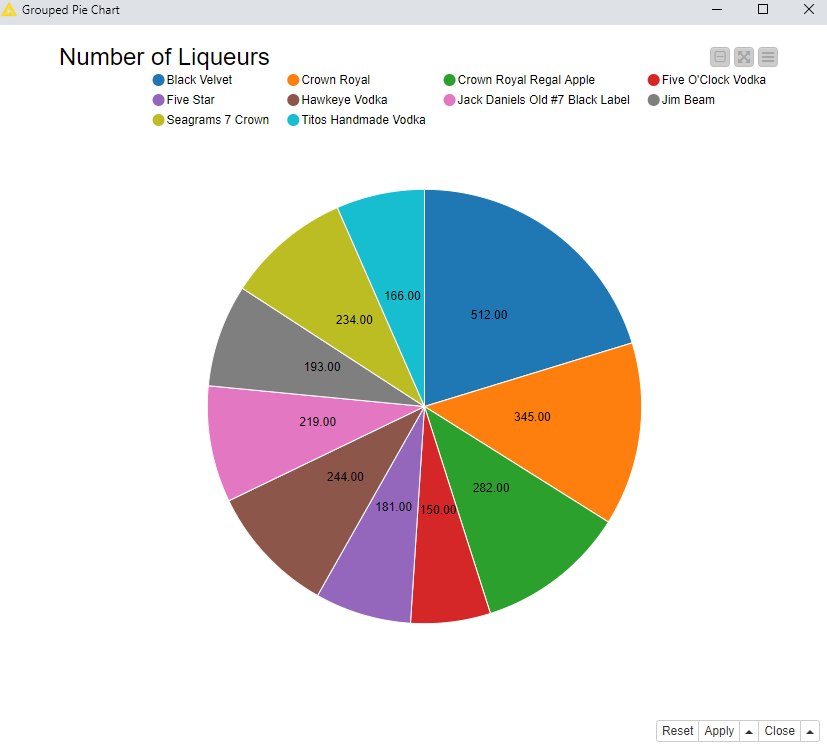


**Figura nº45 – *Output* gerado pelo Sistema de Recomendação mediante o Nome do Licor**

Ainda sobre os dois restantes nodos implementados neste *workflow* (***Bar Chart*** e ***Pie/Donut Chart***), os menos têm como objetivo a visualização de dados, segundo um *COUNT* de licores, uma ordenação pelo mais vendido e um filtro de *TOP 10*, implementados nos nodos que se encontram na parte inferior do *workflow* do metanode ***Association Rules***.



**Figura nº46 – *Output* do Bar Chart (TOP 10 de Licores)**



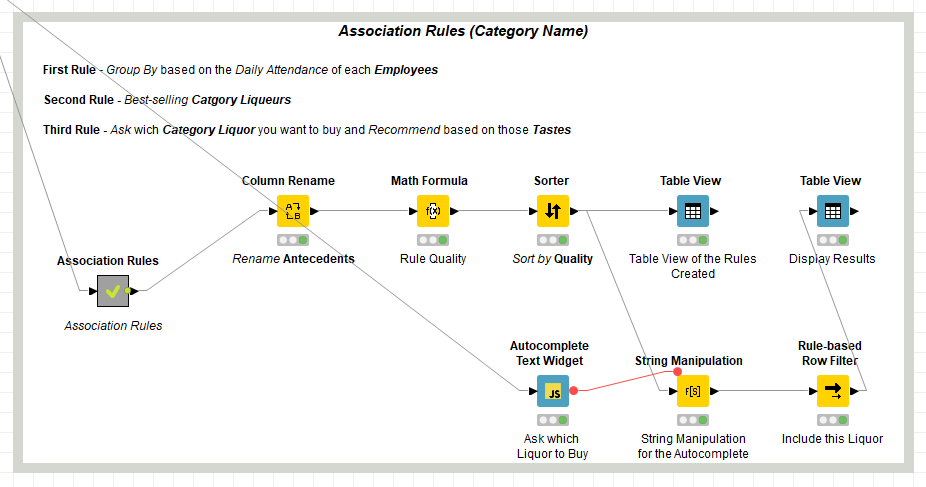
**Figura nº47 – *Output* do Pie/Donut Chart (TOP 10 de Licores)**

Foi adicionada nesta secção para análise do sistema de recomendação baseada no *feedback* obtido pelos utilizadores, isto é, após a visualização dos dados é perguntado ao utilizador se concorda com as recomendações, mediante isso seria possível obter estatísticas como:

1. Quais os produtos mais recomendados?
2. Nível de precisão da recomendação?
3. Número de clientes com recomendações.

Mediante estas questão e análises o sistema de recomendação poderia ser refinado gradualmente.

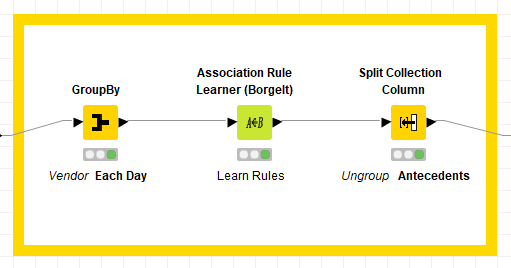
### Regras de Associação Implementadas baseadas na *feature* ‘*Category Name’*



**Figura nº48 – *Workflow* implementado para a Definição de Regras de Associação**

À semelhança do implementado nas regras de associação segundo o nome do produto, foi feita uma abordagem semelhante, tendo as principais diferenças residido na implementação de nodos no metanode ***Association Rules***, e na configuração do nodo ***Autocomplete Text Widget***. O objetivo da implementação das regras de associação baseadas na *feature* ‘*Category Name*’ continuou a ser em primeiro lugar a agrupação dados, baseado no atendimento diário de cada funcionário, depois conseguir saber quais as categorias de licores e bebidas alcoólicas mais vendidas, de modo numa fase final saber qual o produto que o utilizador pretende adquirir, e com base nesse produto, fazer recomendações de acordo com os seus gostos.

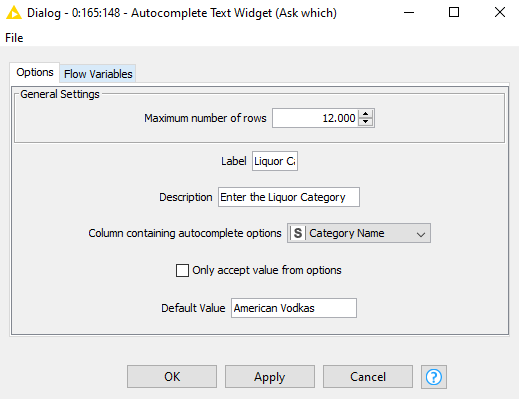
Ao nível da configuração do metanode ***Association Rules***, e com o objetivo de não tornar a informação demasiado repetitiva, não implementamos os nodos que nos permitiram anteriormente obter o TOP 10, ficando assim apenas com os nodos que nos permitem fazer e implementar o nosso sistema de recomendação baseado em associações.



**Figura nº49 – Nodos implementados no Metanode *Association Rules***

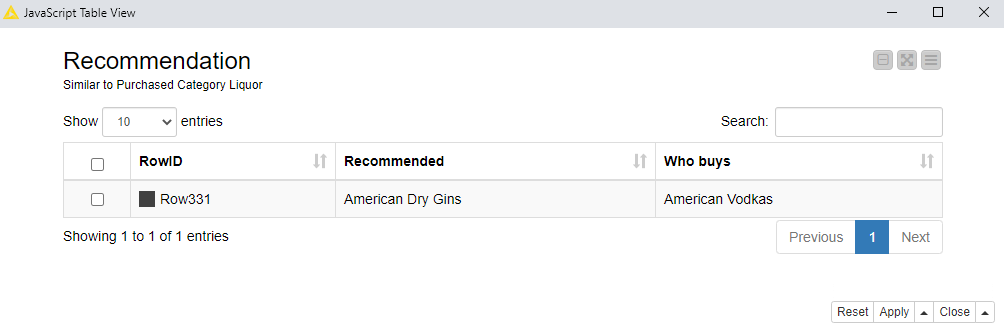
No nodo ***Column Rename***, mais uma vez são renomeados os antecedentes, e com o objetivo de calcularmos a qualidade da relação, recorremos à fórmula anterior ‘*$ItemSetSupport$ \* $RuleConfidence%$*’, implementada no nodo ***Math Formula***. Posteriormente é feita uma ordenação descendente de registos, mediante a *feature* ‘*Rule Quality*’. O ***Table View*** implementado, serviu mais uma vez como como teste, de modo a conseguir visualizar se mediante as regras implementadas no nosso sistema, o mesmo recomendava outros licores mediante a categoria do licor comprado.

Sobre a configuração do nodo ***Autocomplete Text Widget***, esta sim, com alguma diferença da implementada no sistema de recomendação anterior baseada no ‘*Product Name*’, irá receber os pretendidos *Input’s* relacionados com a categoria do produto. Para isso selecionamos a *feature* do *dataset* que nos dava o nome dos licores (‘*Category Name*’) e definimos com valor *default* a categoria de licores ‘A*merican Vodkas*’.



**Figura nº50 – Configurações aplicadas no nodo *Autocomplete Text Widget***

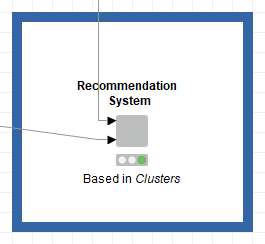
O nodo ***String Manipulation***, tem igualmente como objetivo fazer um *join* com a categoria de licores que o utilizador escreve, com o que é guardado na variável do *Autocomplete*. Já no nodo ***Roled-based Row Filter***, implementamos a regra que nos permite fazer a associação entre a categoria do produto comprado e a categoria do produto a recomendar.



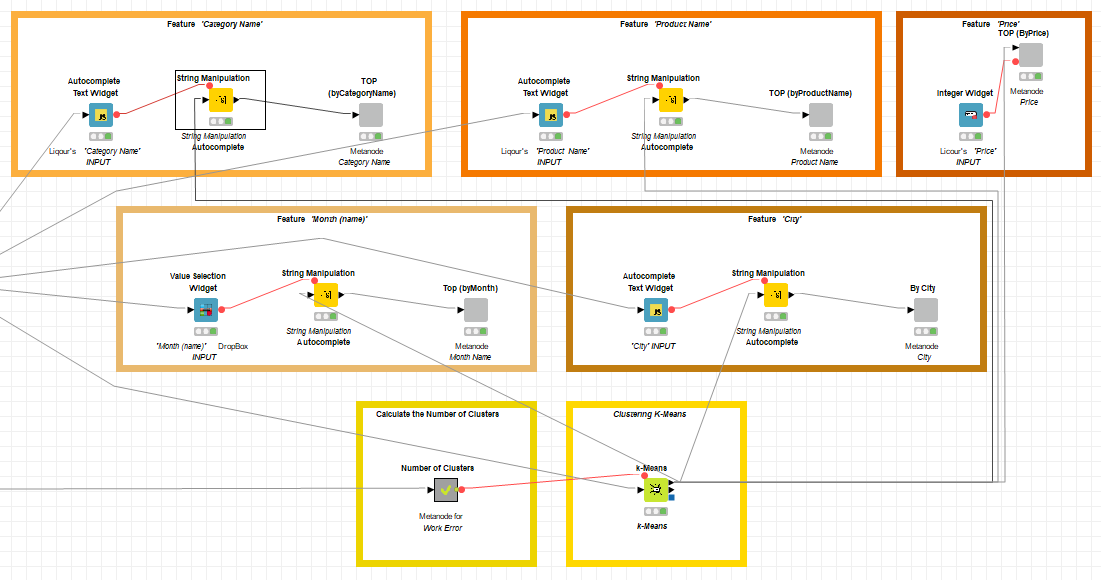
**Figura nº51 – *Output* gerado pelo Sistema de Recomendação mediante a Categoria do Licor**

## Aplicar de nodos, baseados em *Clusters* no Modelo

Para este processo de conceção e implementação de um Sistema de Recomendação baseado em *Clusters*, criámos um metanode (***Recommendation System***) onde foram implementados cinco sistemas baseados em clusters, para as *features* ‘*Category Name*’, ‘*Product Name*’, ‘*Price*’, ‘*Month (name)*’ e ‘*City*’. Cada um destes *workflows* implementados e as suas particularidades, encontram-se devidamente explicados nos tópicos seguintes.



**Figura nº52 – Componente criado para o Sistema de Recomendação Baseado em *Clusters***

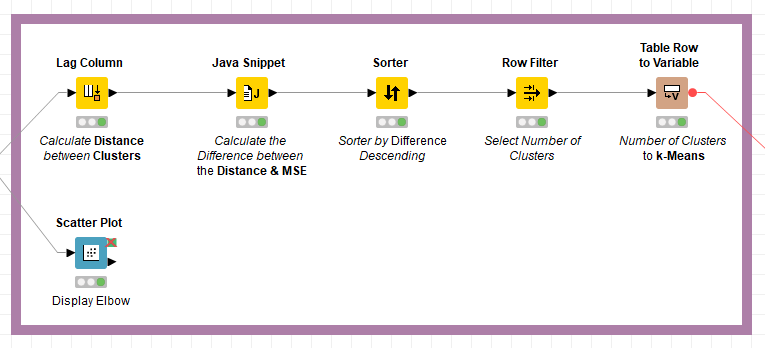


**Figura nº53 – *Workflow* Implementado para o Sistema de Recomendação Baseado em *Clusters***

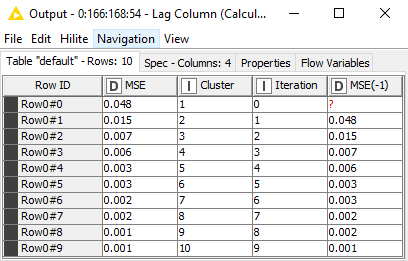
Sobre o *workflow* acima evidenciado, como já foi referido na ***Tarefa 1*** ***e*** (Aplicar nodos, de modo a fazer a Otimização do Modelo), a solução ótima de otimização encontrada foi recorrendo ao *Clustering k-Means* com o cálculo do erro segundo a fórmula matemática do *MSE*. A forma de como foi feita a implementação deste tipo de otimização encontra-se devidamente explicada no tópico referenciado, pelo que de modo a não ter informação repetida, não é necessário voltar a explicar abordagem feita, e quais foram os motivos pelos quais optamos por este tipo de *clustering*.

Sobre o metanode ***Number of Clusters***, que foi implementando com o objetivo de determinar o número ideal de clusters, foi calculada a distância entre clusters em primeiro lugar. Para isso recorremos ao nodo ***Lag Column***, onde definimos que a *Column to lag* seria a do *MSE*, com um *Lag* e *Lag interval* de *1*. De seguida, foi calculada a diferença entre a **distância *clusters*** e o ***valor obtido no erro*** (*MSE*), recorrendo à função implementada em *Java* presente na **Figura nº54**. Recorrendo ao ***Sorter***, foi feita uma ordenação de forma descendente do valor da diferença obtido no nodo anterior. Mediante o output obtido no ***Sorter***, foi percetível que *Row0#1* é aquele que apresenta uma maior diferença entre a distância e o erro, pelo que foi o selecionado no ***Row Filter***, onde o número ideal de *clusters* é 2. No nodo ***Table Row to Variable***, foi armazenado numa variável o número ideal de *clusters* a passar para o nodo ***k-Means***.

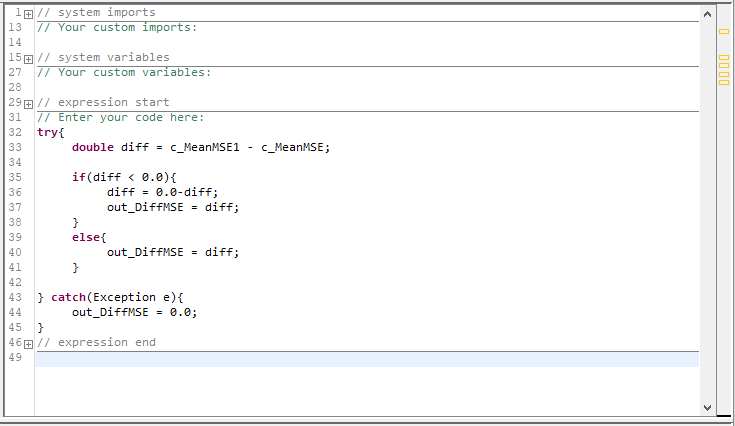
Através do output visualizado na ***Scatter Plot***, foi possível confirmar que um número ideal de *clusters* era de facto 2, dado que a maior diferença entre o número de *clusters* é entre 1 e 2.



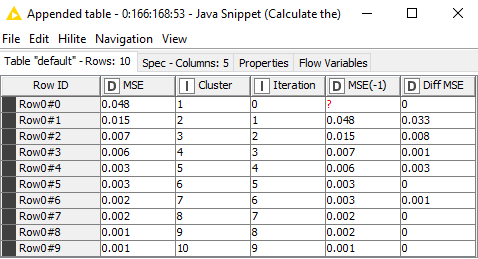
**Figura nº54 – Nodos implementados no Metanode *Number of Clusters***



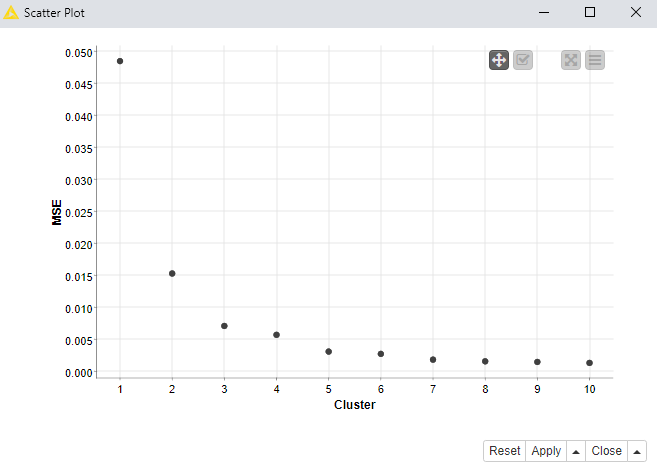
**Figura nº55 – *Output* obtido com o Cálculo da Distância em *Clusters***



**Figura nº56 – Função em Java, para Cálculo da Diferença entre a Distância e o Erro**

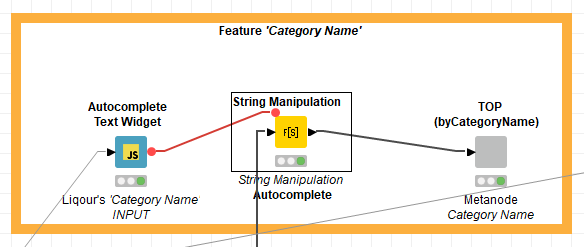


**Figura nº57 – *Output* obtido com o Cálculo da Diferença entre a Distância e o Erro**

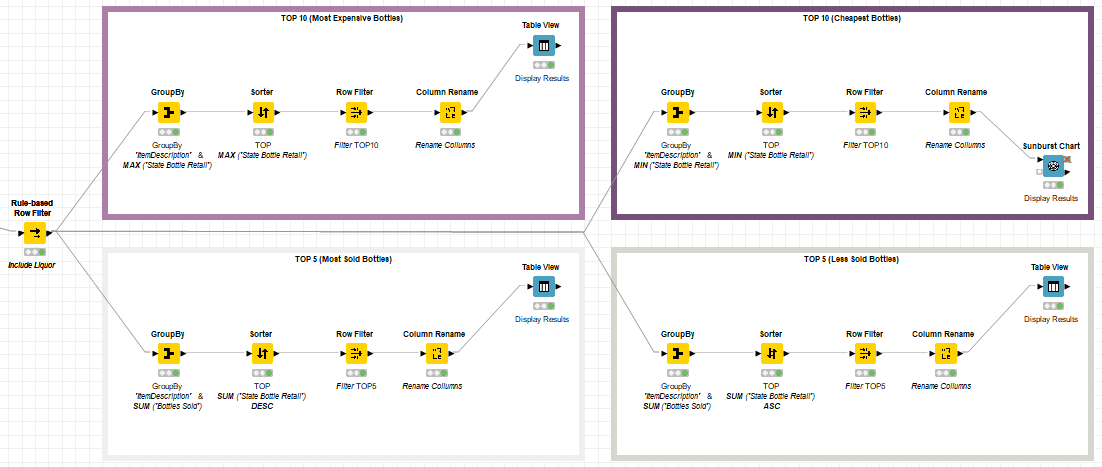


**Figura nº58 – *Output* Scatter Plot**

### *Clusters* baseados na *feature* ‘*Category Name’*



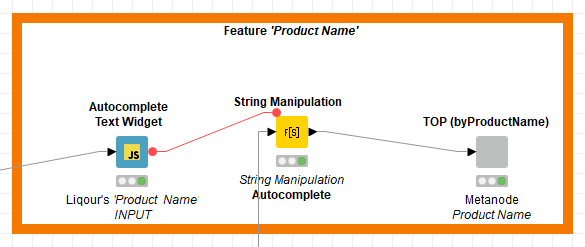
**Figura nº59 – Nodos implementados para o Sistema de Recomendação baseado na Categoria do Licor**



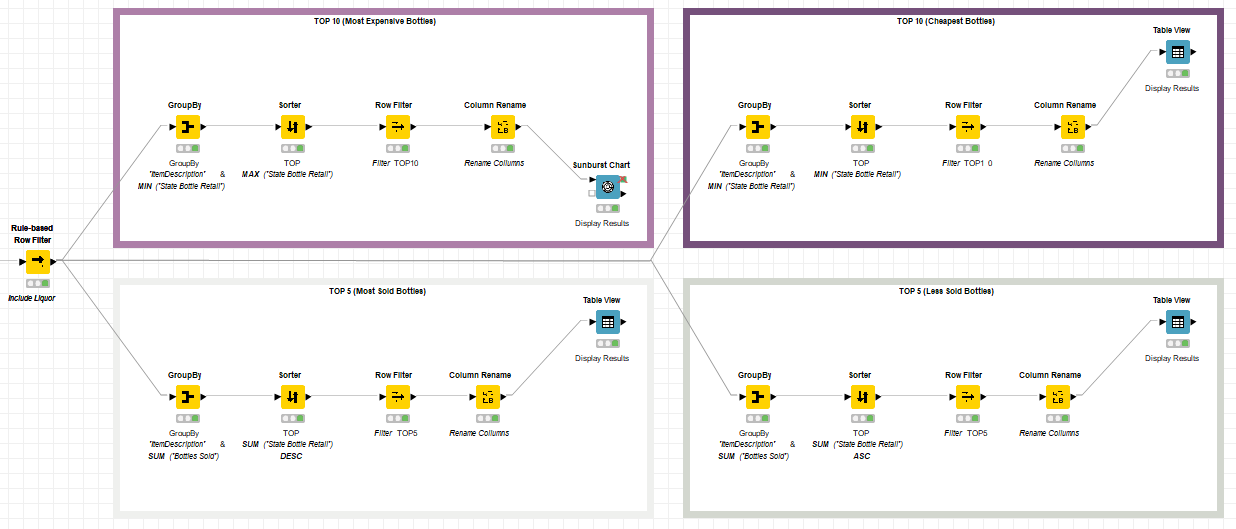
**Figura nº60 – *Workflow* Implementado**

De acordo com as **Figuras nº59 e 60**, podemos observar que recorrendo ao nodo ***Autocomplete Text Widget***, recebemos o *INPUT* do utilizador sobre a Categoria de Licores a pesquisar. O valor *default* introduzido é o ‘*Gold Rum*’. No nodo ***String Manipulation***, fazemos um *join* com a categoria de licores que o utilizador escreve, com o que é guardado na variável do *Autocomplete*. Já no nodo ***Roled-based Row Filter***, implementamos a regra que nos permite fazer a associação entre a categoria do licor e o licor a recomendar. No caso concreto deste caso, fazemos recomendações mediante o TOP 10 de Garrafas mais e menos caras, e o TOP 5 de Garrafas mais e menos vendidas.

### *Clusters* baseados na *feature* ‘*Product Name’*



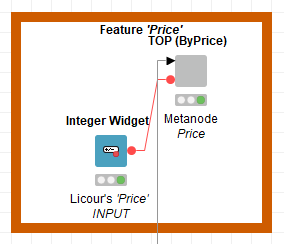
**Figura nº61 – Nodos implementados para o Sistema de Recomendação baseado no Nome do Licor**



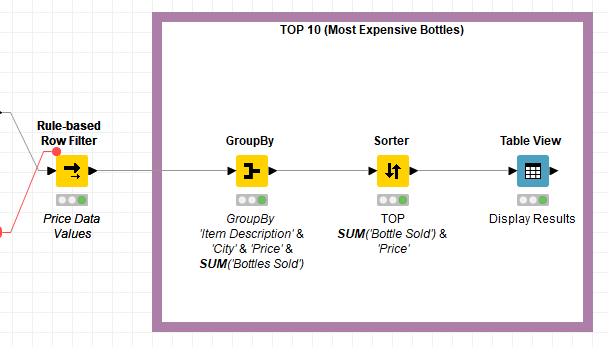
**Figura nº62 – *Workflow* Implementado**

De acordo com as **Figuras nº61 e 62**, e à semelhança das configurações efetuadas anteriormente, podemos observar que recorrendo ao nodo ***Autocomplete Text Widget***, recebemos o *INPUT* do utilizador sobre o Nome de Licores a pesquisar. O valor *default* introduzido é o ‘*Five Star*’. O nodo ***String Manipulation*** e o ***Roled-based Row Filter***, têm o mesmo objetivo dos implementados anteriormente. No caso concreto deste caso, fazemos recomendações mediante o TOP 10 de Garrafas mais e menos caras, e o TOP 5 de Garrafas mais e menos vendidas.

### *Clusters* baseados na *feature* ‘*Price’*



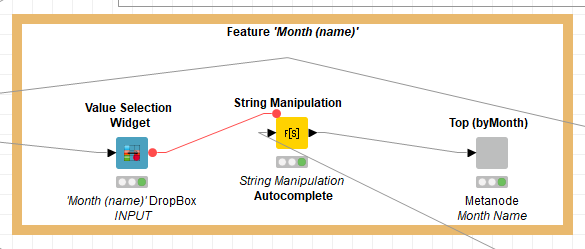
**Figura nº63 – Nodos implementados para o Sistema de Recomendação baseado no Preço**



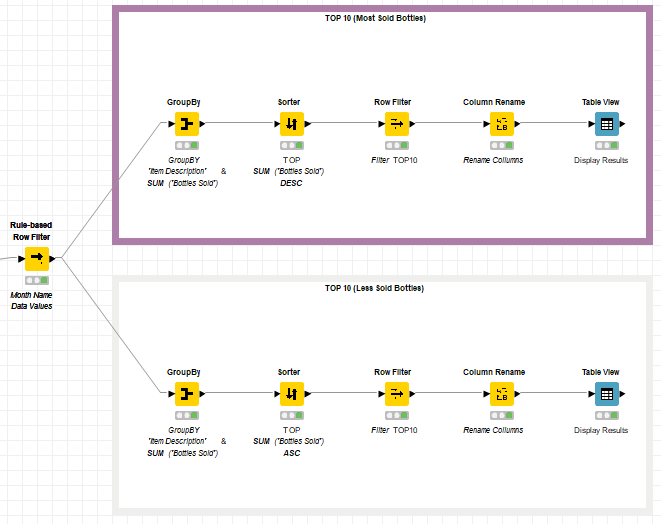
**Figura nº64 – *Workflow* Implementado**

De acordo com as **Figuras nº63 e 64**, e à semelhança das configurações efetuadas anteriormente, podemos observar que recorrendo ao nodo ***Integer Widget***, onde recebemos o *INPUT* do utilizador sobre o Preço de Licores a pesquisar. O valor *default* introduzido é o ’*10 $*’. O ***Roled-based Row Filter***, tem o mesmo objetivo dos implementados anteriormente. No caso concreto deste caso, fazemos recomendações mediante o TOP Preços de acordo com o valor introduzido.

### *Clusters* baseados na *feature* ‘*Month (name)’*



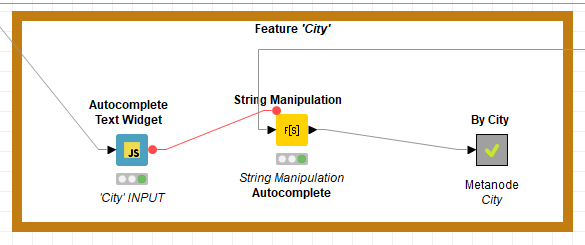
**Figura nº65 – Nodos implementados para o Sistema de Recomendação baseado no Mês de Venda**



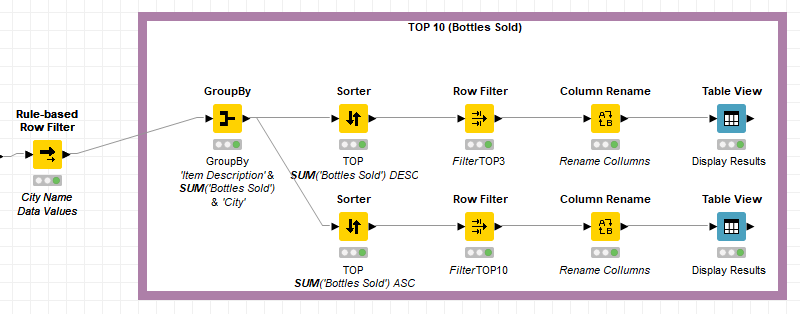
**Figura nº66 – *Workflow* Implementado**

De acordo com as **Figuras nº65 e 66**, recorrendo ao nodo ***Value Selection Widget***, onde recebemos o *INPUT* do utilizador sobre o mês da venda de Licores a pesquisar. O valor *default* introduzido é o ’*Janeiro*’. O ***Roled-based Row Filter***, tem o mesmo objetivo dos implementados anteriormente. No caso concreto deste caso, fazemos recomendações mediante o TOP 10 de Garrafas mais e menos vendidas.

### *Clusters* baseados na *feature* ‘*City’*



**Figura nº67 – Nodos implementados para o Sistema de Recomendação baseado na Cidade de Venda**



**Figura nº68 – *Workflow* Implementado**

De acordo com as **Figuras nº67 e 68**, podemos observar que recorrendo ao nodo ***Autocomplete Text Widget***, recebemos o *INPUT* do utilizador sobre a Cidade de Venda de Licores a pesquisar. O valor *default* introduzido é o ‘*Dubuque*’. No nodo ***String Manipulation***, fazemos um *join* com o que é guardado na variável do *Autocomplete*. Já no nodo ***Roled-based Row Filter***, implementamos a regra que nos permite fazer a associação. No caso concreto deste caso, fazemos recomendações mediante o TOP 10 de Garrafas mais e menos vendidas.

## Avaliação do Sistema de Recomendação

## Regras de Associação

De seguida demonstramos a avaliação do Sistema de Recomendação baseado em Regras Associativas:

|  |  |  |
| --- | --- | --- |
| Categoria Pesquisada | Categorias Recomendadas | Conclusão |
| *Aged Dark Rum* | *Flavored Rum*  *White Rum*  *Spiced Rum* | Podemos observar que quem consome “*Aged Dark Rum*” o sistema recomenda mais 3 tipos de Rums. |
| *Whiskey Liqueur* | *Canadian Whiskies*  *Straight Bourbon Whiskies* | Podemos observar que quem consome “*Whiskey Liqueur*” o sistema recomenda mais 2 tipos de whiskies. |

**Tabela nº3 – Avaliação do Sistema de Recomendação (Categorias de Licores)**

|  |  |  |
| --- | --- | --- |
| Nome Pesquisado | Licores Recomendados | Conclusão |
| *Black Velvet Reserve* | *Black Velvet Traveler*  *Black Velvet*  *Black Velvet Toasted Caramel*  *Black Velvet Mini* | Podemos observar que quem consome “*Black Velvet Reverse*” o sistema recomenda mais 4 tipos da mesma gama. |
| *Five O'Clock Vodka PET* | *Five Star*  *Five O'Clock Gin* | Podemos observar que quem consome “*Five O'Clock Vodka PET*” o sistema recomenda mais 2 tipos da mesma gama. |

**Tabela nº4 – Avaliação do Sistema de Recomendação (Nomes de Licores)**

## Baseado em *Clusters*

|  |  |  |
| --- | --- | --- |
| Nome Pesquisado | Licores Recomendados | Conclusão |
| *Cream Liqueurs* | *Kirkland Signature Irish cream*  *McCormick's Irish cream*  *Baileys Original Irish cream*  *Ryan's cream Liqueur* | Podemos observar que quem consome licores cremosos o sistema recomenda mais 4 tipos da mesma gama. |
| *White Rum* | *Paramount White Rum*  *Barton Rum Light*  *Paramount White Rum PET*  *Bacardi Superior (RUM)* | Podemos observar que quem consome Rum branco o sistema recomenda mais 4 tipos de Rum. |

Claro está que futuramente deveria ser implementado um limite, de modo a que a regra de qualidade seja confiável. Para concluir, de modo geral, as recomendações são bastante coerentes e parecem assertivas.

# **Conclusão**

A elaboração deste Projeto/Trabalho Pratico, cujo objetivo principal recaía sobre a construção de um Sistema de Recomendação, aplicando e explorando todo aquilo que nos foi passado ao longo do semestre, com o propósito de ter um Sistema de Recomendação funcional e fiável.

Durante o desenvolvimento do sistema, foi crucial ir fazendo uma análise dos resultados que íamos obtendo, uma vez que no processo de desenvolvimento de modelos de *Machine Learning* é importante, não só́ saber analisar os dados e obter conclusões sobre estes, mas também, encontrar formas de moldar os dados de modo a obtermos os resultados esperados. Após a “limpeza” dos dados e do *dataset* devidamente tratado, iniciamos a implementação do Sistema de Recomendação. Este processo foi bastante demorado e custoso, tem sido a maior adversidade encontrada na realização deste trabalho, uma vez que residia a dúvida sobre qual ou quais as melhores recomendações baseadas nos *inputs* definidos pelos utilizadores. Porém com todo o emprenho e dedicação tido pelos elementos do grupo, esta dificuldade foi superada com sucesso.

Qualquer projeto prático contém possíveis melhorias a adicionar e este não é exceção, a principal melhoria será a avaliação do k ótimo para a clusterização através do cálculo dos centroides, que infelizmente devida a escassez de tempo não conseguimos implementar. Para concluir, com a realização deste trabalho prático e com o fim deste relatório, consideramos ter atingido todos os objetivos propostos e descrito todo o desenvolvimento da forma mais intuitiva possível.