

**Desenvolvimento de um Agente Inteligente para gerenciamento de estoque de  
supermercado utilizando rede bayesiana**

Sofia A. Rosa

Faculdade de Ciências - Unesp Bauru

Ciência da Computação - Inteligência Artificial

Prof. Dr. Clayton Reginaldo Pereira

Outubro de 2024

### **Resumo**

Neste projeto, um *notebook jupyter* foi construído com o desenvolvimento detalhado de um agente inteligente programado para gerenciar o estoque de um supermercado. No relatório a seguir serão apresentados detalhes sobre a situação fictícia que deu origem à ideia do trabalho, o dataset utilizado e a lógica por trás do agente resultante.

### **A situação**

A “situação do mundo real” fictícia que deu origem ao projeto gira em torno de uma rede de supermercados que tem seu estoque mal gerenciado: mercadorias são perdidas, pois não são vendidas, enquanto outras faltam para os clientes. Portanto, a ideia do agente é gerenciar o estoque do supermercado considerando fatores que possivelmente afetam a demanda de cada produto. Para simular essa situação, foi utilizado um dataset retirado da plataforma Kaggle: [Groceries Dataset for Market Basket Analysis](#), mais especificamente, foi utilizada a segunda versão desse dataset, que contém dados sobre a data de cada compra.

## Método

### O dataset e as bibliotecas utilizadas

Como dito, foi utilizado o dataset *Groceries Dataset for Market Basket Analysis*, disponibilizado por um usuário na comunidade da plataforma *Kaggle*. O dataset não tinha o formato perfeito para analisar especificamente os fatores que poderiam afetar a demanda e as compras dos produtos, por possuir as seguintes colunas: *Member\_number*, *Date*, *itemDescription*, *year*, *day*, *month* e *day\_of\_week*. Porém, dessas colunas, é possível retirar um fator que pode influenciar nas compras das pessoas: a estação do ano. Portanto, foi decidido que seriam analisadas as compras a partir dos fatores dia da semana e estação do ano.

É importante utilizar os dados diretamente de um dataset bem alimentado, pois o agente será programado de forma a tomar decisões a partir da análise **estatística** do histórico de vendas do mercado. Então, não foram adicionadas novas informações fictícias a esse dataset.

Além disso, foram utilizadas as bibliotecas *pandas*, *pgmpy* e *matplotlib*, para a análise e alteração do dataset, montagem da rede bayesiana e para plotagem de gráficos, respectivamente. A rede bayesiana foi construída com o uso de uma biblioteca auxiliar, para facilitar o cálculo das probabilidades, já que foi utilizado um dataset bastante extenso. A biblioteca *matplotlib* foi utilizada somente uma vez, para a construção de um gráfico que mostrava quais situações a rede bayesiana deveria retornar uma demanda alta, para que fosse possível basear os testes após o fim da construção do modelo.

### Desenvolvimento da Rede Bayesiana

Com a utilização da biblioteca *pandas*, o dataset foi reorganizado e, em seguida, utilizado para o cálculo das probabilidades e probabilidades condicionais relacionadas à variável principal do projeto: a variável **demanda**. A ideia é que o agente perceba, por meio da data, que dia da

semana é e em qual estação do ano ele está e, dessa forma, calcule a variável **demand** para cada produto. A variável **demand** é classificada como ‘High’ ou ‘Normal’.

Após a reorganização dos dados, foram calculadas as distribuições de probabilidade para as seguintes variáveis independentes entre si: produto, dia da semana e estação do ano. Essas distribuições de probabilidade foram inseridas num objeto da classe *TabularCPD*, da biblioteca *pgmpy*, e são essenciais para a construção da rede bayesiana.

O produto foi programado como uma variável que afeta a demanda e a demanda será calculada várias vezes, sendo uma para cada produto. Quando o agente utilizar a rede bayesiana, o produto será considerado uma variável “preenchida”, ou seja, com um valor de verdade. Isso foi feito para facilitar o desenvolvimento do agente, e não altera a acurácia das suas respostas.

Por fim, com o uso do dataset reorganizado, foi calculada a distribuição de probabilidade condicional para a variável demanda. A variável demanda depende dos três fatores citados anteriormente, portanto, foi calculada a probabilidade de a demanda ser alta ou normal para cada uma das combinações possíveis desses três fatores. Essa distribuição de probabilidade também é inserida num objeto da classe *TabularCPD*, porém, desta vez, são sinalizados os atributos de evidência, que indicam que essa variável depende de outras variáveis, que devem ser inicializadas em seus próprios *cpds* antes.

Para, enfim, construir o modelo da rede bayesiana, foi utilizada a classe *BayesianNetwork*, onde, simplesmente, são inseridas tuplas de variáveis, onde a segunda inserida depende da primeira, para construir os nós e os segmentos entre eles no grafo direcionado. Para esses nós serem identificados de forma correta, as distribuições de probabilidade construídas são inseridas nesse modelo como parâmetro.

## Desenvolvimento do Agente

StorageAgent
-product_mean_sales -storage -inference
+__init__(product_mean_sales, storage, bayesian_network) +get_demand(product: str, day_of_week: str = None, season: str = None) +get_storage(product: str) +get_mean_sales(product: str) +get_storage_state(product: str) +get_day_of_week(date: datetime = None) +get_season(date: datetime = None) +run(date: datetime = None)

O agente desenvolvido tem a função de, todos os dias e para cada produto, calcular a demanda e verificar se o estoque condiz com a tal. O mecanismo que será adotado pelo agente será: se a demanda for alta e o estoque para um produto estiver abaixo do seu número médio de vendas diárias, o estoque deste produto será recomprado.

Para obter a demanda, o agente utiliza o método de inferência estatística pela rede bayesiana. Isso é feito, também, através da biblioteca *pgmpy*, mais especificamente, pela classe *VariableElimination*.

Obviamente, em seu uso normal, o agente utilizará todas as evidências (dia da semana e estação) para calcular a demanda de cada produto. Porém, com a rede bayesiana, ele poderá, por exemplo, calcular a demanda do produto somente com a evidência do dia da semana, ou somente com a evidência da estação do ano, ou até mesmo sem nenhuma evidência além do produto desejado.

## **Resultados**

### **Resultado da Rede Bayesiana**

A rede bayesiana teve sua inferência testada de maneira simples: utilizando um caso onde é esperada alta demanda. No caso do teste, que também está no *notebook jupyter*, foi utilizada a situação: água engarrafada, numa segunda-feira de outono. Como esperado, a inferência realizada a partir da rede bayesiana resultou em demanda alta com probabilidade de 71%.

### **Resultado do Agente**

O agente foi testado em sua função normal, de calcular a demanda de um produto dado o dia da semana e a estação em que ele está. Ele realizou as atividades como esperado, realizando as compras para aqueles produtos que tinham o estoque menor do que a média de vendas.

### **Discussão**

Futuramente, a rede bayesiana pode ser utilizada e adaptada como um modelo de aprendizagem de máquina simples para a classificação da demanda, utilizando a biblioteca *scikit learn*.



### Referências

**RAHMAN, Rashik.** Groceries dataset for market basket analysis (MBA). Disponível em: <https://www.kaggle.com/datasets/rashikrahmanpritom/groceries-dataset-for-market-basket-analysis/data>. Acesso em: 09 out. 2024.

**PGMPY.** Bayesian Network. Disponível em: <https://pgmpy.org/models/bayesiannetwork.html>. Acesso em: 09 out. 2024.