**APRENDIZADO DE MÁQUINA**

**Prof. Cristiano Carvalho**

**Projeto Final (60 pontos)**

A proposta deste trabalho é aplicar técnicas de aprendizado de máquina e extrair conhecimento de bases de dadospara identificar padrões que possam ser úteis para o apoio à tomada de decisão de algum problema social ou empresarial.

O processo de *Data Science* é iterativo, o que pressupõe o retorno a etapas anteriores do processo para ajustes caso necessário.

# ATIVIDADES

1. Cada grupo deverá ter até 5 alunos.
2. Cada grupo deverá escolher uma base de dados **(de livre escolha)**. Neste trabalho serão dadas algumas ideias de bases de dados disponíveis que podem ser utilizadas.
3. Cada grupo deverá, a partir da base de dados escolhida, aplicar o processo de *Data Science,* para extrair informação relevante sobre um problema diverso.
4. Poderá ser utilizada qualquer ferramenta de aprendizado de máquina***.***
5. **O grupo deverá:**
   1. Identificar e detalhar o problema que será abordado;
      1. Sugere-se que o problema possua escopo bem delimitado.
   2. Compreender os dados e como eles podem ser utilizados para resolver o problema;
   3. Realizar a seleção dos dados relevantes (registros e atributos);
   4. Enriquecer e melhorar os dados;
   5. Preparar os dados de acordo com os algoritmos;
   6. Aplicar algoritmos de *machine learning* de dados (pelo menos 2);
   7. Explicar o motivo do uso de cada técnica;
   8. Fazer análise dos resultados.
6. **Deve ser entregue:**
   1. Tabelas com os dados pré processados;
   2. Arquivos contendo os códigos utilizados;
   3. Documentação contendo:
      1. Análise dos dados;
      2. Relatórios dos algoritmos;
      3. Motivos do uso de cada técnica;
      4. Apresentação e análise dos resultados alcançados.
7. **Data de entrega:** até 15 dias após o término da disciplina.
8. De 6 a 12 páginas (se tiver problemas com essa delimitação basta sinalizar)

**DEFINIÇÃO DE GRUPOS DE CLIENTES E MIX DE PRODUTOS – BASE DE DADOS ONLINE RETAIL II (UCI – ML REPOSITORY)**

**Grupo 7:**

Nomes: Mateus Girardi e Wesley Pedro Scorsatto

Matrículas: 149078 e 149432

**Problema:**

Descobrir/definir grupos de clientes de acordo com seu perfil de consumo e região, analisando suas compras e localização, e descobrir os mixes de produtos mais frequentes.

**Base de Dados:**

**Nome:** Online Retail II UCI

**Fonte:** <https://archive.ics.uci.edu/ml/datasets/Online+Retail+II>

**Disponível no Kaggle pelo link:** <https://www.kaggle.com/datasets/mashlyn/online-retail-ii-uci>

**Responsável/colaborador da base:** Dr. Daqing Chen, Course Director: MSc Data Science. chend@ lsbu.ac.uk, School of Engineering, London South Bank University, London SE1 0AA, UK.

De acordo com o colaborador da base, Dr. Daqing Chen, o conjunto de dados do Online Retail II contém todas as transações que ocorreram em um varejo on-line baseado no Reino Unido, entre 01/12/2009 e 09/12/2011. A empresa vende principalmente presentes exclusivos para todas as ocasiões. Muitos clientes da empresa são atacadistas.

**Atributos da base de dados:**

**Invoice:** Número do pedido, é um número inteiro de 6 dígitos único para cada transação. Se o número do pedido começar com a letra ‘C’ indica que ele foi cancelado.

**StockCode:** Código do produto, é um número inteiro de 6 dígitos único para cada produto.

**Description:** Descrição do produto.

**Quantity:** Quantidade do item por transação.

**InvoiceDate:** Data do pedido.

**UnitPrice:** Preço unitário em libras esterlinas.

**CustomerID:** Código do cliente, é um número inteiro de 5 dígitos único para cada produto.

**Country:** Nome do país que o cliente reside.

Analisando a base de dados também pudemos definir que quando uma transação é cancelada, além do número do pedido possuir a letra ‘C’ como prefixo, as quantidades de cada item são negativas.

**Definição dos grupos de clientes (clusterização):**

Descobrir/definir grupos de clientes de acordo com seu perfil de consumo e região.

**Algoritmo utilizado:**

Algoritmo de clusterização K-means.

**Seleção de atributos relevantes**

Uma imagem contendo Calendário

Descrição gerada automaticamente

Os atributos relevantes para realizar a clusterização dos clientes são os seguintes:

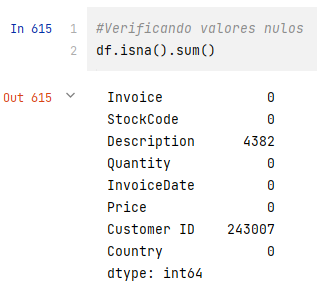
* Necessário para calcular as métricas utilizadas no K-means:
  + Invoice.
  + StockCode.
  + Quantity.
  + Price.
* Realização de agrupamento:
  + Customer ID.
  + Country.

O atributo “Country” é o único atributo que específica algum endereço do cliente, porém, o ideal seria termos o endereço completo dos clientes e obter suas regiões em seus respectivos países, dessa forma obteríamos uma clusterização mais específica e eficaz na criação de grupos de clientes com finalidade comercial.

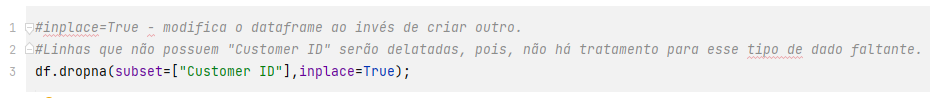
Somente os atributos “InvoiceDate” e “Description” não foram utilizados na clusterização, pois, não tem utilidade neste caso.

**Melhoria e preparação de dados:**

A primeira coisa que verificamos na base de dados é a presença de valores nulos, através do código abaixo.



Conforme o resultado acima, podemos observar que há registros com descrições e códigos de clientes nulos, porém, para clusterização apenas o código do cliente interessa. Nesse caso, em que não há como prever quantos pedidos um cliente realizou e inserir um valor aleatório de código de cliente, pois, prejudicaria muito a análise, decidimos excluir os valores nulos.



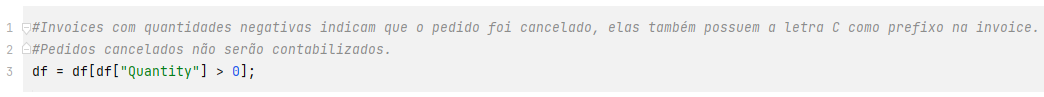
A linha de código acima exclui todos os registros que possuem a coluna “Customer ID” nula e grava o resultado no próprio dataframe.

Por fim, verificamos que não sobrou nenhum registro nulo na base, outra coisa, é que todos os registros que não possuíam descrição também não possuíam o código de cliente e foram deletados juntos.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Depois de excluir os valores nulos, também foi necessário tratar os pedidos cancelados. Conforme descrito na definição da base de dados, pedidos que possuem itens com quantidades negativas, são pedidos que foram cancelados, e para realizar a clusterização precisamos apenas de pedidos que foram verdadeiramente efetivados.



A linha de código acima filtra apenas registros com quantidade maior que zero.

**Criação de features:**

Para conseguir calcular as métricas foi necessário criar a coluna que representa o valor total do item.

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Para criar as features realizamos o agrupamento dos dados pelos campos “Customer ID” e “Country”. Dessa forma, realizamos as operações de agrupamento, como soma, média, contagem e contagem de distintos.

Texto

Descrição gerada automaticamente

Com esse agrupamento realizado podemos calcular as features que serão utilizadas no algoritmo K-means para agrupar os clientes.

Texto

Descrição gerada automaticamente

As features utilizadas no K-means são:

* **Country:** Traz a identificação da região do cliente.
* **Avarage\_Quantity:** Quantidade média comprada por produto.
* **Avarage\_Value:** Valor médio por pedido.
* **Avarage\_Items:** Quantidade média de itens por pedido.
* **Avarage\_Products:** Quantidade de produtos distintos por pedido.

Para o atributo “Country” identificamos registros onde está preenchido com valor indefinido, mas decidimos manter, pois, os clientes possuem as outras feautures para realizar a clusterização.

Importante! Esses valores são agrupados por cliente.

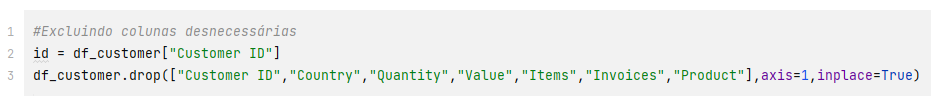
Como o atributo “Country” é uma string com o nome do país e o K-means só aceita valores numéricos, é necessário transformar o campo para inteiro.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

No código acima utilizamos a função “fit” de uma instância de “LabelEncoder” da biblioteca sklearn para gerar um dicionário, onde cada valor textual tem seu valor numérico correspondente e depois utilizamos a função “transform” que retorna esses valores numéricos, gravando na nova coluna “NCountry”.

E ao final, excluímos as colunas que não serão necessárias para o algoritmo, salvando apenas os códigos dos clientes para ligação com os clusters no final.



Com isso a base final está pronta para ser processada pelo KMeans.

Tabela

Descrição gerada automaticamente

**Aplicação do algoritmo:**

Escolhemos o algoritmo K-means para clusterização, pois, ele apresenta algumas vantagens, tais como:

* Fácil implementação.
* Escalonável e rápido para um grande conjunto de dados

Porém, uma de suas desvantagens é necessidade de o usuário informar a quantidade de clusters que quer formar, isso pode ser um pouco difícil de avaliar, por esse motivo, é que uma análise do valor de inércia é realizada para um determinado X números de clusters permitindo tomar uma melhor decisão de quantos clusters utilizar.

A inércia mede o quão bem um conjunto de dados foi agrupado por K-Means. É calculado medindo a distância entre cada ponto de dados e seu centróide, elevando essa distância ao quadrado e somando esses quadrados em um cluster.

Um bom modelo é aquele com baixa inércia e um baixo número de clusters (K). No entanto, isso é uma compensação porque, à medida que K aumenta, a inércia diminui.

Então, uma das formas para encontrar o K ideal em um conjunto de dados, é utilizar o método cotovelo, nele, através do gráfico encontramos o ponto onde a diminuição da inércia começa a diminuir.

Preenchendo os valores de inércia para K-means de 1 até 10 clusters.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Gerando o gráfico.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Analisando o gráfico podemos verificar que ainda temos um ganho de qualidade nos clusters até 5, então, vamos aplicar 5 clusters no algoritmo.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Calculando o K-means.



Gerando a predição (clusterização) com K-means e atribuindo os clusters gerados em uma nova coluna chamada “Cluster”.

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Como resultado temos a definição do cluster para cada cliente.

Tabela

Descrição gerada automaticamente

**Análise dos Resultados Obtidos:**

Como utilizamos 5 features para a clusterização necessitamos aplicar o PCA para converter os dados em 2 dimensões e conseguirmos visualizar o gráfico de dispersão.

Texto, Aplicativo

Descrição gerada automaticamente

**Resultado**

Podemos observar que os grupos 0 e 2 são bem próximos, mas marcam uma boa separação, o grupo 3 é um pouco mais disperso dos outros dois anteriores e os grupos 1 e 4 são totalmente outliers.

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

**Outras Análises**

Texto

Descrição gerada automaticamente com confiança baixa

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

**Definição de Mix de Produtos Vendidos:**

**Algoritmo utilizado:**

Algoritmo de Associação FP-growth (método de geração de padrões frequentes de itens sem a geração de candidatos).

Realizamos a escolha desse algoritmo, pois o mesmo traz resultados de uma maneira muito mais eficiente que o algoritmo Apriori. Além disso, o FP Growth varre o banco de dados apenas duas vezes, já o Apriori varre as transações para cada iteração.

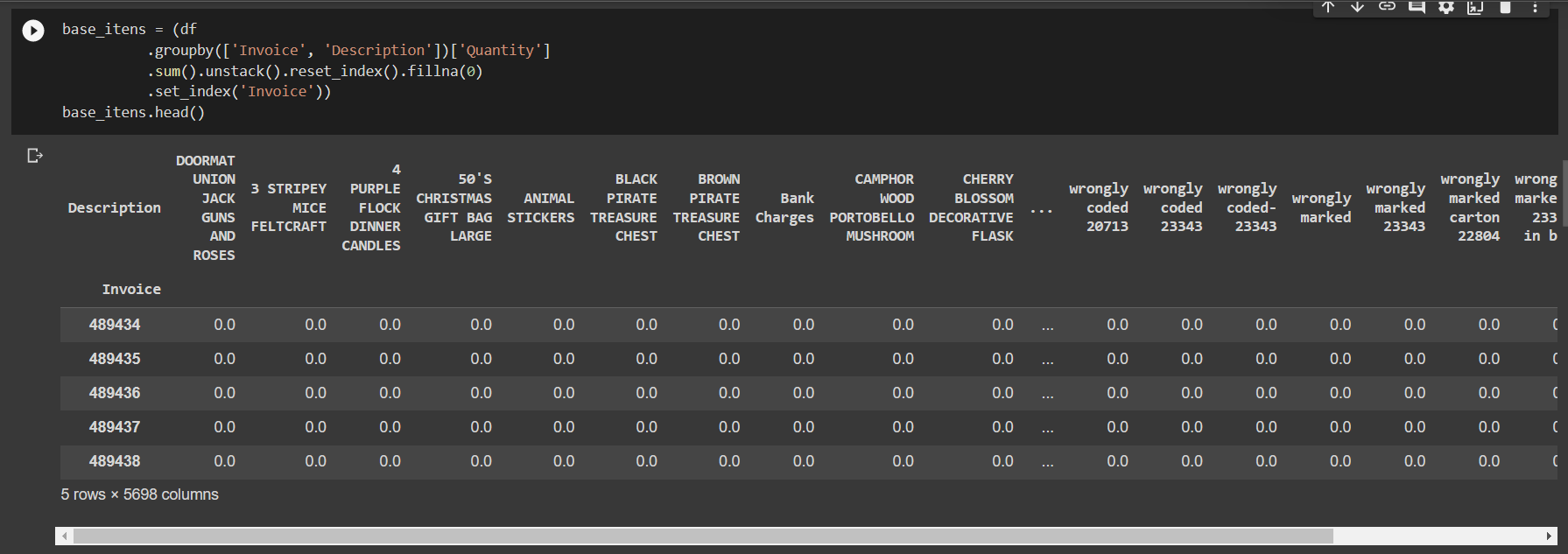
**Seleção de atributos relevantes:**

Na análise inicial realizada na base de dados, identificamos que todos os dados são relevantes, porém os atributos utilizados para a análise do mix de produtos foram as colunas ‘Invoice’, ‘Description’ e ‘Quantity’.

**Melhoria e preparação de dados:**

Para a aplicação do algoritmo, realizamos os seguintes passos de preparação de dados:

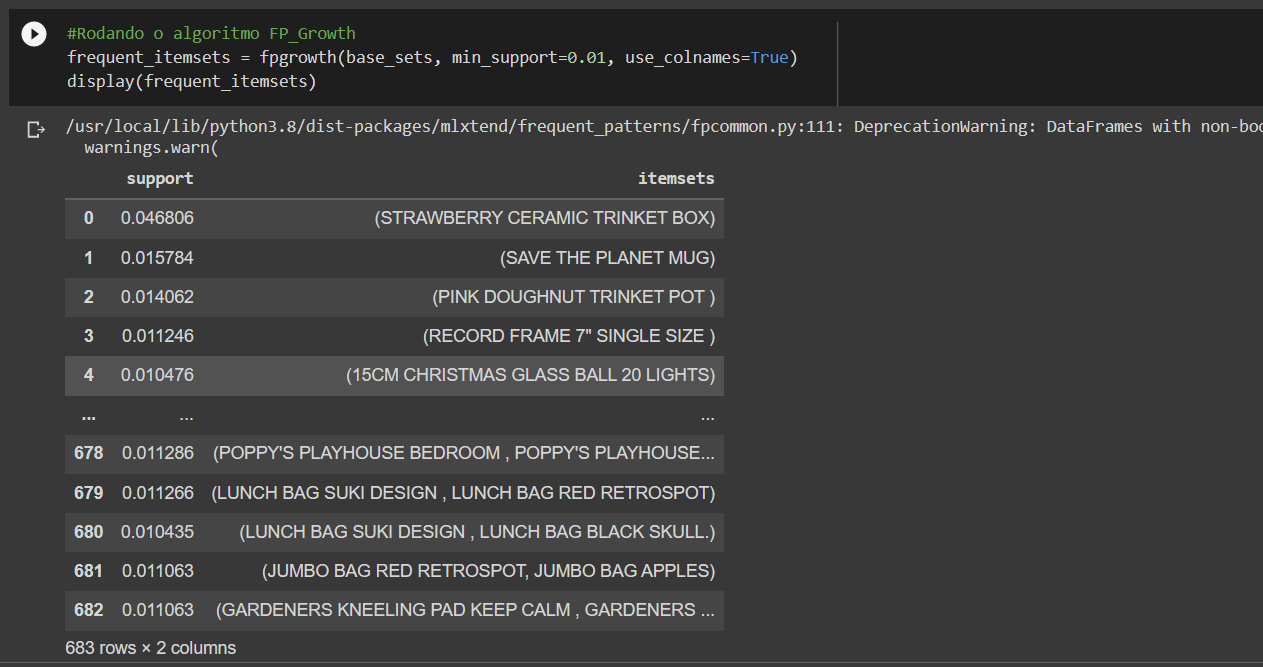
* Transformamos a coluna 'Invoice' de tipo ‘object’, para 'str';
* Geramos uma tabela pivô em que cada coluna corresponde à um produto e cada linha corresponde ao somatório da quantidade comprada daquele produto em um determinado pedido.



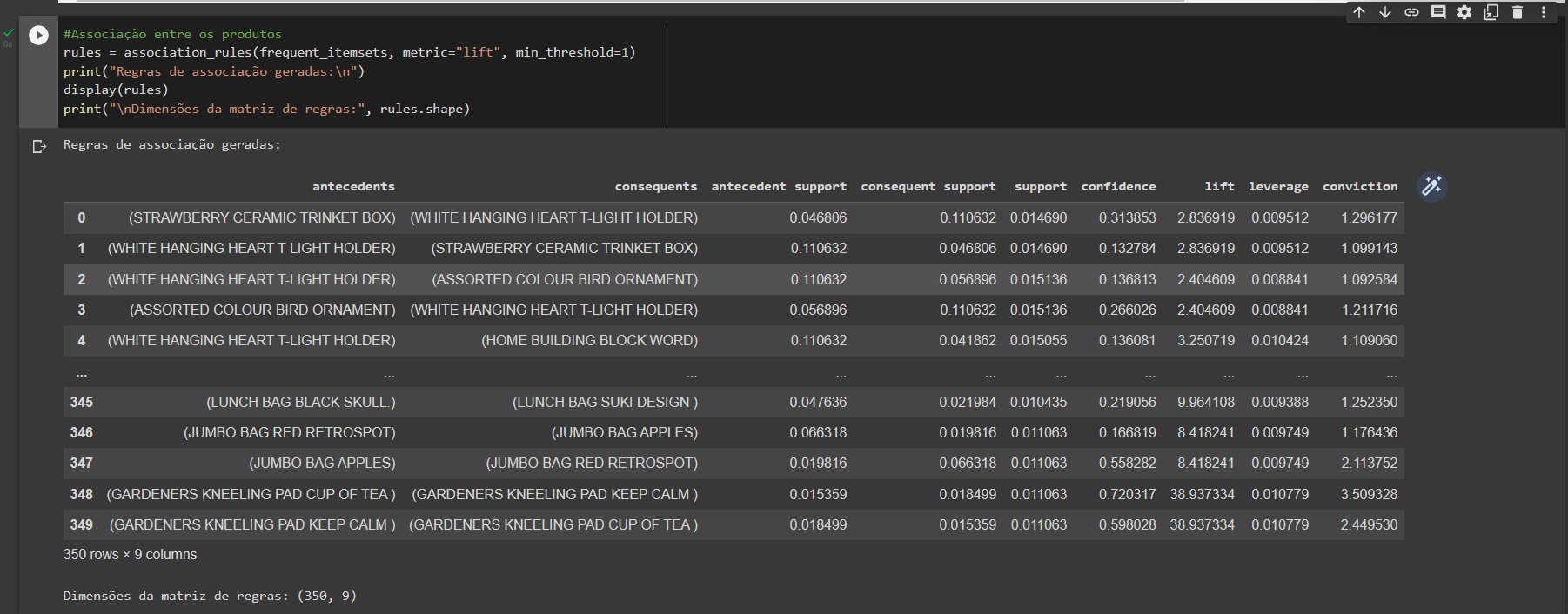
* Após gerar a tabela acima, transformamos as quantidades em 0 ou 1.

**Aplicação do algoritmo:**

Para aplicar o algoritmo FP-growth, executamos a função fpgrowth, da biblioteca mlxtend. Na função, passamos a base de dados tratados e o número mínimo de suporte (0.01 utilizado para o algoritmo). Como resultado, obtivemos o respectivo produto e o sua métrica de suporte.



Após esses resultados, realizamos a função ‘association\_rules’, da mesma biblioteca ‘mlxtend’. Nela passamos a base retornada pelo algoritmo FP-growth e a métrica ser utilizada (Lift). Como resultado, obtivemos os produtos que possuem associação no momento da venda.



**Análise dos Resultados Obtidos:**

Ao aplicar o algoritmo FP-growth com o mínimo de suporte permitido 0,01 (mínimo que trouxe mais resultado), obtivemos 683 linhas de produtos, com a sua respectiva métrica de suporte.

Após a extração desses dados, aplicamos a função ‘association\_rules’ mencionada acima e nela obtivemos 350 associações de produtos, com a sua respectiva métrica Lift.

Além disso, geramos um gráfico das associações geradas:

