**Projeto Final**

1. **Exposição do problema**

Com o passar dos anos e o avanço da tecnologia muitas ferramentas surgem e/ou são aperfeiçoadas e com isso aparece a necessidade de atualizar-se tanto no que diz respeito ao manuseio de tais ferramentas quanto na sua aplicação no mundo coorporativo e dos negócios.

Um ramo científico que vem crescendo com o tempo é o da Ciência de Dados. Esta, juntamente com o processo de Análise de Dados servem como base de apoio para entendimento do *business* em questão e para futura tomada de decisões que venham a impactar (positiva ou negativamente) o futuro de uma empresa, comércio, etc.

Dentre os muitos exemplos sobre aplicação de Ciência de Dados, podemos citar a segmentação de clientes de uma empresa, assunto este que iremos abordar em detalhes neste trabalho. Neste contexto, afim de aumentar as vendas ou verificar o engajamento do público com relação aos serviços oferecidos, é fundamental entender o ponto de vista dos consumidores. O que eles compram, seu nível de satisfação com o serviço oferecido, o total de receita gerado pelas transações, etc. Ao responder estas e outras perguntas é possível traçar um perfil bem preciso dos clientes e posteriormente tomar decisões para fidelizá-los, como oferecer promoções, enviar propaganda ou anúncios específicos, entre outros.

Neste trabalho iremos realizar a citada segmentação de clientes e para isso usaremos um modelo denominado **RFM** (do inglês *recency*, *frequency*, *monetary*) onde nosso objetivo principal é agrupar os clientes de uma empresa de acordo com as variáveis ditas acima. Além disso, exploraremos as compras para tirar informações sobre os departamentos que mais lograram êxito nas vendas. Por fim, iremos criar um modelo para tentar prever quanto dinheiro será ganho nos próximos 30 dias de acordo com vendas futuras.

**Modelo RFM**

Para o processo de segmentação de clientes utilizamos o famoso e consolidado Modelo RFM, que leva em consideração a **recência**, a **frequência** e a **monetaridade** de cada cliente. Temos que as definições destes termos são dadas por:

* Recência: Tempo (em dias) que o cliente não realiza uma compra.
* Frequência: Número de compras realizadas.
* Monetaridade: Quantidade de dinheiro gasto pelo cliente.

Diante deste panorama, identificamos os seguintes pesos de importância para cada uma das variáveis:

* Recência: Quanto menor, melhor (menos dias decorridos desde a última compra).
* Frequência: Quanto maior, melhor (mais vezes o cliente efetuou uma compra).
* Monetaridade: Quanto maior, melhor (mais dinheiro o cliente gastou).

As situações opostas a estas representam um menor peso de importância.

1. **Coleta ou Importação dos dados**

É fato que qualquer empresa, até para próprio controle interno, possui dados históricos acerca de suas operações e estes dados podem (e devem) ser explorados no sentido de trazer *insights* sobre o que realmente está acontecendo e o que se pode fazer a partir daí para seu crescimento.

Para nosso trabalho de segmentação de clientes iremos fazer uso da chamada coleta de dados secundária, que consiste em coletar estes dados já disponibilizado previamente. Neste caso, iremos utilizar uma base de dados pública da Olist, publicada na plataforma de ciência de dados *Kaggle* e que pode ser conferida em: [*https://www.kaggle.com/olistbr/brazilian-ecommerce*](https://www.kaggle.com/olistbr/brazilian-ecommerce).

Esta base de dados consiste em 8 (oito) arquivos no formato *.csv* mas para os propósitos deste trabalho utilizamos apenas 4 (quatro) cujos conteúdos são:

* olist\_customers\_dataset.csv: contém informações dos clientes como seu id único e dados de localização (como cidade, estado e cep);
* olist\_order\_items\_dataset.csv: apresenta informações dos itens que compõem cada ordem de compra do cliente, como id do produto, preço, frete etc;
* olist\_orders\_dataset.csv: possui a ordem de cada compra. Aqui encontramos informações como data da compra, data de aprovação do pedido e data da entrega.
* olist\_products\_dataset.csv: possui descrições do produto como número de fotos no anúncio, peso, dimensões, categoria, entre outros.

A partir destes *datasets* foi possível uni-los em um só para realizar todas a análises necessárias.

Biblioteca(s) utilizada(s) nesta etapa:

* Pandas

1. **Preparação dos dados**

Como era de se esperar, os dados obtidos via Kaggle não vieram perfeitos e nem seque apropriados para sua pronta utilização. Desta forma foi necessária uma grande limpeza nestes dados que incluiu:

* Eliminação de colunas indesejadas.
* Conversão dos tipos das colunas para os formatos adequados (como *datetime* por exemplo).
* Agregação de colunas para obter uma contagem ou soma de seus valores.
* Remoção de outliers

Uma importante ressalva é a de que, embora os *datasets* apresentassem valores *missing* (faltantes), não foi necessário nenhum tipo de tratamento adicional uma vez que tais problemas foram verificados nas colunas que posteriormente foram excluídas do estudo.

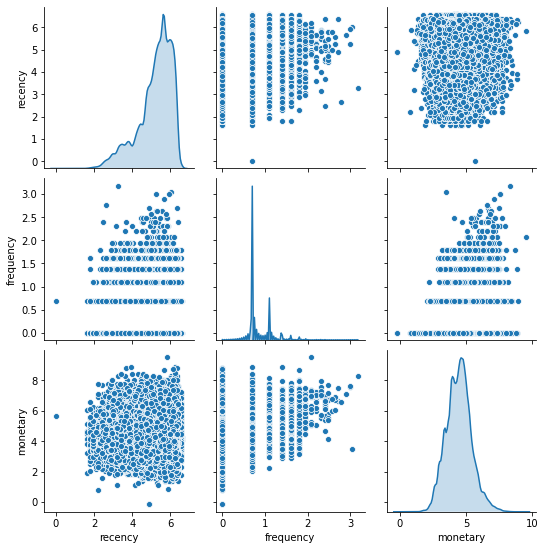
Biblioteca(s) utilizada(s) nesta etapa:

* Pandas
* Timedelta (via datetime)

1. **Análise Exploratória de Dados (AED)**

Após os passos anteriores, obteve-se como resultado um dataframe temporário contendo os dados references à recência, à frequência e à monetaridade, sendo possível fazer uma exploração dos dados.

Como gráfico principal foi definido um *pairplot* onde mostramos a dispersão entre essas variáveis duas a duas, podendo ser visto abaixo:



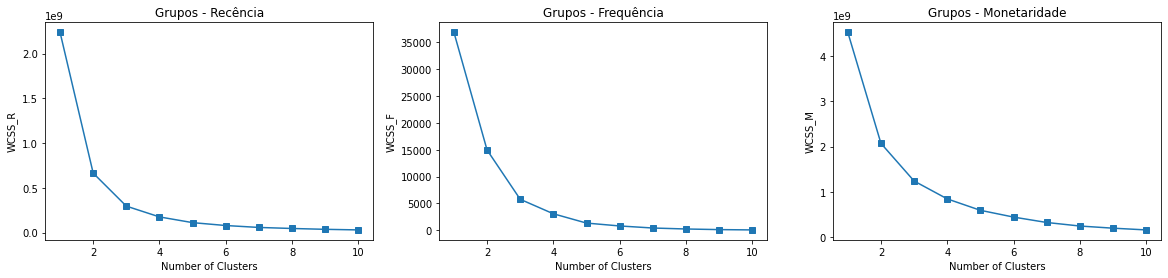
Para obtenção deste gráfico, foi feito inicialmente um escalonamento de forma logarítmica dos dados e este de fato representa um resultado consideravelmente melhor do que ao utilizar os dados originais. Neste caso, é possível notar que as variáveis não apresentam correlação forte entre si e, além disso, vemos que os respectivos histogramas se apresentam bem aproximado a uma distribuição normal, que é importante na a criação do modelo.

* 1. **Elbow Method**

Este método (no português chamado “Método do Cotovelo”) nada mais é do que uma métrica que avalia a melhor escolha para o número de *clusters* (grupos) que se deseja segmentar.

Para encontrarmos então esta quantidade foi utilizado o algoritmo **K-Means** em Python onde foi passado como parâmetro o número de clusters máximo igual a 11 (mais detalhes encontra-se no notebook no repositório do GitHub).

Com resultado desta implementação foi feito um gráfico de cada uma das variáveis e seu respectivo número ideal de grupos. Este plot pode ser visto a seguir.



Assim, ao analisar o resultado dos plots acima foi possível concluir que 5 grupos seria uma boa escolha e, portanto, se adequaria bem ao problema.

Uma observação relevante é o fato de que foram feitas duas abordagens para os plots acima: com os dados originais e com os dados escalonados, o que no final resultou nos mesmos resultados.

* 1. **K-Means**

Ao definirmos o número de grupos nos quais queremos incluir os clientes devemos de fato agrupá-los e para isso usamos novamente o algoritmo K-Means passando agora como parâmetro o número de grupos (neste caso 5).

É importante salientar que ao aplicarmos o K-Means cada grupo é identificado por um número aleatório (que neste caso vai de 0 a 4). Entretanto, queremos aproveitar este número para fazer a pontuação automática de cada cliente a partir dos critérios do RFM vistos na seção 1 (ou seja, pontuação 0 para os “piores” clientes e 4 para os “melhores” clientes). Para rearranjarmos estes labels gerados pelo K-Means com a pontuação de cada cliente baseado no modelo, foi desenvolvida uma função denominada *adjust\_index* que faz automaticamente esta conversão, deixando nossos clientes devidamente pontuados de acordo com seu comportamento em cada variável de interesse.

Biblioteca(s) utilizada(s) nesta etapa:

* K-Means
* Matplotlib
* Nesta etapa nosso objetivo é utilizar uma metodologia aplicando-a no nosso contexto de negócios, que é uma segmentação dos clientes. Para isso, precisamos saber exatamente em quantos grupos podemos agrupar estes clientes no intuito de obtermos um resultado com a maior precisão possível. Logo, usaremos uma metodologia gráfica chamada **Elbow Method**.