HENRIQUE TORRES MENDES

# USO DO MÉTODO DE *RANDOM FOREST* PARA IDENTIFICAR AS CAUSAS DE *CHURN* EM UMA PLATAFORMA DE *E-COMMERCE* DE MATERIAIS DE CONSTRUÇÃO

São Paulo
2021

HENRIQUE TORRES MENDES

# USO DO MÉTODO DE *RANDOM FOREST* PARA IDENTIFICAR AS CAUSAS DE *CHURN* EM UMA PLATAFORMA DE *E-COMMERCE* DE MATERIAIS DE CONSTRUÇÃO

Trabalho de Formatura apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do diploma de Engenheiro de Produção

São Paulo
2021

## AGRADECIMENTOS

A meus pais, Ailton Pereira Mendes e Maria Cileide Torres, e a meu avô, Sebastião Severiano Torres (*in memorian*), cuja presença em minha vida foi fundamental para que eu perpassasse as diversas nuances desta jornada.

A meus amigos, em especial a Bruno Seung Ho Chun e a Suelio Augusto da Silva Araújo, que tornaram milhões de vezes mais leves os meus caminhos mais tortuosos.

A todos os professores da Escola Politécnica da Universidade de São Paulo por todo o aprendizado que embasou minha jornada acadêmica. Um agradecimento especial a meu orientador, Professor Doutor Paulino Graciano Francischini, cujos direcionamentos e paciência foram fundamentais para a conclusão deste trabalho.

## RESUMO

O presente trabalho tem como objetivo utilizar técnicas de *Machine Learning* para identificar as causas de desistência de clientes em uma plataforma de *e-commerce* de materiais de construção civil. A empresa que criou a plataforma, após uma pesquisa qualitativa realizada com os usuários do *marketplace*, percebeu que grande parte destes usuários não recomendaria a plataforma para algum conhecido. Deste modo, a empresa criou uma equipe no começo de 2021 com o objetivo específico de entender os motivos que podem fazer com que um usuário da plataforma desista de comprar neste *marketplace*. Após ajustar a métrica atual que media a desistência na plataforma, a equipe selecionou o método de *Random Forest* para identificar as causas da desistência, pois este método possui uma interpretabilidade mais simples quando comparado com outros métodos de classificação em *Machine Learning* como as Redes Neurais Artificiais e a Regressão Logística. Usando os dados da plataforma disponibilizados pela empresa, os parâmetros do modelo de *Random Forest* foram escolhidos com base na combinação que gerasse o modelo de maior precisão. Assim, o modelo indicou como perfil majoritário de desistência os usuários cuja primeira compra na plataforma foi realizada através de cartão de crédito e não foi finalizada com sucesso. Através deste direcionamento, a empresa foi capaz de entender melhor o perfil do usuário que navega em sua plataforma, além de poupar recursos que estavam alocados em áreas que a empresa considerava como causas da desistência. Assim, ao final deste estudo, a empresa criou uma força-tarefa com recursos realocados para identificar e solucionar os problemas inerentes aos meios de pagamento do *marketplace* e tornar a experiência do usuário na plataforma mais fluida.



**Palavras-chave:** *Machine Learning*; *Random Forest*; desistência.

## ABSTRACT

This work aims to use Machine Learning techniques in order to identify the main causes of user's churn in an e-commerce platform focused on selling construction products. The company that created this platform, after carrying out a qualitative survey with the users of this *marketplace*, noticed that most of these users wouldn't recommend the platform to an acquaintance. Therefore, the company created a team at the beginning of 2021 with the main purpose of understanding the reasons that could make a user churn within the platform. After adjusting the current metric used to measure the marketplace's churn, this team selected the *Random Forest* method to identify the churn causes because this method can be interpreted in a simpler way when compared to other Machine Learning classification methods such as Artificial Neural Networks and Logistic Regression. Using the platform's data provided by the company, the parameters of the Random Forest model were chosen based on the combination which could yield the most accurate model. Thus, the model indicated as the major *churn* profile the users whose first order within the platform was not finished successfully and had credit card selected as its payment method. Given this direction, the company was able to better understand the profile of the user who navigates within its platform, besides saving resources that were allocated in areas which the company considered to be the main causes of this churn. Finally, at the end of this work, the company created a task force with the reallocated resources in order to identify and solve the problems related to payment methods within the marketplace, aiming to make the user's experience smoother.



**Key Words:** Machine Learning; Random Forest; *churn*.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE GRÁFICOS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| AD | Árvore de Decisão |
| CNPJ | Cadastro Nacional da Pessoa Jurídica |
| CPF | Cadastro de Pessoa Física |
| DMI | Decréscimo Médio da Impureza |
| GMV | *Gross Merchandise Value* |
| JSON | *JavaScript Object Notation* |
| LV | Loja Virtual |
| ML | *Machine Learning* |
| NPS | *Net Promoter Score* |
| RF | *Random Forest* |
| RL | Regressão Logística |
| RNA | Redes Neurais Artificiais |

**SUMÁRIO**

# 1. INTRODUÇÃO

Nesta seção, será descrito o contexto geral da empresa dentro da qual o presente trabalho foi desenvolvido, além dos motivos que fomentaram seu desenvolvimento.

## 1.1. CONTEXTO DO *E-COMMERCE* DA CONSTRUÇÃO CIVIL NO BRASIL

De acordo com a 44ª edição do *Webshoppers*, uma pesquisa realizada pela Ebit | Nielsen, no primeiro semestre de 2021, o segmento de *e-commerce* no Brasil cresceu cerca de 31% em termos de receita quando comparado ao mesmo período em 2020. Deste modo, o setor atingiu o valor recorde de cerca de R$ 53 bilhões movimentados nas plataformas *online*.

Segundo Dall'Agnol (2021), no segundo semestre de 2021, o setor de casa e construção civil apresentou uma redução de receita de quase 16% com relação ao mesmo período no ano anterior, tornando-se, assim, o setor com a maior queda quando comparado a outros setores do *e-commerce* brasileiro. Este cenário coloca o setor de construção civil dentro do mercado de *e-commerce* brasileiro em uma situação delicada no período de retomada econômica pós-pandemia, fazendo com que seja de extrema importância que as empresas que atuam neste setor conheçam o comportamento de seus usuários e possam fidelizá-los da melhor forma possível.

## 1.2. CONTEXTO DA EMPRESA E PRODUTOS

A empresa que serviu de base para o desenvolvimento do presente trabalho, por motivos de confidencialidade, receberá a denominação, ao longo de todo o texto, de Empresa X. Tal empresa é usada para fundamentar este trabalho, pois é a organização em que o autor deste trabalho atualmente estagia. A Empresa X é uma organização de tecnologia que atua no setor de construção civil, mais especificamente no ramo do varejo. Em meados de 2018, a organização foi fundada com base no capital privado, a partir de três empresas sócias que representam grandes *players* do setor de construção civil brasileiro. Ao longo deste trabalho, toda ocorrência do termo “sócias” deve remeter às empresas sócias.

A primeira das sócias é a maior produtora e distribuidora de cimento em território nacional, enquanto que as outras duas sócias são grandes nomes nos ramos de tubos, conexões e vigas metálicas. Partindo da ampla base de clientes que as três sócias possuem no Brasil e ainda aproveitando a cadeia de produção e distribuição de seus produtos, a Empresa X tem como objetivo fomentar o varejo da construção civil através da tecnologia. Para cumprir tal objetivo, a Empresa X, atualmente, se vale atualmente dos seguintes produtos tecnológicos que serão detalhados minuciosamente em seções posteriores:

- Programa de fidelidade para lojas de varejo da construção civil;
- Programa de fidelidade para profissionais de obra;
- Plataforma virtual para contratação de serviços de obra;
- Plataforma virtual para comercialização de produtos da construção civil.

## 1.3. DEFINIÇÃO DO PROBLEMA E MOTIVAÇÃO DO TRABALHO

A Empresa X, no início do ano de 2021, realizou uma pesquisa qualitativa com os usuários da plataforma virtual para comercialização de produtos, internamente chamada pela empresa de Loja Virtual (LV). Apesar de, no momento em que a pesquisa foi realizada, a LV viver um momento em que o número de usuários por mês era crescente, os resultados da pesquisa indicaram alta insatisfação do usuário com a plataforma virtual.

A diretoria da Empresa X, portanto, atentando para o fato de a plataforma virtual ter recebido avaliações negativas em sua pesquisa, além de ter notado que muitos comentários nas avaliações remetiam a usuários que diziam nunca mais usar a plataforma pela experiência ruim, propôs a criação de uma equipe para **identificar as causas que mais impactam na desistência dos usuários na plataforma virtual**.

## 1.4. IMPORTÂNCIA DO PRESENTE TRABALHO

Este trabalho tem como objetivo utilizar técnicas de *Machine Learning* para identificar as principais causas da desistência dos usuários de uma plataforma de *e-commerce* do setor de varejo da construção civil. Além de prever as causas da desistência, o trabalho atual busca utilizar métricas que traduzem o real

comportamento da base de usuários da Empresa X. Assim, a empresa poderá obter mais conhecimento sobre o comportamento de seu usuário.

# 2. REVISÃO BIBLIOGRÁFICA

## 2.1. *MACHINE LEARNING*

Segundo Bonaccorso (2017, p. 9), “aprendizado” é a capacidade de se mudar e evoluir de acordo com estímulos externos de modo a se recordar das experiências prévias. Deste modo, *Machine Learning* (ML), é a parte da Ciência da Computação cujo objetivo primário é estudar modelos matemáticos e aplicá-los a rotinas de programação a fim de prever algum evento ou tomar algum tipo de decisão sem o conhecimento total do contexto no qual estas decisões se situam.

Bonaccorso (2017, p. 10) contextualiza que a base para a definição de modelos de ML são os dados, ou seja, características conhecidas associadas ao problema que se deseja prever que servirão como entrada para os algoritmos matemáticos aplicados à linguagem de programação. Existem diferentes métodos usados para alimentar um modelo de ML com dados dentre os quais é possível citar o **aprendizado supervisionado**.

Este tipo de aprendizado presume que os dados iniciais, comumente chamados de variáveis de entrada, serão divididos em dois grupos menores chamados de **dados de treino** e **dados de teste**. O primeiro grupo será usado pelo modelo de ML para que sejam definidos os parâmetros da função matemática usada para prever o resultado. O segundo grupo, por sua vez, será fornecido ao modelo de ML com os parâmetros já definidos a fim de que seu resultado seja previsto pelo modelo. Como os dados de teste, visto que são conhecidos, já possuem seu resultado pré-definido, o resultado de sua previsão pelo modelo de ML deve ser comparado com o resultado já conhecido a fim de que seja determinada a **precisão** do modelo de ML. (BONACCORSO, p.11)

De acordo com Dreiseitl e Ohno-Machado (2002), Árvore de Decisão (AD), Regressão Logística e Redes Neurais Artificiais (RNA) são algoritmos de ML comumente presentes na literatura acadêmica. A diferença principal entre os três métodos é que o segundo e o terceiro geram como produto uma função matemática com parâmetros definidos enquanto que o primeiro não o faz. Deste modo, as Redes Neurais Artificiais e a Regressão Logística geralmente produzem modelos de ML com **alta precisão** por conta do grande número de parâmetros que podem existir na função matemática gerada pelo modelo. Entretanto, a **alta precisão** destes modelos

implica muitas vezes em **baixa interpretabilidade** dos resultados, isto é, apesar de os modelos conseguirem prever muito bem um cenário real, não se sabe quais são as variáveis de entrada que mais impactam naquele resultado. Assim, o algoritmo de árvores de decisão, por possuir uma lógica mais simples, apresenta alta interpretabilidade dos resultados em detrimento de, geralmente, baixa precisão quando comparado com os outros dois modelos.

Segundo Morettin e Bussab (2004, p.132), em um modelo matemático, uma variável aleatória quantitativa discreta é definida como uma variável que pode assumir valores numéricos bem definidos, cada um também com uma probabilidade de ocorrência bem definida. Morettin e Bussab (2004, p.163) também definem uma variável aleatória contínua como sendo uma variável que assume valores dentro de um intervalo de números reais, o que permite que seus valores de saída possam ser definidos por intervalos que envolvem o real valor observado.

Uma variável qualitativa representa valores que não podem ser resumidos por um número. Geralmente, estes valores, em um modelo matemático de ML podem representar nomes, estados ou outros valores. Este tipo de variável se divide entre variáveis qualitativas ordinais e variáveis qualitativas nominais. O primeiro tipo representa um cenário em que as variáveis podem ser ordenadas enquanto que o segundo tipo exprime um cenário em que não há ordenação das variáveis. (MORETTIN; BUSSAB, 2004, p.10)

## 2.2. ÁRVORES DE DECISÃO

No contexto de *Machine Learning*, as árvores de decisão são métodos que buscam classificar dados segundo uma variável de saída desconhecida a partir de um histórico de dados com características já conhecidas. De acordo com Maimon e Rokach (2010, p.149), uma AD é composta por:

- **uma raiz:** ponto de partida da tomada de decisão. A raiz não tem origem em nenhuma decisão, sendo, portanto, a origem de todas as outras decisões que serão tomadas na AD;
- **nós internos:** são pontos de tomada de decisão presentes em uma AD após a criação da raiz. Os nós internos são pontos que segregam (ou classificam) a amostra de dados original em dois ou mais outros grupos;

- **folhas:** representam os pontos finais da AD e contêm os valores das variáveis de saída do grupo de dados que originou a análise.Em caso de teste ou predição de dados com base em aprendizado prévio, é nas folhas que serão encontradas as variáveis de saída.

Ainda de acordo com Maimon e Rokach (2010, p 151), o objetivo primário de uma AD é, a partir de uma amostra de *N* dados com *k* atributos, obter a classificação ótima da amostra através do menor erro de generalização. Entende-se por erro de generalização, ou ***overfitting***, um contexto das AD's dentro do qual, ainda supondo *N* dados com *k* atributos, uma AD gera exatamente *N* folhas, ou seja, sua classificação foi 100% específica e nada genérica. Uma AD que apresenta *overfitting* muitas vezes também é chamada de árvore *profunda*.

### 2.2.1. Método do ganho de informação por entropia

Existem alguns critérios dentro da literatura acadêmica de AD's que permitem que a classificação ótima seja alcançada pelas árvores de decisão. Dois critérios, entre os critérios mais utilizados, são o **ganho de informação por entropia** e o **índice de Gini**. (MAIMON; ROKACH, 2010, p. 153)

Maimon e Rokach (2010, p.151 - 155) propõem que o ganho de informação é um critério que busca otimizar a classificação de uma AD através do conceito de entropia. Cada nó, dentro de uma AD, realiza uma segmentação da amostra de dados e, para cada divisão, é possível calcular a entropia do conjunto. Supondo um nó que segmenta uma amostra de *N* dados em dois grupos, grupo A e grupo B, a entropia deste nó pode ser calculada segundo a fórmula a seguir:

$$H(s) = -p_{(A)}.log_2(p_{(A)}) - p_{(B)}.log_2(p_{(B)}),\ com\ p_{(A)} = \frac{m}{N}\ e\ p_{(B)} = \frac{n}{N}$$

Onde, *H(s)* é a entropia associada ao dado nó que dividiu a amostra de *N* dados em *m* elementos no grupo A e *n* elementos no grupo B. Cabe ressaltar que foi suposto um nó que divide uma amostra em apenas dois grupos, mas há casos, se necessário, em que a AD divide os dados iniciais em *j* grupos de modo que a entropia mais genérica passa a ser dada pela fórmula:

$$H(s) = -\sum_{i=1}^{j} p_{(i)} . log_2(p_{(i)})$$

Maimon e Rokach (2010, p.155) apresentam que o método do ganho de informação baseado na entropia, propõe, em seguida, que a AD seja analisada de modo que todos os nós tenham sua entropia calculada. O ganho de informação de uma AD é dado então pela fórmula:

$$I = H_o(s) - \sum_{i=1}^{j} p_i H_i(s)$$

Onde, *I* é o ganho de informação de uma AD genérica, $H_o$ é a entropia do nó raiz que divide a amostra de *N* elementos em *j* nós internos, $H_i$ é a entropia de cada um dos *j* nós internos e $p_i$ é igual ao número de dados que saem do nó interno dividido pelo número de dados inicial no nó raiz.

Por fim, o método do ganho de informação define a AD ótima como sendo a AD cuja composição de nós raiz e internos maximiza o valor do ganho de informação. Assim, o método propõe que de modo iterativo, os nós sejam permutados a fim de se obter o valor ótimo. (MAIMON; ROKACH, 2010, p. 155)

### 2.2.2. Método do ganho de informação pelo índice de Gini

De maneira semelhante à entropia, Maimon e Rokach (2010, p.154) definem o índice de Gini ou *impureza* de Gini como sendo:

$$Gini(s) = 1 - \sum_{i=1}^{j} (p_j)^2 \text{ , } com\ p_j = \frac{m_j}{N}$$

Onde, *Gini(s)* é o índice de impureza de Gini de um nó que dividiu a amostra de *N* dados em *j* grupos de modo que cada grupo contenha $m_j$ elementos. De maneira análoga ao ganho de informação baseado na entropia, Maimon e Rokach (2010, p.155) estabelecem o ganho de informação de Gini de acordo com a fórmula:

$$I_{Gini} = Gini_o(s) - \sum_{i=1}^{j} p_i Gini_i(s)$$

Em que $I_{Gini}$ é o ganho de informação de Gini, $Gini_o$ é o índice de Gini do nó raiz que divide a amostra de *N* elementos em *j* nós internos, $Gini_i$ é a entropia de cada um dos *j* nós internos e $p_i$ é igual ao número de dados que saem do nó interno dividido pelo número de dados inicial no nó raiz.

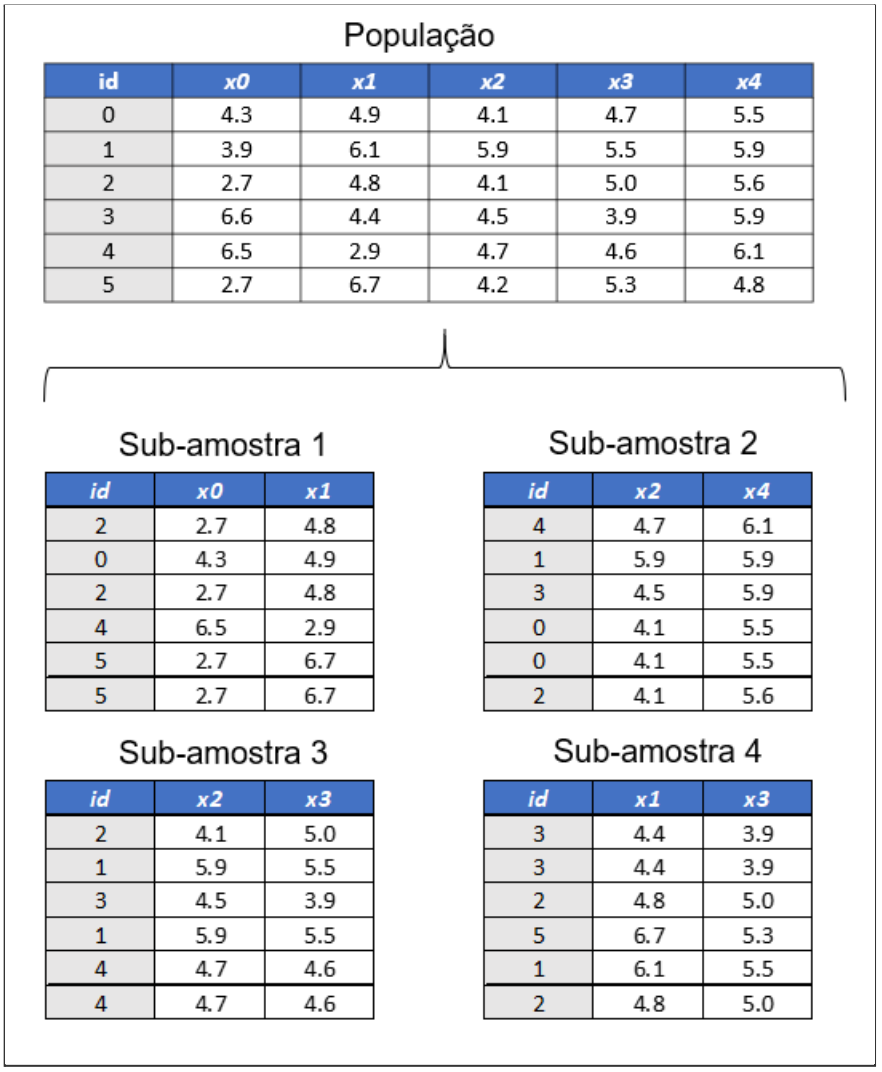
O método do ganho de informação baseado no índice de Gini propõe também que os nós sejam permutados de modo que a AD ótima seria a AD que apresente maior valor de $I_{Gini}$. Ambos os métodos são semelhantes, mas sua diferença básica reside no fato de que o método do índice de Gini quando comparado ao método da entropia, por não possuir em suas fórmulas a função logarítmica, é mais eficiente em termos computacionais, sendo usado muitas vezes nos programas de inteligência artificial. (MAIMON; ROKACH, 2010, p. 155)

## 2.3. *RANDOM FOREST*

De acordo com Maimon e Rokach (2010, p.226) uma das ferramentas mais usadas em *Machine Learning* a fim de se evitar o erro de generalização, ou *overfitting*, nas AD's é a combinação das técnicas de amostragem *bootstrap* e *aggregation*, que, quando realizadas em conjunto, recebem o nome de *bagging*. Particularmente, a técnica de amostragem chamada de *bootstrap* consiste em criar, a partir de uma população inicial com *D* dados e *M* características, sub-amostras compostas de *d* dados e *m* características aleatoriamente selecionadas com reposição com $d < D$ e $m < M$.

A Figura 1 representa um exemplo de *bootsrap*, onde uma população com 6 dados e 5 características (*x0, x1, x2, x3 e x4*) é dividida em 4 sub-amostras, cada uma com 6 dados escolhidos a partir da primeira coluna da população **com reposição** e 2 características distintas escolhidas entre as 5 características da população.

Figura 1 - Exemplo de *bootstrap* aplicado a uma população




População

| id | x0 | x1 | x2 | x3 | x4 |
|---|---|---|---|---|---|
| 0 | 4.3 | 4.9 | 4.1 | 4.7 | 5.5 |
| 1 | 3.9 | 6.1 | 5.9 | 5.5 | 5.9 |
| 2 | 2.7 | 4.8 | 4.1 | 5.0 | 5.6 |
| 3 | 6.6 | 4.4 | 4.5 | 3.9 | 5.9 |
| 4 | 6.5 | 2.9 | 4.7 | 4.6 | 6.1 |
| 5 | 2.7 | 6.7 | 4.2 | 5.3 | 4.8 |

Sub-amostra 1

| id | x0 | x1 |
|---|---|---|
| 2 | 2.7 | 4.8 |
| 0 | 4.3 | 4.9 |
| 2 | 2.7 | 4.8 |
| 4 | 6.5 | 2.9 |
| 5 | 2.7 | 6.7 |
| 5 | 2.7 | 6.7 |

Sub-amostra 2

| id | x2 | x4 |
|---|---|---|
| 4 | 4.7 | 6.1 |
| 1 | 5.9 | 5.9 |
| 3 | 4.5 | 5.9 |
| 0 | 4.1 | 5.5 |
| 0 | 4.1 | 5.5 |
| 2 | 4.1 | 5.6 |

Sub-amostra 3

| id | x2 | x3 |
|---|---|---|
| 2 | 4.1 | 5.0 |
| 1 | 5.9 | 5.5 |
| 3 | 4.5 | 3.9 |
| 1 | 5.9 | 5.5 |
| 4 | 4.7 | 4.6 |
| 4 | 4.7 | 4.6 |

Sub-amostra 4

| id | x1 | x3 |
|---|---|---|
| 3 | 4.4 | 3.9 |
| 3 | 4.4 | 3.9 |
| 2 | 4.8 | 5.0 |
| 5 | 6.7 | 5.3 |
| 1 | 6.1 | 5.5 |
| 2 | 4.8 | 5.0 |

Fonte: Elaborada pelo autor

Quando as sub-amostras criadas pela amostragem *bootstrap* são usadas para que seja inferido o comportamento médio da população, ocorre o evento chamado de agregação, ou *aggregation*. Assim, ao usar em conjunto a técnica de *boostrap* com a técnica de agregação, é possível, ao inferir o comportamento médio da população de dados, diminuir o erro de generalização, o que, quando se trata de AD's, possibilita a criação de AD's menos profundas que conseguem classificar a amostra de dados de maneira mais genérica. (MAIMON; ROKACH, 2010, p. 227)

O método de *Random Forest* (RF), segundo Breiman (2001, p. 2), consiste em usar o conceito de *bagging* e, para cada sub-amostra criada, construir suas respectivas AD's ótimas, ou seja, com o maior ganho de informação. Como o método pressupõe a criação de várias árvores de decisão, seu nome é associado ao termo "floresta". Cabe ressaltar que, como em cada sub-amostra o número de

características é menor do que o número de características da população, as AD's criadas terão profundidades menores do que a AD construída com base na população. A partir das várias AD's criadas, pode-se, no caso de problemas de classificação, escolher as características mais populares, isto é, que estão presentes na maioria das AD's. Estas características mais populares, pela inferência garantida pelo *baggin* podem ser tomadas como as características da população original que melhor classificam os dados, gerando o **menor erro de generalização possível**.

Maimon e Rokach (2010, p.226) também apontam o método de *Random Forest* como uma ferramenta de *Machine Learning* útil para realizar previsões. Assim, as várias AD's criadas a partir das sub-amostras, podem ser usadas para classificar dados que não estavam presentes na população original de acordo com uma variável de saída. A classificação gerada pela maioria das AD's será a classificação de previsão dos dados cuja variável de saída deseja-se prever.

### 2.3.1. Importância de uma *feature* e seus vieses

De acordo com Breiman (2001), o método de *Random Forest*, tanto para classificação de uma amostra quanto para predição de sua característica, tem seu mecanismo baseado em reduzir o viés e os erros de generalização. Por isso, este método parte do processo de *bagging*. Um dos usos mais recorrentes do RF é a classificação de uma amostra segundo uma variável de saída. Nestes casos, os *inputs* para o modelo são as características iniciais da amostra segundo as quais deseja-se construir as AD para a classificação. Estes *inputs* são, no contexto de *Machine Learning*, chamados de *features*.

Strobl et al. (2007, p.3) evidenciam que, dado um modelo de RF já treinado a partir de um aprendizado supervisionado, é possível medir a **importância** de uma *feature* para a classificação do modelo através das AD's que compõem este modelo.

Um dos métodos utilizados para definir a importância de uma *feature* é o método de Decréscimo Médio da Impureza (DMI). Assim, supondo um modelo de RF com *M features* para o qual foram criadas *N* árvores de decisão, a **importância** da *feature* $M_i$ , segundo o DMI é calculada pelas seguintes etapas:

I. Entre todas as *N* árvores que compõem o modelo, identificar as *n* AD's que, após o processo de *baggin*, contêm a feature $M_i$. Cabe ressaltar que $n < N$;

II. Para cada uma das *n* AD's, identificar quais os nós que utilizam a *feature* $M_i$ como categorização e calcular a *impureza de Gini* antes e após o nó;
III. Definir, para cada uma das *n* AD's, a variação $\Delta G$ como sendo a diferença entre a impureza de Gini depois e antes dos nós que contêm a *feature* $M_i$;
IV. Calcular a média entre todos os $\Delta G$ de todas as *n* árvores e atribuir a esta média o nome de Decréscimo Médio da Impureza.

Ainda de acordo com Strobl et al. (2007, p.3), uma *feature* **importante** é aquela que reduz ao máximo a impureza de Gini ao longo da AD. Deste modo, em um modelo de RF, as *features* com os maiores valores para DMI serão as *features* mais importantes do modelo. Em outras palavras, uma *feature* pouco importante é aquela que, se retirada do modelo, pouco impactaria na classificação da amostra inicial que serviu de entrada para o modelo.

Um ponto de atenção ao uso do método de DMI para definição da importância das *features* é que este método tende a dar mais importância (viés) para as *features* que representam variáveis contínuas, isto é, são compostas por dados não discretos. Isto acontece pois as variáveis contínuas tendem a aparecer mais de uma vez nas AD's do modelo de RF por conta de sua infinidade de valores. Assim, é sugerido que, em um modelo de ML baseado em RF, caso seja desejado encontrar a importância das *features* pelo método de DMI, as variáveis contínuas sejam tratadas para que se tornem variáveis discretas (ou categóricas) com poucas categorias. A este processo dá-se o nome de **redução de cardinalidade**. (Strobe et.al, p. 3-7)

## 2.4. REDES NEURAIS ARTIFICIAIS

Segundo Resende, Coimbra e de Paula (2018, p. 47) uma Rede Neural Artificial (RNA) é uma forma de aplicação de *Machine Learning* que possui como princípio a simulação de um córtex cerebral do cérebro humano, onde ocorre o processamento de informações interconectadas por elementos chamados neurônios artificiais (unidades).

Mitchell (1997, p. 82) afirma que:

> "O estudo de redes neurais artificiais foi inspirado, em parte, pela observação de que sistemas de aprendizado biológicos são compostos de muitas redes complexas de neurônios interconectados. [...] redes neurais

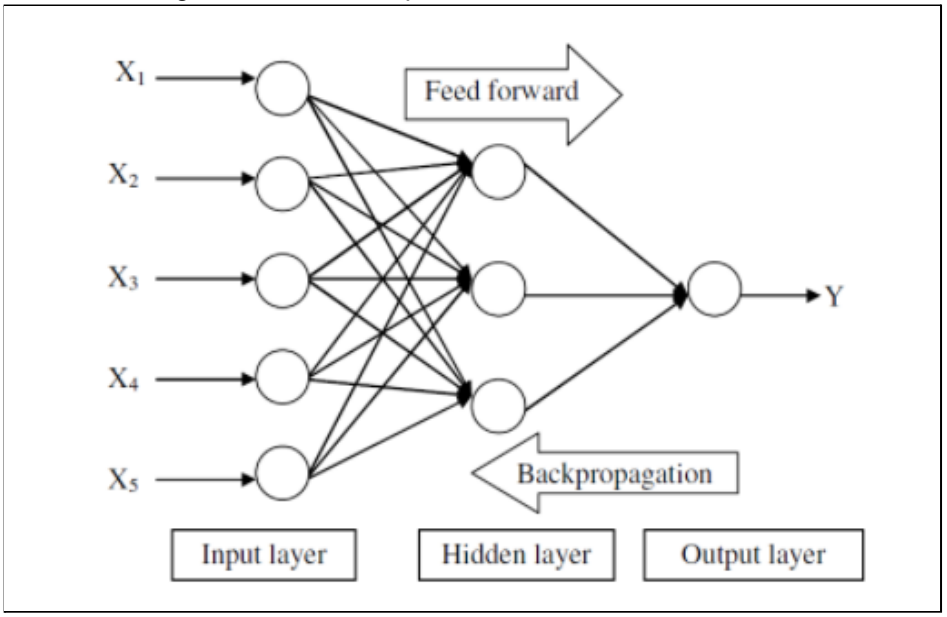
> artificiais são construídas através de um conjunto de unidades simples densamente conectadas, onde cada unidade recebe um único dado real de entrada e, com base neste dado de entrada, produz um único valor real de saída".

As redes neurais artificiais possuem a habilidade de aprender pela experiência adquirida para melhorar sua performance e se adaptarem a mudanças no meio. Deste modo, podem ser usadas não somente em cenários de aprendizagem supervisionada, mas também em casos de aprendizagem não-supervisionada. Assim, este algoritmo de ML é muitas vezes utilizado para realizar previsões sobre contextos dos quais pouco se conhece. (JAMALALDIN et al., 2011, p. 976).

A base do algoritmo de RNA em aprendizado supervisionado é, através das variáveis de entrada, transformar os dados de entrada de acordo com funções matemáticas definidas pelo próprio modelo que os transformarão na variável de saída. A aplicação destas funções aos dados de entrada ocorre na camada escondida do modelo, que possui este nome pois, dependendo da complexidade do problema, pode englobar diversas funções matemáticas de transformação. (JAMALALDIN et al., 2011, p. 976 - 978)

Assim que a variável de saída for determinada, o valor previsto é comparado com o valor esperado e, assim, são determinadas quais transformações presentes na camada escondida são mais úteis ao modelo e quais não o são. Esta comparação nas RNA's do valor previsto com o valor esperado é chamada de *backpropagation*. Assim, itera-se para cada dado presente na base de dados teste e é definida uma RNA ótima para o caso em estudo. Este processo é ilustrado pela Figura 2. (JAMALALDIN et al., 2011, p. 976 - 978)

Figura 2 - Estrutura típica de uma rede neural artificial




Fonte: JAMALALDIN et al., 2011, p. 976

## 2.5. REGRESSÃO LOGÍSTICA

Gonzalez (2018, p.15) define a Regressão Logística (RL) como sendo uma ferramenta de ML cujo objetivo é produzir, a partir de um conjunto de observações de entrada, uma função matemática que permita a predição de valores tomados por uma variável categórica, frequentemente binária, em função de uma ou mais variáveis independentes contínuas e/ou binárias.

Assim, a função matemática obtida será formada por parâmetros numéricos que permitem determinar a probabilidade de um evento ocorrer dependendo dos valores das variáveis independentes de entrada, bem como definir o efeito do conjunto de variáveis de entrada sobre a variável de saída. Dependendo do contexto dentro do qual a Regressão Logística é aplicada, a função matemática pode ser complexa o suficiente a ponto de possuir um grande número de parâmetros, o que faz com que este tipo de modelo, muito embora possua alto grau de precisão, gere cenários pouco interpretáveis. (GONZALEZ, 2018, p. 15-16)

Figueira (2006, p.68) define que existem diversos tipos de regressão logística como:

- **Regressão logística binária**: tipo de regressão em que a variável de saída possui apenas duas categorias, ou seja, tem natureza dicotômica, e apenas uma variável independente de entrada envolvida;

- **Regressão logística múltipla**: tipo de regressão que recebe como variáveis de entrada dados com mais de uma características que potencialmente impactam na variável de saída;
- **Regressão logística multinomial**: modelo de regressão logística em que a variável de saída deve ser qualitativa nominal e distribuída em três ou mais categorias.

## 2.6. *NET PROMOTER SCORE*

De acordo com Grisaffe (2007), o conceito de Net Promoter Score (NPS) foi introduzido ao mundo das organizações no ano de 2003 por um renomado palestrante da Harvard Business School. O NPS metrifica o quanto um usuário de uma determinada marca recomenda tal marca a um conhecido. Este questionário oferece uma escala de números inteiros entre 0 e 10 para que o usuário escolha o quanto recomendaria a marca. Caso a escolha do usuário esteja compreendida entre 0 e 6, o usuário é considerado um usuário detrator. Entretanto, se a nota dada pelo usuário for 7 ou 8, considera-se o usuário neutro. Caso a nota seja 9 ou 10, o usuário é definido como promotor. A métrica NPS é, então, definida pela seguinte fórmula:

$$NPS = \frac{N_p - N_d}{N_p + N_n + N_d}$$

Onde $N_p$ é o número de promotores, $N_d$ é o número de detratores e $N_n$ é o número de usuários neutros associados à pesquisa.

## 2.7. TAXA DE DESISTÊNCIA OU *CHURN RATE*

Segundo Christoff (2020), *churn* é o termo usado para descrever o número de clientes que param de usar um produto após certo tempo de uso. Empresas empenham muito tempo e dinheiro na obtenção de novos clientes através de jogadas de *marketing* e estudos de mercado. Desta forma, é de vital importância que os clientes sejam fidelizados e sintam-se inclinados a continuar comprando e consumindo o produto de determinada empresa. O corte desse relacionamento é

prejudicial à empresa uma vez que mais recursos teriam que ser alocados para que novos clientes fossem obtidos. Assim, é mais viável a manutenção do ambiente para que os usuários que já utilizam o produto queiram continuar usufruindo dos serviços da empresa e o *churn* seja reduzido ao máximo.

Além disso, pode-se dizer que:

> "*Churn* é o contrário de crescimento. Perder clientes implica em sérios impactos na performance geral de uma empresa. Mais especificamente, significa perda em vendas e receita, além de transmitir uma visão negativa da imagem da organização na visão dos concorrentes. [...] A troca de fornecedores por um cliente é descrita como *Churn Rate* e é um dos problemas mais desafiadores e críticos enfrentados pelas companhias durante os últimos anos. O *Churn* pode ser descrito como um KPI (Indicador de Desempenho) que fornece o número de clientes perdidos durante um período de tempo específico dividido pelo número total médio de clientes durante o mesmo período de tempo" (Katelaris e Themistocleous, 2017).

Ainda de acordo com Christoff (2020), são diversas as causas de *churn*. Entre elas podem-se citar:

- **Fatores pessoais**: a perda de um cliente não está necessariamente associada a uma falha da empresa e pode ser devido a problemas pessoais do cliente, como um problema financeiro, por exemplo;
- **Perda de interesse**: pode ser que produto oferecido pela empresa não seja mais relevante para o cliente;
- **Atendimento ruim**: o problema pode estar diretamente relacionado ao tratamento que o cliente recebe. Um mau atendimento pode fazer com que o cliente não queira mais continuar utilizando os serviços oferecidos pela empresa;
- **Excesso de burocracia**: os clientes buscam, cada vez mais, facilidades. Caso a empresa exija muito esforço e demande muito tempo do cliente, pode ser que o mesmo busque um concorrente que ofereça as facilidades que ele procura;
- **Frustração**: pode ser que o cliente se decepcione com o serviço oferecido pela empresa. Este problema pode estar relacionado à qualidade do serviço ou produto oferecido, bem como à atrasos na entrega, falhas no pós-venda como suporte técnico falho para o cliente, entre outros.

## 2.8. *MARKETPLACE* E *E-COMMERCE*

Laudon e Laudon (2007) definem *e-commerce* como o uso da *internet* para conduzir negócios, referindo-se às transações comerciais que acontecem digitalmente entre organizações e indivíduos (B2C) ou entre organizações (B2B). “Na maioria dos casos, isso significa transações que ocorrem pela *Internet* e pela *Web*. Transações comerciais envolvem a saída de valores (por exemplo, dinheiro) das fronteiras individuais ou organizacionais em troca de produtos e serviços” (LAUDON; LAUDON, 2007, p. 271).

De acordo com *Corporate Finance Institute* (ca. 2020), *Gross Merchandise Value* (GMV) corresponde ao valor monetário transacionado por uma empresa em sua plataforma virtual dentro de um determinado período de tempo, sendo este valor calculado antes de deduzir possíveis despesas acumuladas.

## 2.9. ESTATÍSTICA DESCRITIVA E MEDIDAS DE CENTRALIDADE

Segundo Santos (2018), a estatística descritiva é a área da estatística responsável por coletar, organizar, sintetizar e descrever os dados. Para enriquecer a análise dos dados obtidos e completar o estudo desses dados utilizando a estatística descritiva, é necessário o conhecimento de algumas medidas de tendência central como, por exemplo, média e mediana.

Morettin e Bussab (2004, p.35) definem:

> “A mediana é a realização que ocupa a posição central da série de observações, quando estão ordenadas em ordem crescente.[...] Quando o número de observações for par, usa-se como mediana a média aritmética das duas observações centrais.[...] a média aritmética [...] é a soma das observações dividida pelo número delas.”

Além disso, há casos em que a média é afetada por valores extremos ou, quando tomada sozinha, não consegue transmitir a característica de simetria da amostra. Nestes casos, a fim de fornecer uma visão mais robusta da simetria da amostra, recomenda-se o uso da mediana frente a média. (BUSSAB; MORETTIN, 2004, p.41)

Por definição, supondo uma amostra de N elementos dispostos em ordem crescente segundo uma característica genérica x, a mediana de uma amostra é um

valor de x superior a 50% do número de elementos da amostra e inferior também a 50% do valor de N. Sob esta ótica, a mediana também é chamada de *quantil de ordem 50*. Assim, de maneira genérica, supondo 0% < p < 100%, pode-se definir *q(p)* como sendo o quantil de ordem p, o qual deixa p% dos valores da amostra abaixo de si e (100 - p)% dos valores da amostra acima de si. (BUSSAB; MORETTIN, 2004, p.42)

# 3. DIAGNÓSTICO DA SITUAÇÃO ATUAL

## 3.1. OS PRODUTOS DA EMPRESA X

No ano de 2021, a Empresa X conta com quatro produtos tecnológicos usados para fomentar o desenvolvimento do varejo da construção civil no Brasil. Os produtos, bem como seu histórico de criação são detalhados a seguir.

### 3.1.1. Programa de fidelidade para lojas de varejo da construção civil

O primeiro produto lançado no mercado pela Empresa X, no ano de 2018, com o objetivo de alavancar as vendas das três empresas sócias, além de avaliar potenciais parceiros para adentrar no ecossistema que a Empresa X criava foi o programa de fidelidade, ou *Loyalty* como denominado internamente na empresa. Este produto, de modo análogo ao tradicional programa de acúmulo de milhas das empresas aéreas, consiste, do ponto de vista do usuário do programa, na seguinte sequência de passos:

I. **Usuário do programa compra da Empresa X:** usuário, o qual deve ser obrigatoriamente uma pessoa jurídica, ou seja, deve possuir um Cadastro Nacional de Pessoa Jurídica (CNPJ) atrelado a si, compra algum produto da Empresa X. Pelo fato da obrigatoriedade de pessoa jurídica, o usuário aqui será também chamado de loja. Cabe ressaltar aqui que, como a Empresa X é formada por três empresas sócias do setor de construção civil, os produtos disponíveis para compra na Empresa X são obrigatoriamente produzidos e distribuídos pelas sócias, as quais possuem centros de distribuição alocados em todas as regiões do país. Além disso, um ponto importante sobre este primeiro passo é o canal pelo qual o usuário entra em contato com a Empresa X. Não há nenhuma plataforma virtual construída pela Empresa X para facilitar o contato entre o usuário e as empresas sócias fornecedoras. Assim, os canais usados para compras são os próprios canais virtuais das empresas sócias que envolvem a presença de um representante comercial de alguma das sócias na loja do usuário;

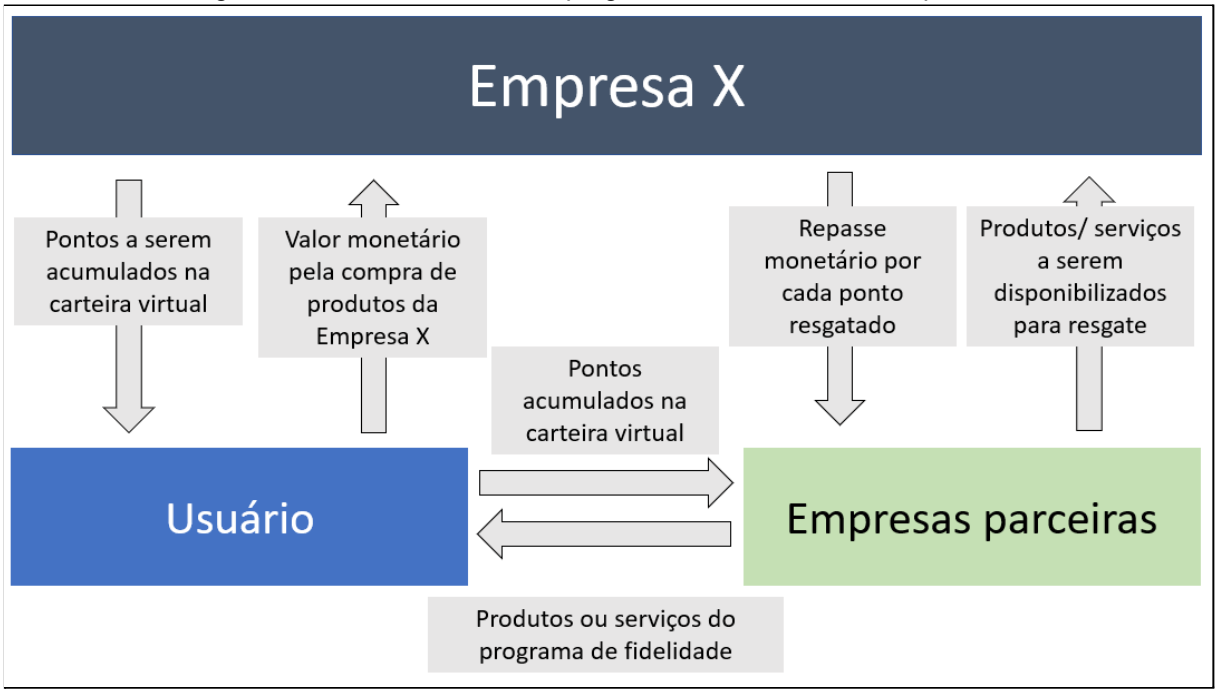
II. **Ao comprar, usuário acumula pontos no programa de fidelidade**: de acordo com as disponibilidades de estoque ou relações de oferta e demanda pelos produtos das empresas sócias, a Empresa X determina quantos pontos no programa de fidelidade o usuário acumulará pela compra de cada um dos produtos pertencentes ao programa. Cabe ressaltar aqui que não é todo o portfólio de produtos das sócias que está passível de acúmulo de pontos pelo programa de fidelidade. Uma vez que os usuários acumulam os pontos, estes são guardados em uma carteira virtual presente em uma plataforma virtual desenvolvida pela Empresa X;

III. **Usuário, pela plataforma virtual do programa de fidelidade, resgata os pontos:** a partir do momento em que os pontos estão acumulados na plataforma virtual, o usuário pode ter a opção de “resgatá-los”. Este é um termo interno à Empresa X e representa a troca dos pontos por bens ou até serviços de empresas parceiras ao programa. Assim, o usuário pode trocar seus pontos por produtos de outras empresas que não são as sócias da Empresa X, mas sim suas parceiras no programa de fidelidade. Estes produtos e serviços a serem resgatados não necessariamente estão presentes no contexto da construção civil. A maior parte dos produtos trocados por pontos são eletrodomésticos;

IV. **Empresa X repassa para as empresas parceiras um valor pré-acordado pelos pontos resgatados:** a cada ponto resgatado pelo usuário, a Empresa X repassa um valor monetário pré-estabelecido entre a Empresa X e a empresa parceira cujo produto foi trocado pelos pontos do programa. Assim, quanto mais produtos das empresas parceiras forem trocados por pontos, sendo, portanto, direcionados para o usuário, maior será o repasse financeiro da Empresa X para a empresa parceira.

No ano de 2021, o programa de fidelidade da Empresa X conta com uma base de mais de 400 mil usuários cadastrados e aptos a resgatar pontos. A Figura 3 sintetiza as relações de troca envolvidas no programa de fidelidade da Empresa X.

Figura 3 - Trocas existentes no programa de fidelidade da Empresa X




Fonte: Elaborado pelo autor

### 3.1.2. Programa de fidelidade para profissionais de obra

De maneira análoga ao programa de fidelidade para lojas de varejo, após 6 meses desde a criação do primeiro produto da empresa, em 2018, foi criado também o programa de fidelidade para profissionais de obra, o segundo produto da Empresa X. Este produto tem como objetivo estabelecer as mesmas relações de trocas do primeiro produto, mas voltadas para pessoas físicas que trabalham como autônomos no setor de construção civil. A este grupo de usuários, que obrigatoriamente possui um Cadastro de Pessoa Física (CPF) atrelado a si, foi dado o nome de **profissional de obra** pela Empresa X.

As relações de troca entre o usuário e as empresas parceiras são idênticas ao programa de fidelidade para lojas do varejo. A única diferença entre ambos os produtos é a magnitude da base de usuários cadastrados. Este segundo produto possui, em 2021, uma base de 15 mil usuários cadastrados e aptos a participarem do programa

### 3.1.3. Plataforma virtual para contratação de serviços de obra

Este produto é o lançamento mais recente da empresa estudada e surgiu a partir de um movimento estratégico da Empresa X visando constituir um portfólio de produtos que possibilite a criação e controle de um ecossistema voltado à tecnologia aplicada ao setor de construção civil. No primeiro trimestre de 2021, a Empresa X adquiriu, a partir de seu próprio capital fechado, uma startup originada em Porto Alegre no ano de 2016, que havia criado um aplicativo que buscava unir a demanda por serviços de obra com a oferta desses serviços por profissionais de obra cadastrados no aplicativo.

A partir da aquisição dessa *startup*, a Empresa X adquiriu tanto os recursos humanos da incipiente empresa de tecnologia quanto a plataforma virtual a qual, atualmente, está à disposição dos usuários, sejam eles pessoas físicas ou jurídicas, da Empresa X para que possam contratar serviços de obra diversos. Dentro do modelo de negócio da Empresa X, para este produto, os usuários que contratam os serviços pela plataforma virtual realizam o pagamento à Empresa X, a qual assume papel tanto na tecnologia que une oferta à demanda, quanto nas questões que envolvem o pagamento pelo serviço e repassa aos profissionais contratados uma parte do total pago pelo usuário.

A receita da Empresa X dentro deste produto é gerada a partir do *take rate* aplicado ao valor pago pelo usuário, isto é, a Empresa X cobra uma taxa por cada transação realizada em seu *marketplace*. Atualmente, o uso deste produto da Empresa X conta com cerca de 100 mil usuários cadastrados e aptos a usar o produto e 900 profissionais ativos habilitados a oferecer seu serviço à plataforma virtual. Cabe ressaltar que este produto é feito para que usuários contratem serviços dos colaboradores de modo que nenhum produto, ou seja, bem material, seja transacionado por essa plataforma virtual.

### 3.1.4. Loja Virtual

Depois do programa de fidelidade, tanto para pessoas físicas quanto para pessoas jurídicas, este produto é o segundo produto tecnológico a compor o portfólio de produtos da Empresa X e, dentro do contexto da empresa, será usado como base para o desenvolvimento deste trabalho. Seu uso como base para o trabalho é justificado pela atuação do autor do presente trabalho que, dentro do organograma

da empresa, estagia em uma equipe voltada exclusivamente ao funcionamento deste produto.

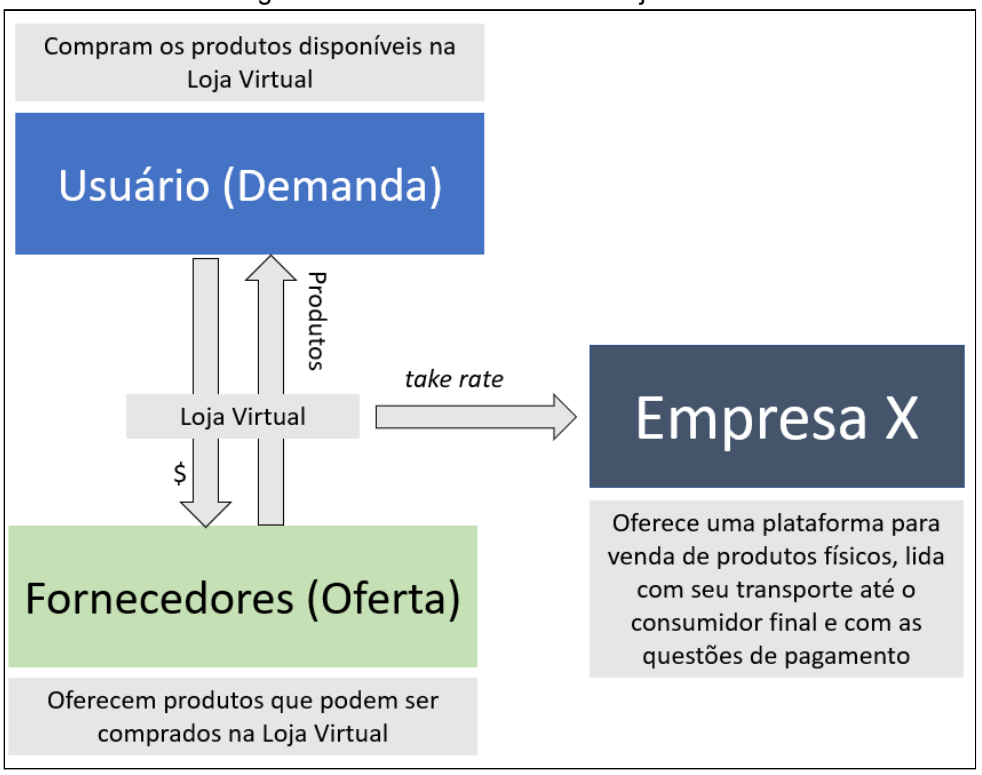
Ao longo do presente trabalho, a fim de evitar repetições, este produto terá o nome de Loja Virtual (LV) visto que é dessa maneira que todos os colaboradores da Empresa X se referem à plataforma virtual para comercializar produtos da construção civil. Postas as devidas premissas, a LV surgiu em meados do ano de 2019 a partir de uma iniciativa da primeira das sócias, que propôs que fosse criada uma plataforma virtual que possibilitasse aos usuários comprarem produtos de qualquer uma das três sócias sob as mesmas condições de pagamento sem necessitar da presença de um representante comercial de alguma das sócias.

Cabe ressaltar aqui que, na época do surgimento da LV, a Empresa X já era madura o suficiente no programa de fidelidade, mas os usuários deste programa, como citado anteriormente, só poderiam comprar produtos de alguma das sócias através de canais próprios das sócias, por meio, majoritariamente, da ação de representantes comerciais. Até então, o processo de compra dos produtos das sócias era heterogêneo e sujeito a políticas de frete e meios de pagamento particulares de cada sócia. A Loja Virtual surgiu inicialmente como uma tentativa de criar um canal único dentro do qual as sócias poderiam se valer de seus produtos, centros de distribuição e logística para entregar seus produtos aos usuários sem o intermédio de representantes comerciais, oferecendo um fluxo de compra uniforme, independente da sócia cujo produto for comprado.

Assim, a LV evoluiu, durante o ano de 2020 de uma ideia para um *marketplace b2b*, isto é, um ambiente virtual utilizado para unir a demanda por bens materiais dos usuários com a oferta e distribuição destes produtos por fornecedores participantes do *marketplace*. A Empresa X, nesse caso, de maneira análoga ao aplicativo para contratar serviços de obra, une demanda com oferta, oferecendo uma plataforma de tecnologia para tanto. A diferença com o produto mais recente da empresa é que, na Loja Virtual, apenas são comercializados bens materiais e não serviços. Além disso, a receita da Empresa X, na Loja Virtual, surge a partir do *take rate* aplicado ao GMV dos produtos comercializados na plataforma. É importante ressaltar que, por ser definida como um *marketplace b2b*, os produtos da loja virtual são destinados a pessoas jurídicas e, na prática, a maior parte do público consumidor da Loja Virtual são lojas de varejo da construção civil. Mais detalhes sobre a segmentação dos clientes da Loja Virtual serão disponibilizados em seções

seguintes. A Figura 4 apresenta os mecanismos de troca existentes na Loja Virtual.

Figura 4 - Trocas realizadas na Loja Virtual




Fonte: Elaborada pelo autor

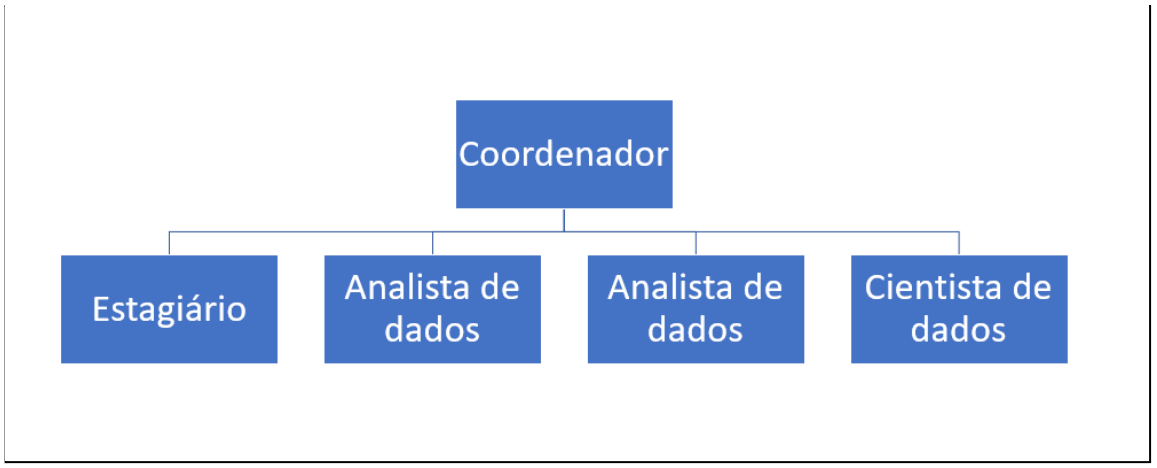
## 3.2. DISTRIBUIÇÃO DAS EQUIPES

Dado o contexto da Empresa X, o autor do presente trabalho, desde novembro de 2020, trabalha como estagiário em análise de dados no *squad* da Loja Virtual. Como a maior parte das empresas emergentes de tecnologia, comumente chamadas de *startups*, a Empresa X adotou um modelo de divisão e alocação dos seus recursos humanos em *squads*, isto é, grupos multidisciplinares formados por pessoas de diferentes funções, mas unidas em prol de um projeto comum. No caso da Empresa X, cada produto citado nas seções anteriores, possui um ou mais *squads* dedicados a si dependendo da maturidade do produto. No caso da Loja Virtual, há 4 *squads* dedicados a este produto, cada um com sua função específica. Estes *squads* são brevemente descritos a seguir e seus nomes serão apresentados da mesma maneira a qual são referidos internamente pelos colaboradores da empresa:

- ***Squad* comercial:** este grupo é formado por 10 colaboradores de formações distintas, porém com função comum de trazer novas empresas fornecedoras para a Loja Virtual. Atualmente, além das três sócias, a Loja Virtual conta com mais outras 18 empresas fornecedoras. Estas fornecedoras têm seus produtos apresentados na plataforma virtual e aptos a serem comprados pelos usuários. Dá-se aos fornecedores de uma plataforma virtual, que terão seus produtos comprados, o nome de *sellers* ou vendedores. Resumindo, este *squad* trata das questões contratuais e seu objetivo é sempre trazer novos *sellers* para a plataforma. Este é o squad mais antigo da Loja Virtual, existindo desde os primórdios do produto. Na data de criação deste trabalho, em outubro de 2021, a LV conta com 21 *sellers* distintos no *marketplace*;
- ***Squad* de performance da plataforma:** este grupo tem o objetivo de garantir o bom funcionamento da plataforma, cuidando, além de sua estabilidade, de possíveis melhorias. O *squad* de performance conta com 20 colaboradores, sendo 15 desenvolvedores de *software*, comumente chamados de programadores;
- ***Squad* de experiência do usuário:** formado por 5 colaboradores, este *squad* tem a função de entender o comportamento dos usuários da plataforma a fim de identificar padrões e possíveis oportunidades de melhorias, as quais podem ser endereçadas ao *squad* de performance e eventualmente implementadas na Loja Virtual. O seu surgimento, em julho de 2020, se dá pela preocupação da diretoria com a complexidade de uso da plataforma, muitas vezes explicitada nas avaliações dos usuários dentro do *site*. Uma das maiores rotinas deste grupo é realizar pesquisas qualitativas com bases de usuário segmentadas a fim de validar hipóteses;
- ***Squad* de ciência e análise de dados:** é neste *squad* que se insere o autor do presente trabalho. Esta equipe conta com 5 colaboradores e possui dois principais objetivos: manter a sanidade das bases de dados geradas pela plataforma da Loja Virtual e analisar estes dados a fim de validar hipóteses de negócio. O squad de dados é um grupo que alimenta todos os outros *squads* da Loja Virtual com dados e análises, sendo, portanto, um dos *squads* mais interdisciplinares dentro da empresa. De maneira análoga ao *squad* de experiência do usuário, este *squad* surgiu

pouco antes de janeiro de 2021, mais precisamente em dezembro de 2021. O organograma do *squad* de análise de dados é apresentado na Figura 5 . O Cientista de Dados é focado em garantir a atualização e sanidade das bases de dados enquanto que os dois analistas de dados têm seus objetivos centrados em analisar este dados a fim de validar hipóteses de outros times.

Figura 5 - Organograma do *squad* de análise de dados da Loja Virtual




Fonte: Elaborada pelo autor

## 3.3. RESULTADOS DA PESQUISA QUALITATIVA

O surgimento do *squad* de experiência do usuário, buscando entender o comportamento do usuário da Loja Virtual, incentivou a Empresa X a promover uma pesquisa quantitativa a fim de medir o grau de fidelidade de seus usuários à plataforma virtual.

Dado este contexto, de acordo com informações da Empresa X, a pesquisa realizada em novembro de 2020 englobou uma amostra com 450 usuários ativos, ou seja, que já compraram pelo menos alguma vez na plataforma virtual. Destes 450 usuários, 102 foram classificados como promotores e 225, como detratores, o que gerou para a Loja Virtual um NPS de -27%. O NPS negativo na pesquisa realizada em janeiro de 2021 ia de encontro ao momento de crescimento que vivia a Loja Virtual em termos de usuários por mês. Esta tendência de crescimento será melhor detalhada em seções posteriores deste trabalho, mas, de uma maneira geral, no segundo semestre de 2020, a Loja Virtual apresentou 13.640 usuários distintos

comprando em sua plataforma, um crescimento de 232% com relação ao semestre anterior.

Havia, portanto, uma dicotomia existente no final do ano de 2020 na Loja Virtual que trazia um crescimento, em termos de usuários, frente a um valor negativo para a pesquisa do NPS realizada. Por isso, a Empresa X atribuiu ao *squad* de experiência do usuário a tarefa de entender o comportamento detrator do usuário da Loja Virtual. Assim, o *squad* em questão, em uma de suas análises, no começo de 2021, verificou que, em média, 24 % dos usuários que compram em um determinado mês na Loja Virtual, não compram no mês seguinte. A esta métrica, a Empresa X atribuiu o nome de “desistência” e criou, em fevereiro de 2021, o *squad* de análise e ciência de dados para **identificar as causas que mais impactam na desistência dos usuários**.

## 3.4. O *SOFTWARE* COMPUTACIONAL USADO

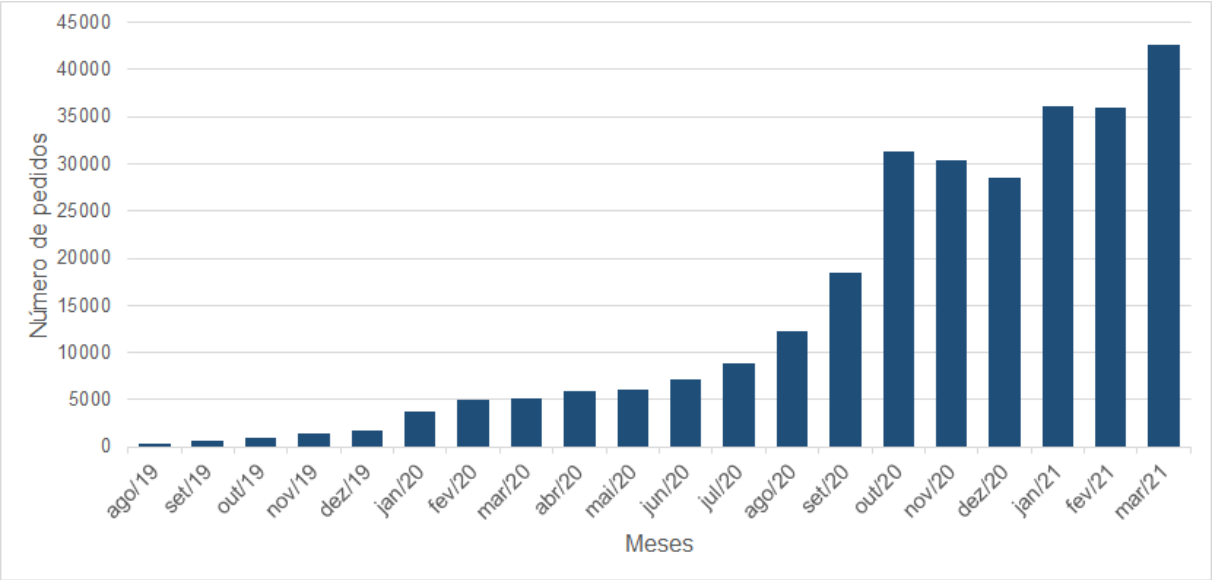
Para realizar tanto as análises associadas à métrica de desistência atual da Empresa X, bem como para gerar os dados necessários para a nova proposta de métrica de *churn*, e até mesmo os modelos de *Machine Learning*, foi utilizada a linguagem de programação *Pyspark*, uma interface que conecta a tradicional linguagem de programação *Python* com o ambiente de alto poder computacional voltado para *Big Data Apache Spark*. Faz parte da cultura organizacional da Empresa X que as análises dos dados sejam feitas em um servidor virtual, sem utilizar a memória local dos computadores dos colaboradores. Então, o poder computacional das análises deste trabalho foi gerado por um servidor (ou *Cluster*) virtual particular da Empresa X, com 16 núcleos, 32 GB de memória e suportando a versão 3.1.1. da linguagem *Spark*. Os trechos de código utilizados para realizar as análises estão disponíveis na seção de apêndice deste trabalho, bem como a base de dados utilizada. Cabe ressaltar aqui que, nos trechos de código, a base de dados usada para as análises está explicitada como *dfInput*. Além disso, os nomes das colunas, quando associadas a tabelas dentro do código, são escritas em língua inglesa a fim de se manter o padrão de trabalho dentro da Empresa X. As colunas que possuírem dados relacionados a datas, estarão sempre no formato “YYYY - MM - DD”, onde ‘Y’ são os algarismos que compõem o ano, ‘M’ são os algarismos que compõem o mês e ‘D’, compõem o dia.

# 4. ANÁLISE DA MÉTRICA DE DESISTÊNCIA

## 4.1. CÁLCULO DE TAXA DE DESISTÊNCIA ANTES DA CRIAÇÃO DO SQUAD DE ANÁLISE DE DADOS

Antes de se diagnosticar o cálculo da métrica de desistência atual usada para a Empresa X, é importante ressaltar que a Loja Virtual, desde sua criação em 2019, viveu dois momentos distintos tanto em termos de *shoppers* comprando quanto em número de pedidos.

Gráfico 1 - Evolução mensal do número de pedidos na Loja Virtual

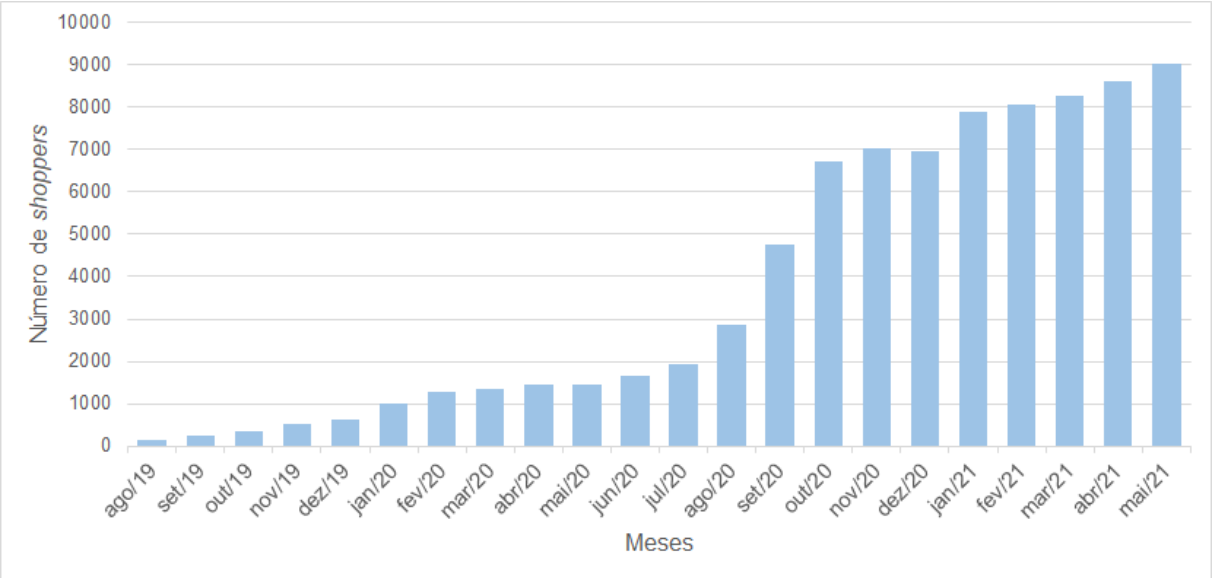



Fonte: Elaborado pelo autor

O Gráfico 1 apresenta a evolução mensal do número de pedidos realizados dentro da Loja Virtual desde sua criação até o diagnóstico do problema em abril de 2021. O Gráfico 2 por sua vez explicita, no mesmo contexto, a evolução mensal do número de *shoppers* na plataforma. Através da análise dos gráficos, é possível perceber que a Loja Virtual vem sofrendo um crescimento acelerado desde julho de 2020. Este fato, explicado pela Empresa X, é devido à migração da base de clientes de uma das sócias para a plataforma da Loja Virtual. Em outras palavras, uma das sócias optou por incentivar seus clientes, que antes compravam seus produtos

através de canais *offline*, a comprarem pelo *marketplace* estabelecido pela Empresa X.

Gráfico 2 - Evolução mensal do número de *shoppers* na Loja Virtual

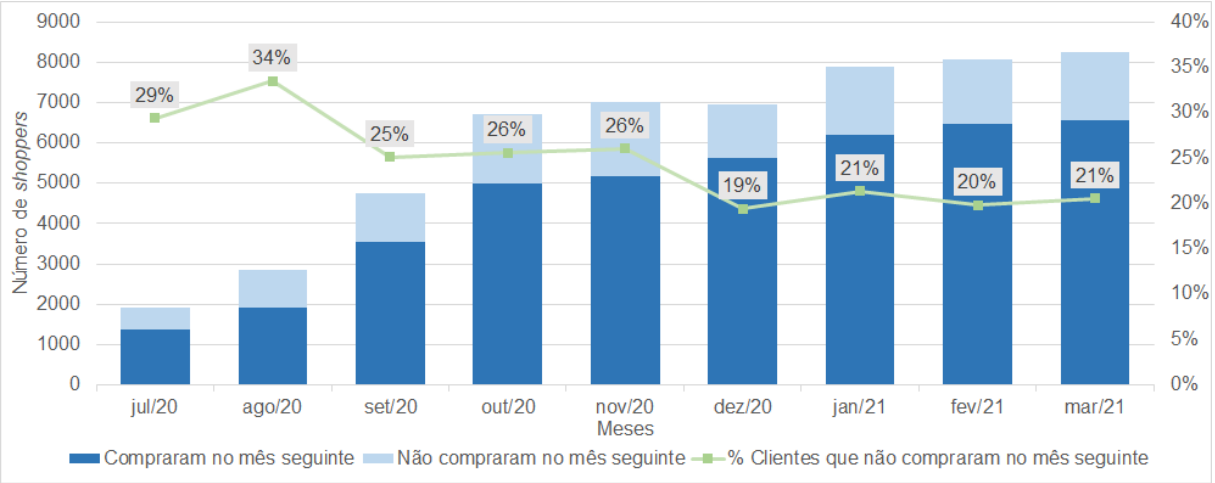



Fonte: Elaborado pelo autor

Assim, a Empresa X definiu que, para o cálculo de sua métrica atual de desistência, seriam apenas levados em conta os dados relativos aos meses posteriores a julho de 2020 até o final de março de 2021, data do diagnóstico do problema.

A preocupação inicial levantada pela diretoria da Empresa X e endereçada ao *squad* de análise de dados girava em torno da métrica de desistência proposta pela empresa, a qual, como apresentado em seções anteriores, dizia de maneira genérica que, em média, 24% dos usuários que compravam em um mês deixavam de comprar no mês seguinte. Através de manipulações da base de dados fornecida pela empresa, no Gráfico 3, é possível identificar a evolução mensal dos clientes de acordo com a segmentação de terem ou não comprado na Loja Virtual no mês seguinte. A média levantada pela Empresa X é, portanto, obtida a partir da média aritmética do percentual de clientes que não compraram no mês seguinte.

Gráfico 3 - Evolução do percentual de clientes desistentes segundo a métrica atual da Empresa X




Fonte: Elaborado pelo autor

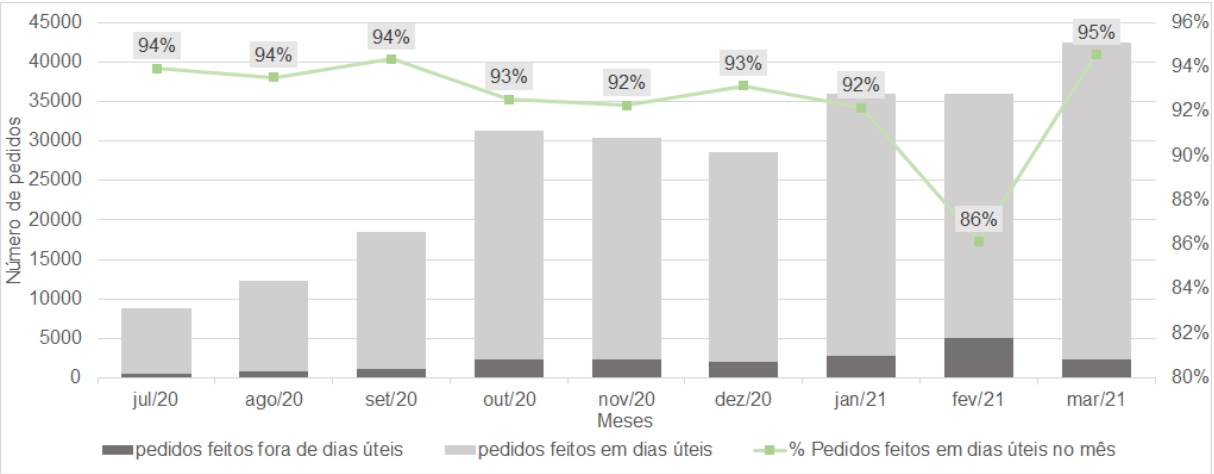
## 4.2. CRÍTICAS AO MÉTODO ATUAL E PROPOSTA DE NOVA MÉTRICA

O modelo usado para calcular a taxa de desistência até então adotado pela Empresa X tem sua principal limitação associada ao período de análise. A métrica atual parte de observações de um dado mês e as classificam de acordo com a interação do cliente única e exclusivamente no mês seguinte. Deste modo, a Empresa X não considera, em sua métrica, por exemplo, casos em que o cliente compra a cada dois ou três meses. Deste modo, o modelo atual pressupõe que a frequência de compra da base de clientes da Loja Virtual seja mensal, perdendo visibilidade dos clientes cujo comportamento de compra não acontece a todo mês.

Desde modo, o *squad* de análise de dados percebeu que a Empresa X sequer possuía uma segmentação clara da sua base de clientes em termos de frequência de compra, ressaltando, assim, uma oportunidade para o time de, antes de definir uma métrica de desistência, propor um método para se definir a frequência de compra dos clientes da Empresa X.

A partir da análise exploratória dos dados de compras fornecidos pela Empresa X, com base no Gráfico 4, é possível afirmar que, com exceção do mês de fevereiro de 2021 com o feriado de carnaval, mais de 92% dos pedidos da Loja Virtual são realizados dentro de dias úteis, isto é, dias da semana com exclusão de finais de semana e feriados. Um pedido, de acordo com a nomenclatura interna da empresa, é definido como uma compra realizada pelo *shopper*, independentemente da quantidade de itens presentes nesta compra.

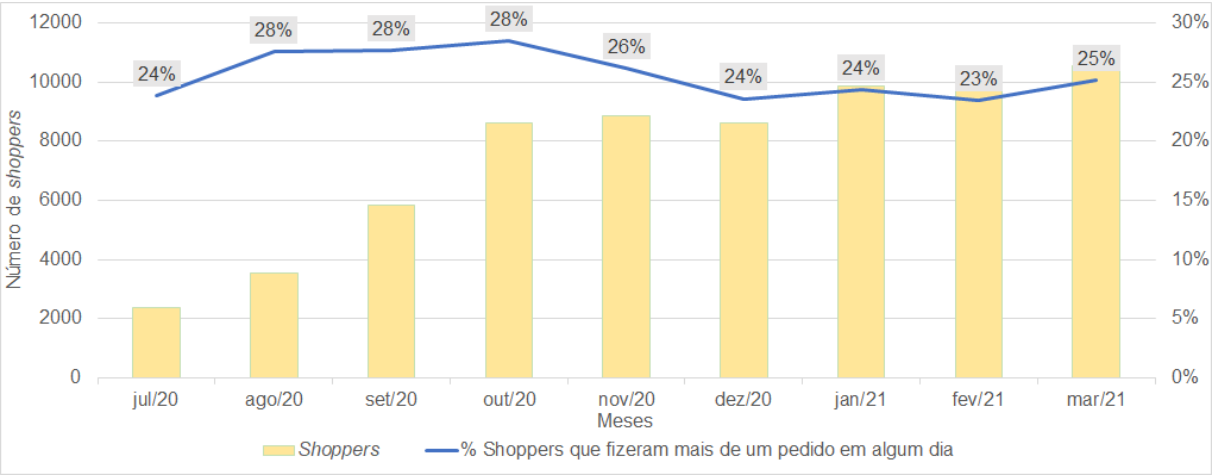
Gráfico 4 - Evolução mensal dos pedidos feitos em dias úteis




Fonte: Elaborado pelo autor

Além disso, o Gráfico 5, permite inferir que, por mês, mais de 70% dos *shoppers* não realizam mais de um pedido dentro do mesmo dia. Esta inferência é necessária, pois a base de dados fornecida pela Empresa X apresenta a data de realização do pedido sem, entretanto, evidenciar a hora do dia em que o pedido ocorre. Assim, em termos de dados, caso, para um mesmo *shopper*, haja mais de um pedido por dia, é impossível distinguir qual dos pedidos aconteceu primeiro. Deste modo, a fim de simplificar este problema e embasando-se no Gráfico 5, para a definição da frequência de compra, será criado o conceito de **dia de compra**.

Gráfico 5 - Evolução mensal dos *shoppers* que realizam mais de um pedido em algum dia do mês




Fonte: Elaborado pelo autor

Um dia de compra representa simplesmente a data do dia em que o *shopper* realizou a compra dentro da Loja Virtual sem considerar quantas compras foram realizadas dentro daquele dia. Este conceito será usado para o cálculo da frequência de compra dos *shoppers* presentes na base de dados fornecida pela Empresa X.

Para a definição da frequência de compra dos clientes da Loja Virtual, com base nos Gráficos 4 e 5 acima, o *squad* de análise de dados propôs as seguintes etapas aplicadas para cada *shopper*:

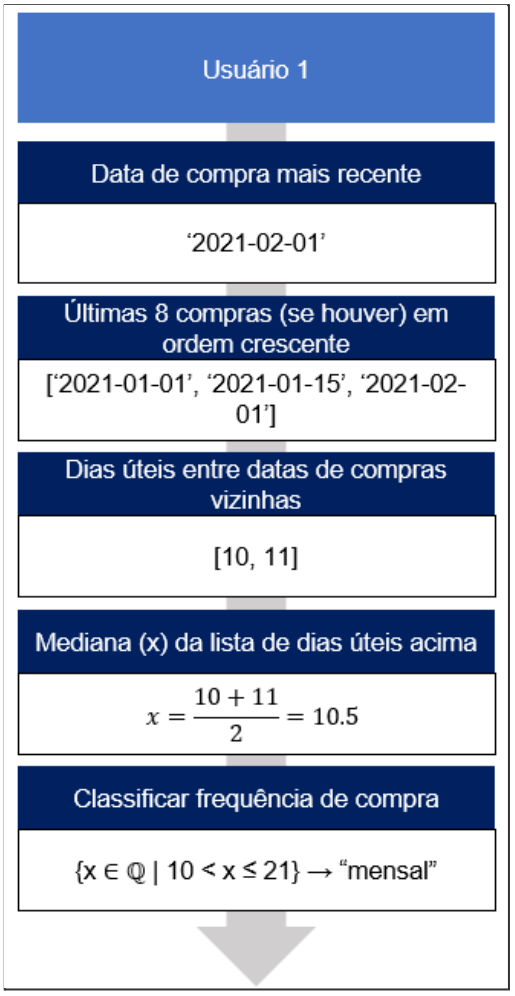
I. Identificar, dentro do intervalo de julho de 2020 até março de 2021, a data de compra mais recente do *shopper*;
II. Identificar e colocar em uma lista, a partir da data de compra mais recente, as 8 datas de compras anteriores. Caso o *shopper* possua menos que 8 datas de compra antes da data mais recente, serão consideradas quantas datas houver. Este passo é importante, pois garante que o método de definição da frequência de compra não analise o histórico de compra do *shopper*, mas apenas para o passado recente, de modo a considerar o comportamento recente para a definição da frequência. Por isso, foi escolhido o valor de 8 datas de compras prévias;
III. Com as datas anteriores agrupadas em uma lista, calcular para esta lista ordenada em ordem crescente, os dias úteis entre os elementos vizinhos desta lista. Assim, caso a lista possua 8 datas de compra por exemplo, serão calculados 7 intervalos de dias úteis entre compras
IV. A fim de evitar *outliers*, calcular a mediana dos intervalos de dias úteis entre compras
V. Com a mediana definida, classificar o *shopper* em termos de frequência de compra de acordo com a Tabela 1. Os valores da Tabela 1 foram definidos com base no número de dias úteis dentro de uma semana. A Figura 6 elucida a lógica por trás da classificação da frequência de compra.

Tabela 1 - Classificação da frequência de compra

| **Intervalo da mediana de dias úteis entre as últimas compras, sendo *x* a mediana** | **Classificação da frequência de compra** |
|---|---|
| $\{x \in \mathbb{Q} \mid 0 < x \leq 5\}$ | semanal |
| $\{x \in \mathbb{Q} \mid 5 < x \leq 10\}$ | quinzenal |
| $\{x \in \mathbb{Q} \mid 10 < x \leq 21\}$ | mensal |
| $\{x \in \mathbb{Q} \mid 21 < x \leq 42\}$ | bimestral |
| $\{x \in \mathbb{Q} \mid 42 < x \leq 63\}$ | trimestral |
| $\{x \in \mathbb{Q} \mid 63 < x\}$ | semestral |
| sem valor | primeira compra |

Fonte: Elaborado pelo autor

Figura 6 - Exemplo de cálculo de frequência de compra




Fonte: Elaborada pelo autor

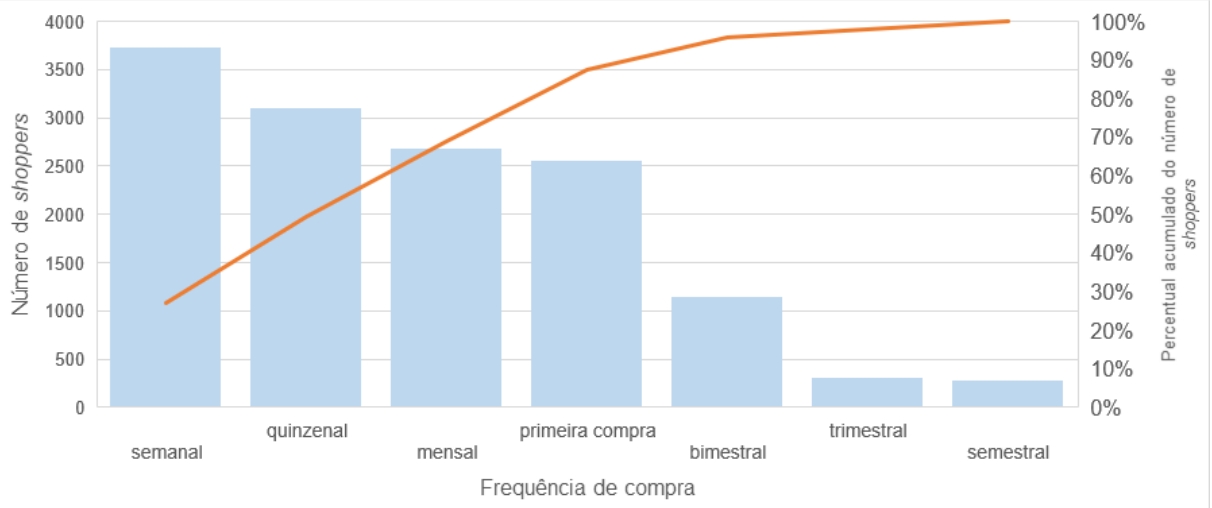
Utilizando a lógica de classificação da frequência de compra proposta anteriormente, a base de *shoppers* que interagiu com a Loja Virtual entre os meses de julho de 2020 e março de 2021 é descrita na Tabela 2. Além disso, cabe ressaltar que foi criado um código de frequência como sendo um número natural entre 0 e 6 a fim de melhor exemplificar o processo de definição da nova métrica em seções posteriores. No caso em que o código de frequência de compra é igual a 0, a observação mais recente do *shopper* dentro do intervalo de tempo analisado só pôde identificar a primeira compra do *shopper*. Assim, seu comportamento quando comparado aos *shoppers* de outras frequências de compra é mais incerto. Quando se fizer necessário estimar alguma característica baseada na frequência de compra de um *shopper*, estes clientes com frequência de compra igual a zero terão seu comportamento baseado na média ponderada do comportamento das outras frequências.

Tabela 2 - Distribuição dos *shoppers* em cada frequência

| **Número de *shoppers*** | **Classificação da frequência de compra** | **Código de frequência** |
|---|---|---|
| 3735 | semanal | 1 |
| 3103 | quinzenal | 2 |
| 2684 | mensal | 3 |
| 1152 | bimestral | 4 |
| 317 | trimestral | 5 |
| 280 | semestral | 6 |
| 2558 | primeira compra | 0 |

Fonte: Elaborada pelo autor

Gráfico 6 - Gráfico de Pareto para o número de *shoppers* e suas frequências de compra




Fonte: Elaborado pelo autor

O Gráfico 6, aliado às informações da Tabela 2, mostram que cerca de 10% dos clientes que compraram da Loja Virtual no período de análise para a atual métrica de desistência se enquadram em frequências de compra bimestral, trimestral ou semestral. Deste modo, pode-se concluir que a métrica atual desconsiderava cerca de 10% da base atual de clientes ao definir como desistência, de maneira genérica, apenas os clientes que não compraram no mês seguinte. A proposta de nova métrica realizada, então, pelo *squad* de análise de dados, leva em consideração a frequência de compra da base atual de clientes.

## 4.3. PROPOSTA DE NOVA MÉTRICA PARA DESISTÊNCIA

Embasado na definição de frequência de compra, o *squad* de análise de dados pôde propor uma nova métrica para definir o *churn* dos clientes que leve em consideração a recorrência segundo a qual aquele cliente compra na Loja Virtual. Cabe relembrar, aqui, que o método de classificação da frequência de compra já engloba as oito últimas datas de compra, incluindo a data de compra mais recente. Tal ressalva deve ser feita, pois a proposta de nova métrica será embasada apenas por duas variáveis de entrada: a frequência de compra do *shopper* e a data de compra imediatamente anterior à data mais recente. Portanto, a frequência de compra, como ressaltado na seção anterior, já traz consigo informações do passado recente de compras de cada um dos *shoppers*.

A lógica elaborada, então, para definir a nova desistência é baseada em delimitar, para cada faixa de frequência de compra, um limite superior em dias úteis desde a última compra. Caso o *shopper*, dada a sua frequência de compra, permaneça há mais dias úteis sem comprar do que o limite superior, será considerada uma desistência.

As etapas do processo de definição dos limites superiores para cada frequência de compra foram baseadas ainda nos dados de julho de 2020 até março de 2021 e são apresentadas a seguir:

I. Supondo uma base de *S shoppers* com $\{i \in \mathbb{N} \mid 1 \leq i \leq S\}$Deve-se identificar, para o i-ésimo *shopper*:
   - sua data de compra mais recente $d_{oi}$;
   - sua data de compra imediatamente anterior à data mais recente $d_{1i}$;
   - seu código de frequência de compra $f_i$ com base na data mais recente;

II. Calcular a diferença ($x_i$) em dias úteis entre a última data de compra $d_{oi}$ e a penúltima data de compra $d_{1i}$, gerando, assim, o par ordenado ($f_i$, $x_i$);

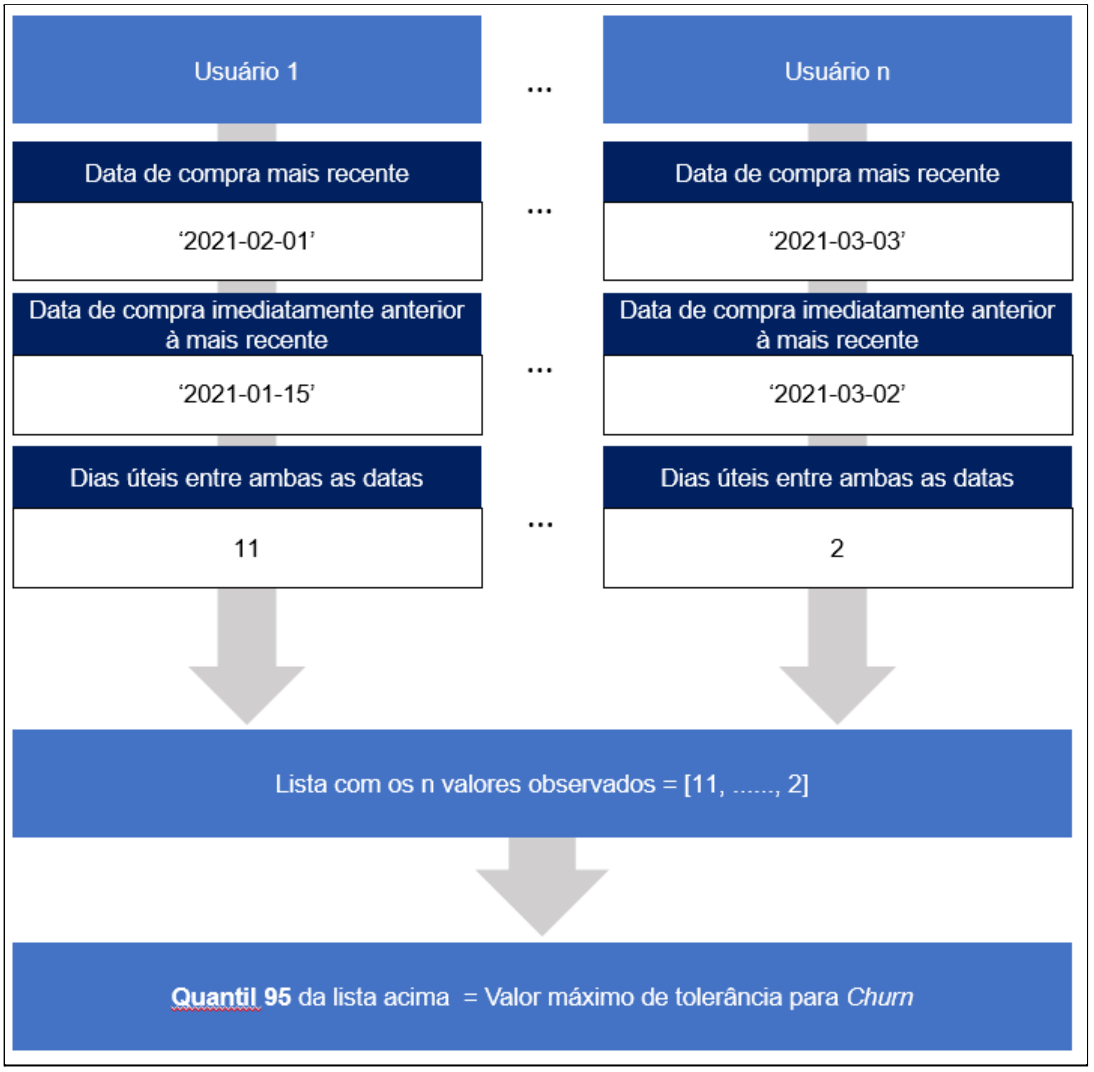
III. Guardar $x_i$ em um conjunto $C_f$ de acordo com o código de frequência de compra do i-ésimo *shopper*. Assim, por exemplo, caso a frequência de compra do i-ésimo *shopper* seja “semanal”, isto é, $f_i$ é igual a 1, $x_i$ para este dado *shopper* deverá ser guardado no conjunto $C_1$ e assim por diante para cada *shopper*;

IV. Ao final do processo, haverá 6 conjuntos $C_f$ com $\{f \in \mathbb{N} \mid 1 \leq f \leq 6\}$. Cada conjunto possuirá $N_f$ elementos de modo que $\sum_{f=1}^{6} N_f = S$. Para cada conjunto $C_f$ , suas observações serão dispostas em ordem crescente e seu quantil de ordem 95 ($q$) será determinado. Caso $q$ esteja situado entre dois elementos da amostra $C_f$, será selecionado o elemento de $C_f$ mais próximo de $q$ como o novo quantil de ordem 95. Este é um método usado pela função em Python usada para calcular quantis de ordem $n$;

V. Definir o valor máximo de dias úteis entre as duas últimas compras, para cada faixa de frequência de compra, como sendo o quantil de ordem 95 citado acima. Deste modo, a partir do comportamento de 95% da amostra, espera-se definir qual o valor máximo de dias úteis que um *shopper* pode ficar sem comprar na Loja Virtual sem que seja considerado como um

desistente. A Figura 7 elucida o processo de criação do valor máximo de dias úteis entre compras.

Figura 7 - Cálculo do número de dias úteis máximo para cada frequência




Fonte: Elaborada pelo autor

Deste modo, para cada faixa de frequência de compra, foi definido um valor máximo de dias úteis entre compras. Caso o *shopper*, dada sua frequência de compra, fique mais dias úteis sem comprar do que o limite máximo, será considerado um *churn*. Este limite máximo é necessário, pois para analisar dados atuais, ou seja, que não fazem parte de um histórico, a empresa precisa de um valor a partir do qual considerará o *shopper* um desistente ou não. Cabe ressaltar aqui que a escolha do quantil de ordem 95 se deu com base na métrica antiga de desistência. Esta métrica ignorava cerca de 10% dos *shoppers* cuja frequência de compra era bimestral, trimestral ou semestral.

Dado que a métrica antiga perdia cerca de 10% das classificações, estipulou-se para a métrica atual uma tolerância ao erro equivalente à metade do erro da métrica anterior, ou seja, de 5%. Assim, ao selecionar o quantil de ordem 95 como limite máximo de dias úteis que um *shopper* pode ficar sem comprar, aceita-se que pode haver *shoppers* que serão considerados como desistentes, mas na prática não o são.

A Tabela 3, por fim, apresenta, para cada faixa de frequência de compra, os valores máximos de dias úteis que um dado *shopper* pode ficar sem comprar na Loja Virtual antes de ser considerado um cliente desistente. Além disso, é importante evidenciar que os *shoppers* cujo código de frequência de compra equivale a 0, isto é, representam primeiras compras na Loja Virtual, não possuíam histórico de compras suficiente para que fosse delimitado seu quantil de ordem 95. Assim, neste método de classificação, o *squad* de análise de dados assume que o valor máximo para este grupo seja equivalente ao valor máximo para a frequência de compra bimestral. Tal escolha se deve ao fato de a frequência de compra bimestral estar inserida exatamente no centro entre o comportamento mais frequente (semanal) e o comportamento menos frequente (semestral).

Tabela 3 - Valores máximos de dias úteis sem compra para cada frequência de compra

| **Código de frequência de compra** | **Frequência de compra** | **Número de *shoppers* dentro de cada frequência** | **Valor máximo de dias úteis a partir do qual o *shopper* será dado como *churn*** |
|---|---|---|---|
| 1 | semanal | 3735 | 16 |
| 2 | quinzenal | 3103 | 27 |
| 3 | mensal | 2684 | 40 |
| 4 | bimestral | 1152 | 62 |
| 5 | trimestral | 317 | 90 |
| 6 | semestral | 280 | 144 |
| 0 | primeira compra | 2558 | 62 |

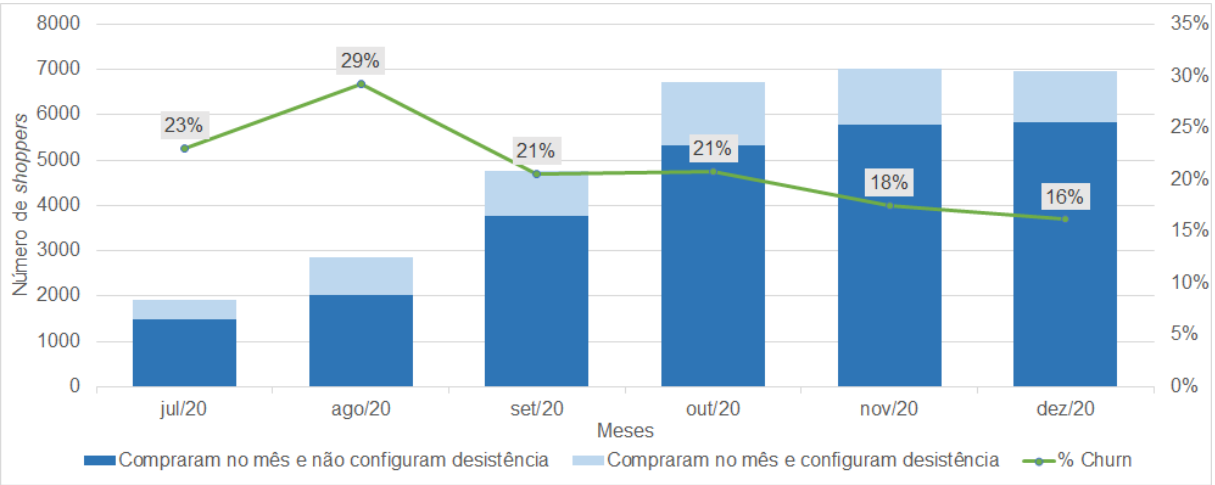
Fonte: Elaborada pelo autor

Com base nos limites máximos, a base de compradores de julho até março foi analisada sob a nova ótica proposta pela nova métrica de *churn* que leva em consideração as frequências de compra. O processo de crítica da métrica atual e proposta de nova métrica durou cerca de 2 semanas, o que fez com que a análise da base histórica sob a nova ótica fosse realizada no final do mês de abril de 2021.

O Gráfico 7 apresenta a base histórica analisada segundo a nova métrica de *churn*. No gráfico em questão, foram excluídos os meses pertencentes ao ano de 2021, pois, como este gráfico parte de *shoppers* que compraram no dado mês, caso a frequência de compra dos *shoppers* que compraram em 2021 seja bimestral ou superior, é preciso que haja pelo menos 62 dias úteis desde a última compra para se afirmar que o *shopper* desistiu. A análise, em abril de 2021, é relativamente recente, o que justifica a exclusão dos meses a fim de evitar análises distorcidas.

Postas as devidas ressalvas, é possível afirmar que, diferentemente do que foi apresentado pela diretoria antes da criação do *squad* de análise de dados, em média, segundo a nova métrica, 21% dos *shoppers* que compram em um determinado intervalo de tempo na plataforma, desistem de usar a Loja Virtual.

Gráfico 7 - Percentual de *shoppers* que desistem da LV segundo a nova métrica




Fonte: Elaborado pelo autor

## 4.4. ESCOLHA DO MODELO DE *MACHINE LEARNING* PARA IDENTIFICAR AS CAUSAS DO *CHURN*

O objetivo da Empresa X ao criar o *squad* de análise de dados para a LV desde seu princípio foi identificar os fatores inerentes ao contexto da LV que mais

impactam na desistência dos clientes. Cabe ressaltar aqui que, em nenhum momento desde sua criação, a Empresa X aplicou métodos de *Machine Learning* em nenhum de seus produtos de modo que a aplicação do presente trabalho é pioneira dentro da empresa em termos de inteligência artificial.

A partir da revisão bibliográfica de alguns métodos de ML disponíveis nas bibliotecas do programa usado como ferramenta de análise de dados dentro da empresa (*Pyspark*), foram evidenciados três candidatos: *Random Forest*, Redes Neurais Artificiais e Regressão Logística.

A presença destes modelos entre os candidatos se dá pelo fato de haver vasta gama de informações disponíveis na literatura acadêmica acerca de sua aplicação em problemas de previsão de *churn* em outras empresas de tecnologia. Um ponto crucial para a escolha do modelo reside no *tradeoff* existente entre **precisão** e **interpretabilidade** dos modelos de ML. O primeiro conceito representa o quão bem pode um modelo prever um dado evento após treinado. O segundo, por sua vez, representa o quão simples é entender quais *features* do modelo mais impactam na previsão.

A revisão bibliográfica aponta que, tanto a RNA quanto a RL são métodos com maior precisão e menor interpretabilidade das *features* uma vez que são baseados em fórmulas matemáticas com parâmetros mais complexos. A heurística por trás do RF, por sua vez, é embasada nas árvores de decisão, o que faz desse modelo menos preciso, porém com *features* mais interpretáveis.

Dado que o problema inicial trazido pela Empresa X é identificar os fatores que motivam a desistência e não prever a desistência em si, o modelo de *Random Forest* será escolhido para aplicação de um aprendizado supervisionado.

# 5. APLICAÇÃO DO MÉTODO DE *RANDOM FOREST*

## 5.1. DESCRIÇÃO DO ACESSO AOS DADOS

Antes de iniciar o procedimento de seleção das características que potencialmente impactam no *churn* da LV, é necessário explicar a estrutura de dados que foi montada dentro da Empresa X respeitando a divisão da organização por cada produto. Como foi explicado na seção de diagnóstico da situação atual, a Empresa X tem seus recursos humanos divididos em *squads* que orbitam em torno de seus 4 produtos tecnológicos. Como seus produtos perpassam estágios de maturidade diferentes, a estrutura de reposição e consumo de dados para cada produto não é homogênea.

Deste modo, o programa de fidelidade, mesmo sendo o produto mais tradicional e maduro da Empresa X, ainda possui um mecanismo de reposição e consumo de dados considerado arcaico quando comparado com outras *startups* que trabalham com grande volume de dados. O programa de fidelidade armazena seus dados em já processados em formato de tabelas dentro de bancos compartilhados na nuvem no formato *SQLite*. O pré-processamento dos dados é feito por uma máquina virtual alimentada com um código que acompanha a empresa desde sua criação. Assim, o time de dados do programa de fidelidade tem pouco poder sobre a engenharia dos dados com que trabalha. Esta estrutura de dados é restrita apenas aos colaboradores que trabalham nas equipes associadas a este produto de modo que o autor deste trabalho, por estar alocado na equipe da LV, não pôde ter acesso aos dados do programa de fidelidade nem dos outros produtos.

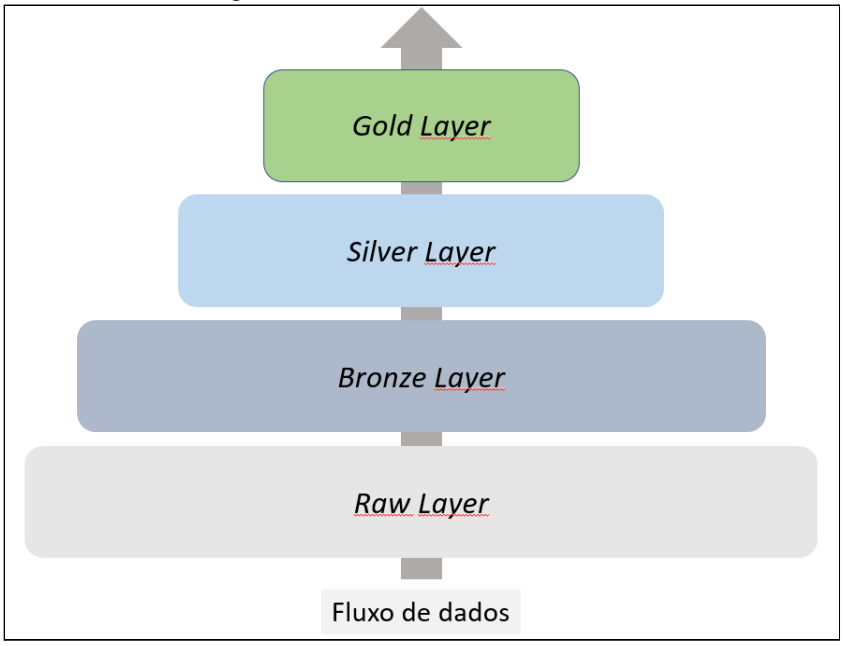
Este ponto deve ser citado, pois uma das hipóteses que poderiam ser validadas com o modelo de *Random Forest* era a relação entre *churn* na LV e uso dos demais produtos do ecossistema da Empresa X. As características de um dado *shopper* na Loja Virtual que serão usadas para alimentar o modelo de ML são chamadas de *features*. Em outras palavras, as *features* serão os candidatos a possível causa do *churn* na LV. Assim, a premissa inicial para a seleção das *features* do modelo de ML é que serão usados apenas dados inerentes à LV, sem levar em consideração os outros produtos da Empresa X dado que o acesso a estes dados é restrito.

## 5.2. A ARQUITETURA DOS DADOS DA LOJA VIRTUAL

A estrutura de dados da Loja Virtual, quando comparada com as outras estruturas de dados da Empresa X, tem sua origem em um momento mais recente no ciclo de vida da Empresa X. Apesar disso, com a criação da equipe de análise de dados da Loja Virtual, a Empresa X concedeu autonomia aos membros da equipe para decidirem que ferramentas usar para coletar, tratar e consumir seus dados. Após uma pesquisa de mercado realizada pelo líder da equipe, o *squad* de dados da LV decidiu usar uma ferramenta de armazenamento, coleta e tratamento de dados chamada *Data Bricks*, produzida pela *Microsoft*.

Esta ferramenta permite que o *squad* de análise de dados da LV tenha acesso aos dados que têm origem no código de programação sobre o qual a plataforma da LV foi construída. Estes dados são armazenados diariamente no repositório do *Data Bricks* no formato *JavaScript Object Notation* (JSON), isto é, um formato de dados não matricial, mas com capacidade de armazenamento maior. Estes dados em JSON, por sua vez, são transformados em tabelas e pré-processados pelos membros do *squad* de análise de dados da LV. A estrutura de armazenamento de dados da LV é mostrada na Figura 8.

Figura 8 - Estrutura de dados da LV



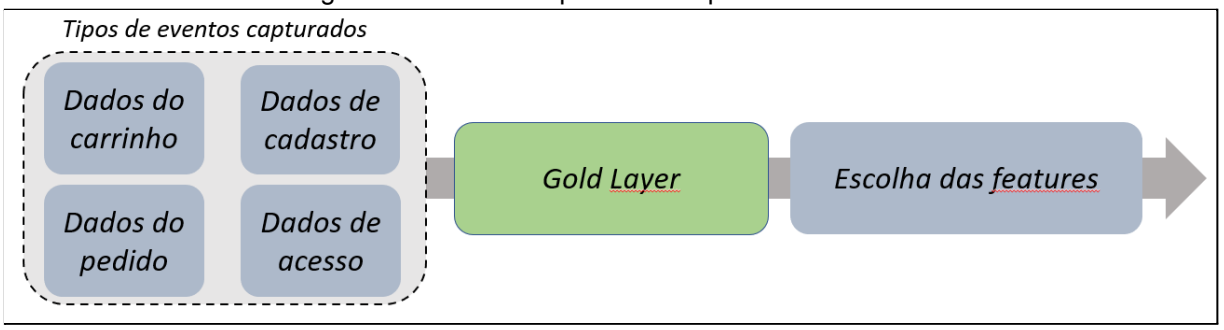

Fonte: Elaborada pelo autor

As camadas de armazenamento de dados da LV, ilustradas pela Figura 8, são apresentadas e descritas por:

- ***Raw Layer**:* nesta camada, os dados que têm origem no código da Loja Virtual no formato JSON são armazenados. Os responsáveis por alimentar esta camada com os dados são os desenvolvedores de *software* da LV. Estes responsáveis são os autores dos eventos que devem ser captados durante a interação do *shopper* com a plataforma. Assim, cada evento é implementado por algum desenvolvedor de *software* da Empresa X a fim de poder ser analisado pelo *squad* de dados;
- ***Bronze Layer:*** a partir desta camada, a responsabilidade pelos dados passa a ser exclusiva do *squad* de análise de dados da LV. É nesta camada em que os analistas de dados convertem todas as informações que estão em JSON na *Raw Layer* para o formato de tabelas, ou *delta tables* segundo a nomenclatura do *Data Bricks*;
- ***Silver Layer:*** esta camada tem como finalidade pré-processar os dados, já em formato de tabelas, da camada *Bronze*. Em outras palavras, é na *Silver Layer* em que as duplicatas das tabelas da *Bronze* são removidas e valores nulos são devidamente preenchidos;
- ***Gold Layer:*** após processados os dados, a camada *Gold* é o destino final de todas as informações inerentes à LV. É aqui o local onde os analistas de dados da LV criam as agregações necessárias para responder às perguntas de negócio. A maioria das agregações é realizada na linguagem *Pyspark*, através de funções nativas desta linguagem como *groupBy()* e *Window()*, por exemplo.

Deste modo, os dados utilizados para formar as *features* do modelo de ML estão todos disponíveis na *Gold Layer* e são os únicos dados aos quais o time de análise de dados da LV tem acesso. Como foi citado anteriormente, todos os dados presentes nesta camada representam eventos que já foram implementados pelos desenvolvedores de *Software* da Empresa X. Assim, uma outra premissa e também limitação para a escolha das *features* é que uma *feature* necessariamente precisa estar associada a um evento capturado durante a navegação do usuário na plataforma virtual da LV. A Figura 9 representa o fluxograma do processo de formação dos dados disponíveis para tornarem-se *features* do modelo de ML.

Figura 9 - Eventos capturados na plataforma da LV

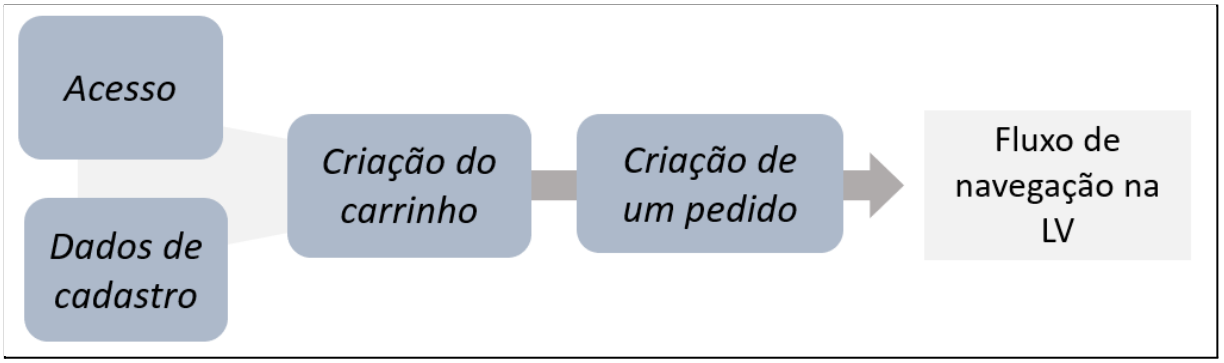



Fonte: Elaborada pelo autor

## 5.3. DESCRIÇÃO DOS EVENTOS CAPTURADOS NO CÓDIGO DA LV

Todas as informações disponíveis para alimentar o modelo de ML, como já citado anteriormente, necessitam ser mapeadas no código que embasa a LV e, portanto, o escopo da seleção das *features* gira em torno destes eventos. A lógica de implementação destes eventos foi estruturada na origem da Loja Virtual seguindo a sequência de passos que devem ser seguidos pelo usuário da plataforma para que conclua um pedido no *marketplace*. Esta sequência de passos é evidenciada na Figura 10.

Figura 10 - Fluxo de navegação na plataforma da LV




Fonte: Elaborada pelo autor

A sequência de eventos capturados durante a navegação na LV, que servirá de base para os dados disponíveis para análise, é mais minuciosamente descrita a seguir:

I. **Dados de cadastro:** para que possa navegar no ambiente virtual da Loja Virtual, um *shopper* precisa ter seu cadastro realizado na plataforma para cada *seller*. É possível, no ambiente da LV, que haja *shoppers* cadastrados

para um dado *seller*, mas não cadastrado para outro. Caso um *shopper* não esteja cadastrado para um dado *seller,* não poderá visualizar seus produtos na LV. Os dados de cadastro disponíveis na *Gold Layer* são: CNPJ, razão social, endereço e *email* do *shopper* cadastrado. Este cadastro pode ser realizado de duas maneiras:

A. O *shopper* preenche um formulário disponível no *site* da LV com seus dados básicos como CNPJ, razão social e endereço, além de informar para qual *seller* deseja realizar seu cadastro. Os dados deste formulário são encaminhados para o time de performance da LV, o qual, finalmente, cadastra os dados na base de dados da LV para o *seller* específico em questão;

B. O *seller* envia para o time de performance da Empresa X uma base com os dados de cadastro de clientes que ainda não estão cadastrados na LV, mas, de acordo com a vontade do *seller*, devem ser incluídos na base a fim de se tornarem clientes;

II. **Dados de acesso:** a partir do momento em que um *shopper* tem seu cadastro realizado para um dado *seller*, está habilitado a comprar deste *seller* e, portanto, a navegar na LV. A partir do momento em que realiza *login* no ambiente virtual da LV, é acionado o evento de ***acesso*** capturado pelo código da plataforma. Neste evento, são capturados o CNPJ do *shopper* que acessou e a data de acesso;

III. **Dados de criação do carrinho:** dentro do ambiente da Loja Virtual, quando um *shopper* navega entre as páginas dos *sellers* a fim de procurar seus produtos, há a opção de adicionar produtos ao carrinho. Um carrinho é criado, portanto, a fim de guardar as informações dos produtos que o *shopper* deseja comprar. Cabe ressaltar que, caso o *shopper* não adicione nenhum produto ao seu carrinho, não é possível que um carrinho seja criado. Uma vez criado o carrinho, é capturado mais um evento no código da LV contendo as seguintes informações: produto adicionado, quantidade do produto adicionada, data de adição do produto, *GMV* associado ao produto, *seller* que oferece este produto e CNPJ do *shopper* que montou o carrinho. Caso o *shopper* não proceda com sua compra, o carrinho é, pois, abandonado e a LV captura o evento da data do abandono do carrinho;

IV. **Dados de criação de um pedido:** um carrinho se torna um pedido uma vez que o *shopper* clica no botão para “finalizar o pedido”. Deste modo, todas as informações contidas no carrinho são guardadas no banco de dados da LV e as características do pedido são, também, armazenadas. Os dados associados ao evento de criação de pedido são: conteúdo do carrinho, data de criação do pedido, CNPJ do *shopper* que finalizou o pedido, *sellers* cujos produtos estão dentro do carrinho do pedido, método de pagamento do pagamento, que pode variar entre boleto ou cartão de crédito e, por fim, o *status* do pedido, que varia entre:
   A. Pedido concluído: este *status* de pedido é obtido quando o pedido é entregue com sucesso no endereço do *shopper*;
   B. Pedido pendente: este *status* de pedido ocorre quando o pedido está em transporte até o endereço de entrega do *shopper* ou quando ainda aguarda aprovação do pagamento por parte do *seller*;
   C. Pedido cancelado: este *status* de pedido ocorre quando o pedido é cancelado por autoria do *seller*. No mapeamento de casos atual da Empresa X, um pedido é cancelado quando há algum problema com o pagamento realizado pelo *shopper*.

Por fim, é essencial ressaltar que o fluxo de navegação de um usuário dentro de uma plataforma, no contexto das *startups*, geralmente é capturado integralmente por ferramentas de análise de *websites* como, por exemplo, o *Google Analytics*. Este cenário, entretanto, não ocorre com a LV. Durante o processo de definição estratégica das implementações pelos desenvolvedores de *Software*, foi decidido não implementar nenhuma ferramenta de análise integral do *website*. Deste modo, para que se tenha visibilidade de qualquer ação do usuário, é necessário que esta ação esteja dentro de um dos quatro tipos de eventos capturados anteriormente. Ações fora destes eventos, portanto, não podem ser analisadas na situação atual da Empresa X.

## 5.4. DEFINIÇÃO DAS *FEATURES*

As *features* usadas para alimentar o modelo de ML têm origem nos dados da *Gold Layer* de acordo com os eventos capturados na LV. O processo de seleção das

*features* foi realizado em conjunto com todo o *squad* de análise de dados da LV, buscando englobar todos os dados disponíveis a fim de elencar as causas do *churn*. Neste processo, buscou-se, além de trazer informações genéricas sobre as interações do *shopper* com a LV dentro do período de análise, considerar algumas informações do passado recente e do passado remoto a fim de validar se, durante seu ciclo de vida na LV, certas características impactam mais na desistência do que outras. Cabe ressaltar que o período de análise para a aplicação do modelo de RF é o ano de 2021 até o dia 1 de outubro, data em que o modelo foi aplicado. As *features* serão detalhadas nas seções posteriores.

### 5.4.1. GMV total

Representa o valor, em unidades monetárias, transacionado pela plataforma virtual para cada *shopper* dentro do período de análise. Este tipo de *feature* inclui apenas o que a Empresa X considera como “GMV válido”, isto é, o GMV associado a pedidos cujo *status* final foi concluído sem ocasionar nenhum erro para o *shopper*. Por se tratar de um valor que resume unidades monetárias, esta *feature* representa uma variável contínua.

### 5.4.2. *Status* do primeiro pedido

Esta *feature* busca trazer ao modelo informações da primeira interação do *shopper* com a LV. Deste modo, ao final da computação do modelo, busca-se responder se as interações mais antigas prevalecem sobre as interações mais recentes quando se trata de *churn*. Além disso, cabe ressaltar que este é o *status* do primeiro pedido dentro do período de análise. As possíveis saídas para esta *feature* são: pedido concluído e pedido cancelado. O *status* de “pedido pendente” foi suprimido do modelo uma vez que representa pedidos em transporte. Os pedidos pendentes, em média, são resolvidos em três dias úteis, tornando-se, assim, pedidos concluídos ou cancelados.

### 5.4.3. Meio de pagamento do primeiro pedido

De maneira análoga à *feature* anterior, esta *feature* traz, para o primeiro pedido dentro do período de análise, o seu meio de pagamento. As saídas desta variável categórica podem ser: pedidos feitos via boleto ou pedidos feitos via cartão de crédito.

### 5.4.4. *Status* do último pedido

Esta *feature* busca, de acordo com a estratégia do time de análise de dados, trazer informações mais recentes sobre o *shopper*. Assim, dentro do período de análise, é escolhido o *status* do pedido mais recente para o dado *shopper*. É importante explicitar que, nos casos em que o *shopper* representa um *churn*, o último pedido no período também é o último pedido do *shopper* na LV dado que é um desistente. As saídas para esta variável categórica podem ser: pedido cancelado ou pedido concluído. Optou-se por omitir os pedidos pendentes pela mesma razão que foram omitidos os pedidos pendentes da *feature* de *status* do primeiro pedido.

### 5.4.5. Meio de pagamento do último pedido

Representa o meio de pagamento utilizado no pedido mais recente do *shopper* dentro do período de análise. As saídas para esta variável categórica são: pedidos feitos via boleto ou pedidos feitos via cartão de crédito.

### 5.4.6. GMV médio por pedido

A proposta, ao incluir esta *feature* no modelo, é considerar os diferentes perfis de cliente existentes na LV. Caso fosse considerado apenas o GMV total, os *shoppers* com mais tempo de vida e frequência de compra poderiam enviesar a escolha das *features* pelo modelo. Assim, esta *feature* representa, para cada *shopper*, o GMV total dividido pelo número de pedidos concluídos. Este termo, no contexto das *startups*, muitas vezes é chamado de *ticket médio*. Trata-se, portanto, de uma variável contínua medida em unidades monetárias e representa, em média, quanto um *shopper* gasta em um pedido na Loja Virtual.

### 5.4.7. Estado

Quando um *shopper* é cadastrado na base de dados da LV, o time de análise de dados consegue ter acesso ao endereço do dado *shopper*. Este endereço é usado na LV para direcionar as entregas dos produtos. Esta *feature*, portanto, representa a Unidade Federativa do Brasil em que o *shopper* pode ser localizado. Assim, esta variável categórica pode assumir um entre os seguintes resultados: Acre (AC), Alagoas (AL), Amazonas (AM), Amapá (AP), Bahia (BA), Ceará (CE), Distrito Federal (DF), Espírito Santo (ES), Goiás (GO), Maranhão (MA), Mato Grosso (MT), Mato Grosso do Sul (MS), Minas Gerais (MG), Pará (PA), Paraíba (PB), Paraná (PR), Pernambuco (PE), Piauí (PI), Rio de Janeiro (RJ), Rio Grande do Norte (RN), Rio Grande do Sul (RS), Rondônia (RO), Roraima (RR), Santa Catarina (SC), São Paulo (SP), Sergipe (SE) e Tocantins (TO).

### 5.4.8. Número de acessos à plataforma por dia de compra

Esta *feature* busca alimentar o modelo de RF com informações de fluxo de navegação na plataforma, representando o total de acessos realizados no período de análise dividido pelo número de dias em que o *shopper* comprou. Idealmente, a Empresa X, do ponto de vista de usabilidade da plataforma, deseja que o *shopper* realize apenas um acesso para concluir sua compra, mas, pode haver casos em que problemas na plataforma faça, com que o usuário precise acessar mais de uma vez a LV a fim de concluir um pedido. Esta também é uma variável contínua.

### 5.4.9. Número total de carrinhos abandonados

Esta *feature* representa, para cada *shopper* dentro do período de análise, o número total de carrinhos abandonados. Cabe ressaltar que um carrinho é abandonado quando há pelo menos um produto em seu interior, mas o pedido não é finalizado pelo *shopper*. Esta variável é contínua.

### 5.4.10. Número de *sellers* cadastrados

Representa, para cada *shopper*, o número de *sellers* em que o *shopper* está habilitado a comprar seguindo as regras de cadastro apresentadas anteriormente. Vale lembrar que, no mês de outubro de 2021, a LV conta com 21 *sellers* disponíveis em sua plataforma, mas nem todo *shopper* está habilitado a comprar de todos os *sellers*. Além disso, um *shopper* pode ser cadastrado em novos *sellers* ao longo do ano pela Empresa X, o que faz com que esta *feature* represente o número de *sellers* em que o *shopper* está cadastrado no mês de realização deste trabalho, ou seja, em outubro de 2021. Esta é uma variável categórica que representa um número *x* descrito segundo o conjunto $\{x \in \mathbb{N} \mid 1 \leq x \leq 21\}$.

### 5.4.11. Número de pedidos cancelados

Representa, para cada *shopper*, o número total de pedidos cancelados, isto é, que não foram finalizados com sucesso. Esta variável contínua busca trazer ao modelo informações sobre falta de fluidez no fluxo de navegação como uma das hipóteses do *churn*.

### 5.4.12. Número médio de *sellers* por pedido

É possível, dentro do contexto da LV, que um pedido contenha produtos de *sellers* diferentes. A Empresa X atribui a este evento o nome de *cross-sell*. Esta *feature*, portanto, representa, para um dado *shopper*, a divisão entre a soma do número de *sellers* em todos os pedidos pelo número total de pedidos. Em outras palavras, esta variável contínua representa, em média, de quantos *sellers* o *shopper* compra a cada pedido que realiza.

### 5.4.13. Frequência de compra

Esta *feature* corresponde ao código de frequência de compra do *shopper* no dado período de análise de acordo com as regras estabelecidas nas seções anteriores de definição da métrica do *churn*. Esta é, pois, uma variável categórica,

representando um número natural entre 0 (primeira compra) e 6 (frequência semestral).

Tabela 4 - Classificação das variáveis que compõem as *features*

| Id da ***feature*** | Nome da ***feature*** | Tipo de variável |
|---|---|---|
| 1 | GMV total | Contínua |
| *2* | *Status* do primeiro pedido | Categórica |
| 3 | Meio de pagamento do primeiro pedido | Categórica |
| *4* | *Status* do último pedido | Categórica |
| 5 | Meio de pagamento do último pedido | Categórica |
| 6 | GMV médio por pedido | Contínua |
| 7 | Estado | Categórica |
| 8 | Número de acessos à plataforma por dia | Contínua |
| 9 | Número total de carrinhos abandonados | Categórica |
| 10 | Número de *sellers* cadastrados | Categórica |
| 11 | Número de pedidos cancelados | Categórica |
| 12 | Número médio de *sellers* por pedido | Categórica |
| 13 | Frequência de compra | Categórica |

Fonte: Elaborada pelo autor

## 5.5. DESCRIÇÃO DA BASE INICIAL E MODELAGEM DOS DADOS

Para servir como entrada para o modelo de *Random Forest* será usada uma base composta por *shoppers* que realizaram pelo menos uma compra na Loja Virtual no ano de 2021 até o dia 1 de outubro de 2021. A fim de preservar o CNPJ dos *shoppers* por se tratar de informação confidencial, o presente trabalho converte os CNPJ's dos *shoppers* em uma variável aqui chamada de ***shopper_id***, a qual corresponde a um número inteiro. A base de dados inicial, então, é composta pelo ***shopper_id*** e pelo campo ***b_isChurn***, um *booleano* que indica se o dado *shopper* corresponde a um *churn*, indicando, em caso de *churn*, o valor "***True***" e, caso contrário, o valor "***False***". Este último campo é a variável dependente categórica do

modelo de ML. Além disso, a base de dados inicial será composta por 13 outras colunas que representarão as variáveis independentes do modelo, isto é, as 15 *features*. A Tabela 5 representa o formato inicial desta base.

Tabela 5 - Distribuição dos *shoppers* em termos de *churn*

| **Número de *shoppers*** | ***b_isChurn*** |
|---|---|
| 6238 | *True* |
| 11339 | *False* |

Fonte: Elaborada pelo autor

### 5.5.1. Modelagem de variáveis contínuas

O modelo de *Random Forest*, de acordo com a revisão bibliográfica, é sensível à cardinalidade das variáveis dependentes, isto é, ao elencar as *features* mais importantes para a classificação, o modelo de RF tende a selecionar as variáveis com alta cardinalidade devido ao processo de aprendizagem do modelo se valer das árvores de decisão. Deste modo, ao modelar as *features* e a fim de não enviesar o algoritmo, é necessário que haja um balanceamento de cardinalidade entre todas as variáveis dependentes. Neste caso, as variáveis com maior cardinalidade possível são as variáveis contínuas e, neste caso, a fim de reduzir a cardinalidade de uma variável contínua, será usado um processo de categorização de variáveis contínuas.

Este processo consiste em, para cada *feature* que representa uma variável contínua, definir grupos discretos e limites superiores e inferiores para cada grupo de modo que as variáveis contínuas sejam alocadas em cada grupo de acordo com os limites. O processo de categorização de variáveis contínuas, neste trabalho, buscando reduzir sua cardinalidade, será composto das seguintes etapas:

I. Definir um conjunto de variáveis independentes contínuas $C$ com $N$ elementos. Cada elemento de $C$ é representado por $x_i$ com $i$ sendo um número natural entre 1 e $N$;
II. Criar $j$ categorias, bem como seu respectivo limite inferior ($I_j$) e superior ($S_j$) de modo que $j$ seja um número natural maior ou igual a 1 e menor ou igual a 10. A decisão de limitar o número de categorias em 10 é embasada em

manter a cardinalidade das variáveis do conjunto *C* baixa com relação ao total de variáveis do problema atual;

III. Categorizar cada um dos *N* elementos de *C* dentro das *j* categorias de acordo com os respectivos limites superior e inferior e calcular o número de elementos $N_j$ alocado em cada uma das categorias. A categorização dos elementos de *C* na *j*-ésima categoria é realizada pela inequação $I_j \leq x_i < S_j$;

IV. Identificar a categoria com o maior número de elementos alocados e guardar este valor em uma variável $N_{max}$;

V. Para cada uma das *j* categorias, avaliar se $N_j < 15\% \, . \, N_{max}$;

   A. Caso alguma das *j* categorias satisfaça a inequação anterior, rearranjar $I_j$ e $S_j$ convenientemente até que a inequação seja satisfeita;

   B. Caso nenhuma das *j* categorias satisfaça a inequação anterior, concluir o processo;

VI. Iterar o processo até que a inequação do passo *V* seja satisfeita.

O processo descrito anteriormente garante que serão criadas categorias para as variáveis independentes contínuas de modo que cada categoria não tenha um número de elementos tão discrepante quando comparada com outras categorias, pois a lógica de criação limita a diferença entre categorias como sendo 15% do número de elementos da categoria com mais elementos. Por fim, é importante ressaltar que valores nulos presentes em alguma das variáveis independentes foram convertidos para zero a fim de garantir a sanidade dos dados para o modelo de ML.

#### 5.5.1.1. Modelagem de GMV total

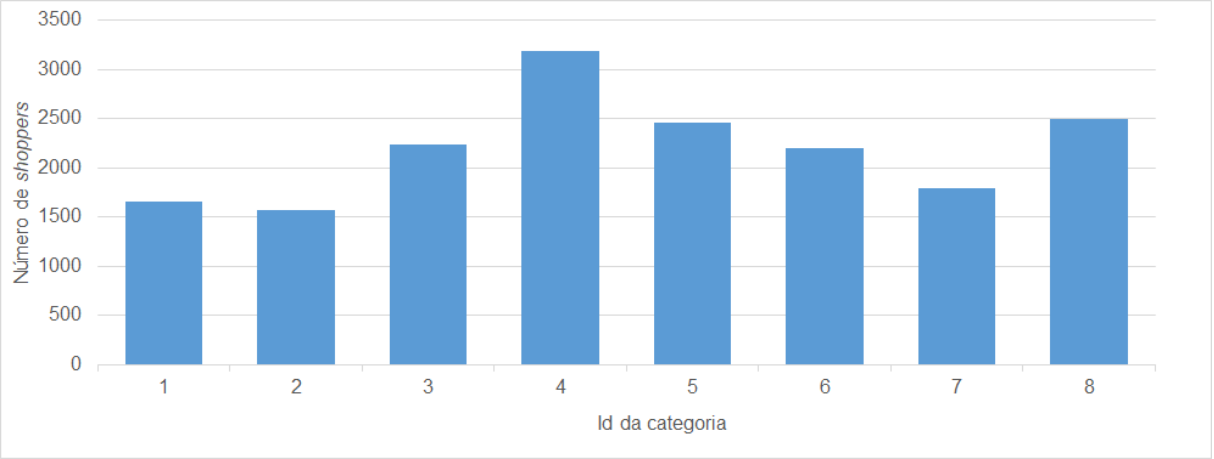
O processo de redução de cardinalidade para a *feature* de GMV total gerou 8 grupos cujos limites superior e inferior são apresentados na Tabela 6. O Gráfico 8 mostra a distribuição dos *shoppers* de acordo com cada categoria.

Tabela 6 - Categorias da *feature* GMV total

| Id da categoria para a *feature* GMV total | Limite inferior (R$) | Limite superior (R$) |
|---|---|---|
| 1 | 0 | 2500 |
| 2 | 2500 | 5000 |
| 3 | 5000 | 10000 |
| 4 | 10000 | 25000 |
| 5 | 25000 | 50000 |
| 6 | 50000 | 100000 |
| 7 | 100000 | 200000 |
| 8 | 200000 | ∞ |

Fonte: Elaborada pelo autor

Gráfico 8 - Distribuição dos *shoppers* nas categorias de GMV total




Fonte: Elaborado pelo autor

5.5.1.2. Modelagem de GMV médio por pedido

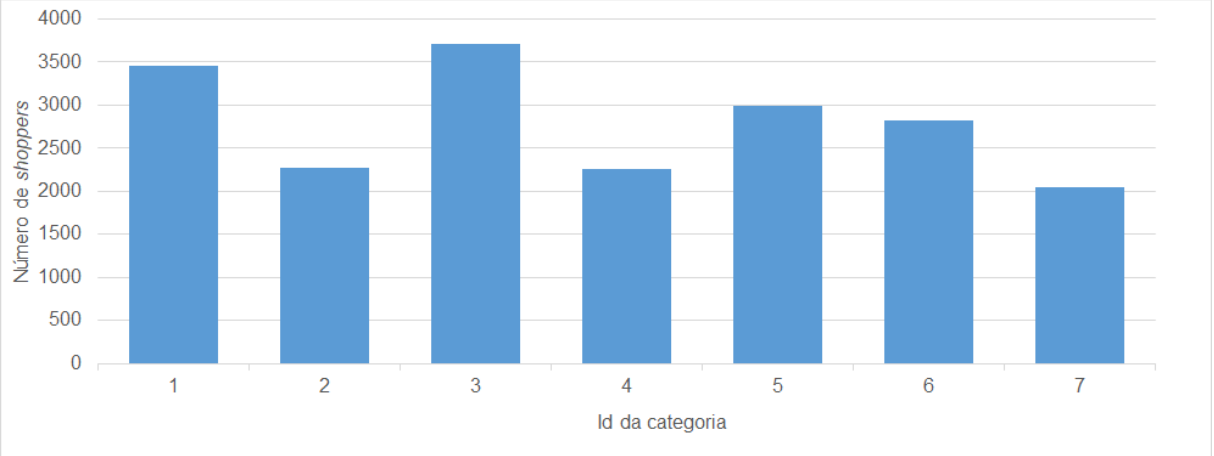
A redução de cardinalidade para a *feature* de GMV médio por pedido (*ticket médio*) gerou 7 grupos cujos limites superior e inferior são apresentados na Tabela 7. Além disso, no Gráfico 9 é possível identificar a distribuição dos *shoppers* de acordo com cada categoria.

Tabela 7 - Categorias da *feature* GMV médio por pedido

| ***Id* da categoria para a *feature* GMV médio por pedido** | **Limite inferior (R$/pedido)** | **Limite Superior (R$/pedido)** |
|---|---|---|
| 1 | 0 | 500 |
| 2 | 500 | 1000 |
| 3 | 1000 | 2000 |
| 4 | 2000 | 3000 |
| 5 | 3000 | 5000 |
| 6 | 5000 | 8000 |
| 7 | 8000 | ∞ |

Fonte: Elaborada pelo autor

Gráfico 9 - Distribuição dos *shoppers* nas categorias de GMV médio por pedido




Fonte: Elaborado pelo autor

5.5.1.3. Modelagem de número de acessos por pedido

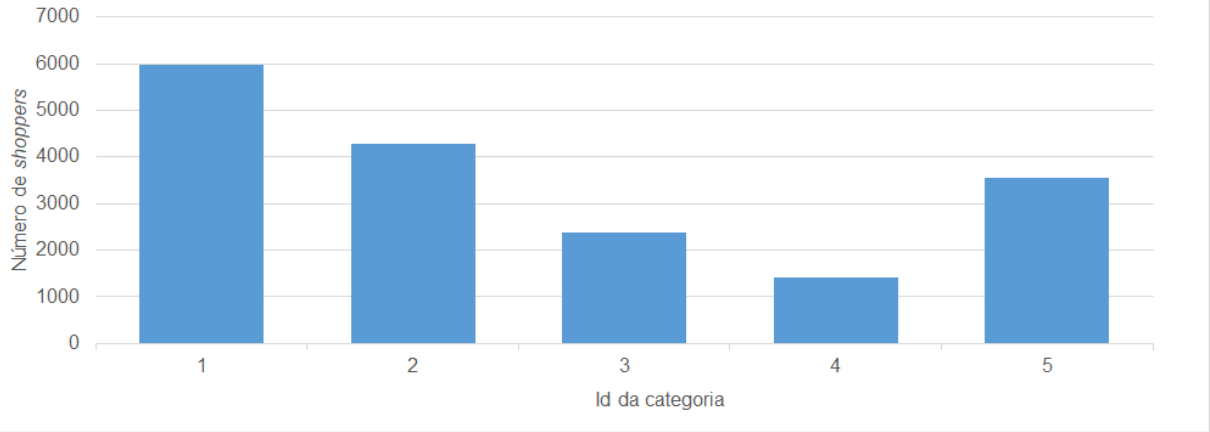
O método de redução de cardinalidade para a *feature* de número de acessos por pedido gerou 5 categorias distintas, com limites superior e inferior na Tabela 8. Além disso, no Gráfico 10 é possível identificar a distribuição dos *shoppers* para cada uma das categorias.

Tabela 8 - Categorias da *feature* número de acessos por pedido

| ***Id* da categoria para a *feature* número de acessos por pedido** | **Limite inferior (acessos/pedido)** | **Limite superior (acessos/pedido)** |
|---|---|---|
| 1 | 0 | 2 |
| 2 | 2 | 3 |
| 3 | 3 | 4 |
| 4 | 4 | 5 |
| 5 | 5 | ∞ |

Fonte: Elaborada pelo autor

Gráfico 10 - Distribuição dos *shoppers* nas categorias de número de acessos por pedido




Fonte: Elaborado pelo autor

### 5.5.2. Modelagem de variáveis categóricas

De maneira análoga ao processo de redução da cardinalidade de variáveis contínuas, as variáveis categóricas deste modelo, caso sejam compostas por diversas categorias, como por exemplo a *feature* de número de carrinhos abandonados, também devem ter seu número de categorias seguindo as mesmas regras aplicadas às variáveis independentes contínuas. No caso de variáveis categóricas que representam números naturais, os possíveis valores nulos na base de dados serão convertidos em zero. Para as variáveis que não puderem ser expressas em forma de números, como por exemplo, a *feature* de unidade

federativa, os valores nulos serão convertidos no valor com a maior frequência observada entre todos.

As variáveis categóricas não numérica terão suas categorias convertidas para números inteiros a fim de manter o padrão dentro de toda a base de dados que alimentará o modelo. Não é necessário modelar a frequência de compra visto que esta *feature* já categoriza os *shoppers* de acordo com as regras da seção 4.

#### 5.5.2.1. Modelagem de número médio de *sellers* por pedido

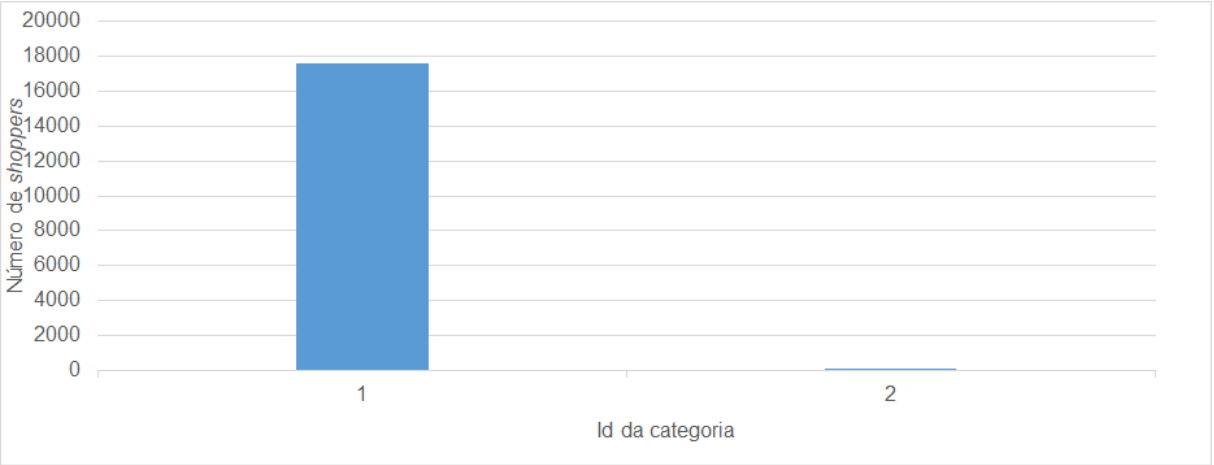
Ao realizar a modelagem desta *feature*, foi possível perceber que a maioria dos *shoppers* da base de dados inicial comprava em média de apenas um único *seller*. Esta análise de *cross-sell* não foi explorada pela Empresa X antes de trazer o problema de *churn* ao time de dados. Deste modo, devido à impossibilidade em categorizar os *shoppers* em mais de uma categoria expressiva, foi decidido eliminar esta *feature* do modelo dado que poderia *enviesar* o algoritmo. Apenas para se ter registro, a Tabela 9 mostra a criação das categorias e o Gráfico 11, os *shoppers* alocados. Este caso específico de eliminação de uma *feature* durante a modelagem é algo comum nos modelos de ML e é de vital importância para garantir a sanidade do modelo.

Tabela 9 - Categorias da *feature* número de *sellers* por pedido

| **Id da categoria para a *feature* número de *sellers* por pedido** | **Limite inferior (*sellers*/pedido)** | **Limite Superior (*sellers*/pedido)** |
|---|---|---|
| 1 | 1 | 2 |
| 2 | 2 | ∞ |

Fonte: Elaborada pelo autor

Gráfico 11 - Distribuição dos *shoppers* nas categorias de número de *sellers* por pedido




Fonte: Elaborado pelo autor

5.5.2.2. Modelagem de número total de carrinhos abandonados

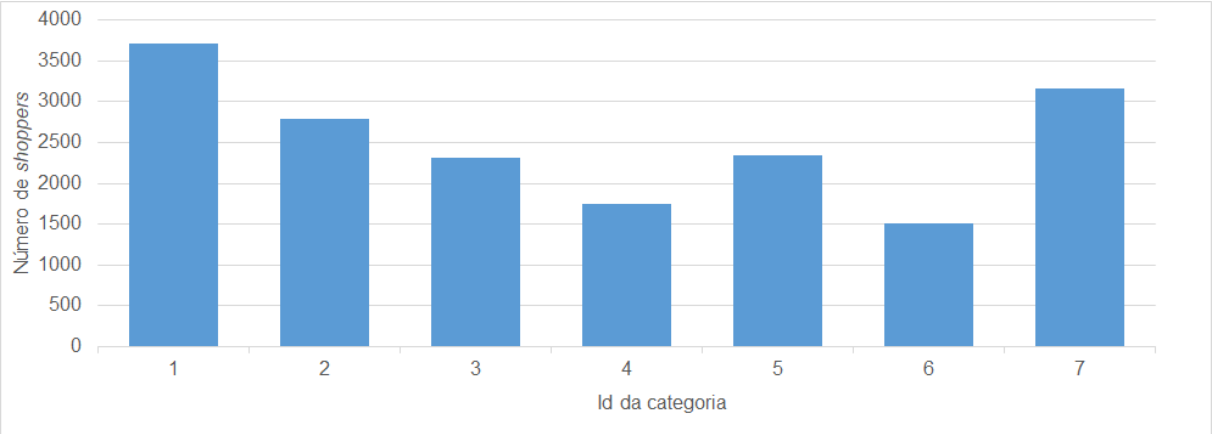
O processo de redução de cardinalidade para a *feature* de número total de carrinhos abandonados gerou 7 grupos com limites superior e inferior apresentados na Tabela 10. O Gráfico 12 mostra a distribuição dos *shoppers* de acordo com cada categoria.

Tabela 10 - Categorias da *feature* número total de carrinhos abandonados

| ***Id* da categoria para a *feature* número de carrinhos abandonados** | **Limite inferior (número de carrinhos)** | **Limite superior (número de carrinhos)** |
|---|---|---|
| 1 | 0 | 1 |
| 2 | 1 | 2 |
| 3 | 2 | 3 |
| 4 | 3 | 4 |
| 5 | 4 | 6 |
| 6 | 6 | 8 |
| 7 | 8 | ∞ |

Fonte: Elaborada pelo autor

Gráfico 12 - Distribuição dos *shoppers* nas categorias de número total de carrinhos abandonados




Fonte: Elaborado pelo autor

5.5.2.3. Modelagem de total de pedidos cancelados

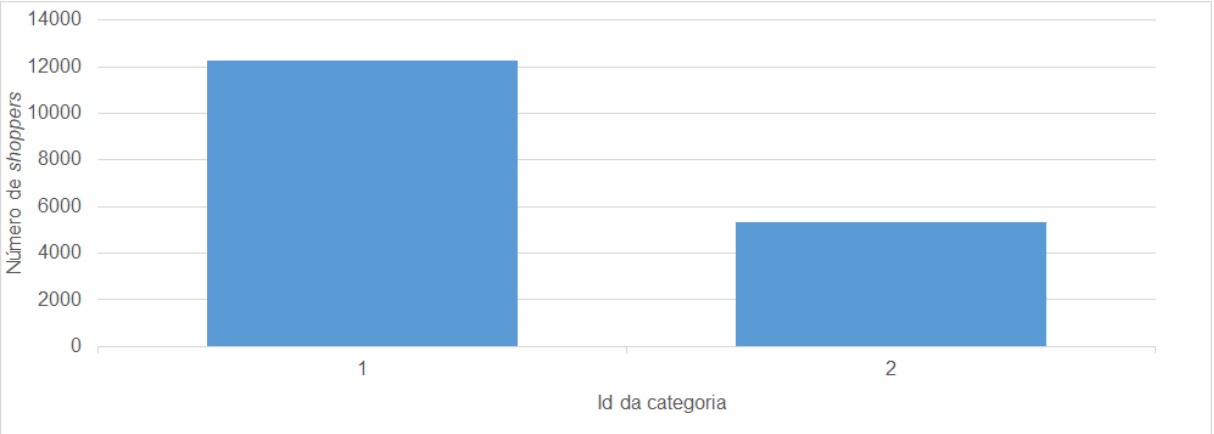
A redução de cardinalidade para esta *feature* originou apenas 2 grupos cujos limites superior e inferior são apresentados na Tabela 11. O Gráfico 13 apresenta a distribuição de *shoppers* em cada categoria. Cabe ressaltar que, neste caso, a categoria 1 representa aqueles *shoppers* que não cancelaram nenhum de seus pedidos.

Tabela 11 - Categorias da *feature* total de pedidos cancelados

| ***Id* da categoria para a *feature* número de pedidos cancelados** | **Limite inferior (número de pedidos)** | **Limite Superior (número de pedidos)** |
|---|---|---|
| 1 | 0 | 1 |
| 2 | 1 | ∞ |

Fonte: Elaborada pelo autor

Gráfico 13 - Distribuição dos *shoppers* nas categorias de total de pedidos cancelados




Fonte: Elaborado pelo autor

5.5.2.4. Modelagem de *status* do primeiro pedido

No caso do *status* do primeiro pedido, só é possível esperar dois resultados. Assim, caso o *shopper* tenha tido seu primeiro pedido cancelado, este *shopper* será alocado na categoria 0 desta *feature* e, em caso contrário, será direcionado para a categoria 1. Na Tabela 12, pode-se observar o número de *shoppers* em cada categoria.

Tabela 12 - Categorias e distribuição de *shoppers* da *feature status* do primeiro pedido

| ***Id* da categoria para a *feature status* do primeiro pedido** | Variável independente correspondente | Número de *shoppers* |
|---|---|---|
| 1 | Pedido concluído | 14149 |
| 0 | Pedido cancelado | 3428 |

Fonte: Elaborada pelo autor

5.5.2.5. Modelagem do meio de pagamento do primeiro pedido

Neste tipo de modelagem, também há apenas dois resultados esperados. Assim, a redução de cardinalidade para este tipo de variável categórica não numérica cria duas categorias. Caso a primeira compra do *shopper* tenha sido

realizada via cartão de crédito, este *shopper* entrará para a categoria 1 e, caso a compra tenha sido realizada via boleto, o cliente é alocado para a categoria 0. Observando a Tabela 13, é possível analisar a distribuição de usuários em cada grupo.

Tabela 13 - Categorias e distribuição de *shoppers* da *feature* meio de pagamento do primeiro pedido

| ***Id* da categoria para a *feature status* do primeiro pedido** | **Variável independente correspondente** | **Número de *shoppers*** |
|---|---|---|
| 1 | Pedido pago via cartão de crédito | 7278 |
| 0 | Pedido pago via boleto | 10299 |

Fonte: Elaborada pelo autor

5.5.2.6. Modelagem do *status* do último pedido

Para o caso desta *feature*, de maneira análoga à *feature* de *status* do primeiro pedido, serão criadas duas categorias representadas por números inteiros com 0 representando um *shopper* cujo último pedido foi cancelado e 1 em caso contrário. A Tabela 14 apresenta a distribuição dos *shoppers* dentro de cada uma das duas categorias.

Tabela 14 - Categorias e distribuição de *shoppers* da *feature status* do último pedido

| ***Id* da categoria para a *feature* meio de pagamento do primeiro pedido** | **Variável independente correspondente** | **Número de *shoppers*** |
|---|---|---|
| 1 | Pedido concluído | 13963 |
| 0 | Pedido cancelado | 3614 |

Fonte: Elaborada pelo autor

#### 5.5.2.7. Modelagem do meio de pagamento do último pedido

De maneira análoga à seção anterior, a última compra do *shopper* é analisada em duas categorias expressas na Tabela 15.

Tabela 15 - Categorias e distribuição de *shoppers* da *feature* meio de pagamento do último pedido

| ***Id* da categoria para a *feature* meio de pagamento do último pedido** | **Variável independente correspondente** | **Número de *shoppers*** |
|---|---|---|
| 1 | Pedido pago via boleto | 3488 |
| 0 | Pedido pago via boleto | 14089 |

Fonte: Elaborada pelo autor

#### 5.5.2.8. Modelagem de estado

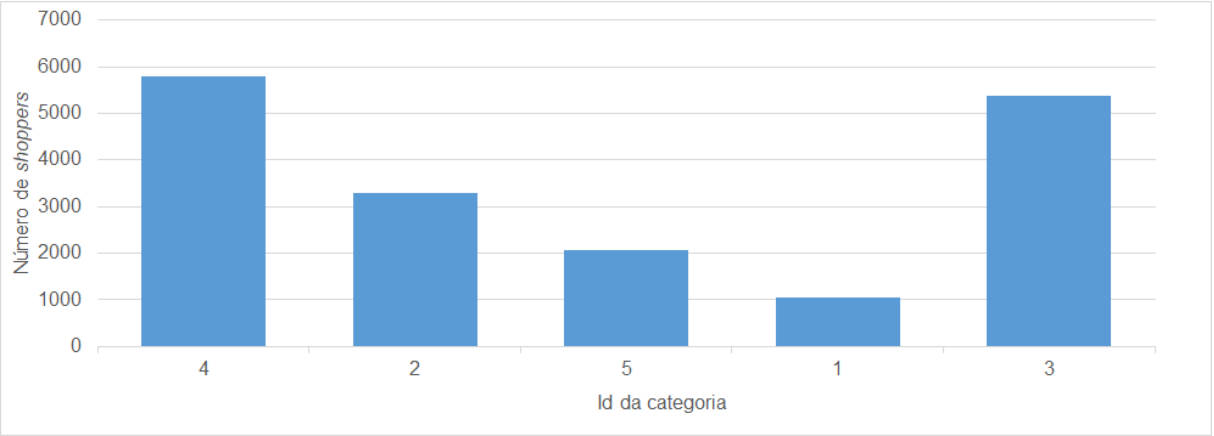
A fim de reduzir a cardinalidade da variável independente que representa o estado em que o *shopper* se localiza, a *feature* de estado foi agrupada segundo as regiões Norte, Nordeste, Sudeste, Sul e Centro-Oeste do Brasil. Deste modo, a Tabela 16 apresenta o id da categoria criada e o Gráfico 14 a distribuição do número de *shoppers* em cada categoria.

Tabela 16 - Categorias da *feature* estado

| ***Id* da categoria para a *feature* número de *sellers* cadastrados** | **Região brasileira representada pela categoria** | **Estados englobados pela categoria** |
|---|---|---|
| 1 | Norte | Amapá, Amazonas, Pará, Rondônia, Roraima e Tocantins |
| 2 | Nordeste | Alagoas, Bahia, Ceará, Maranhão, Paraíba, Pernambuco, Piauí, Rio Grande do Norte e Sergipe |
| 3 | Sul | Paraná, Santa Catarina e Rio Grande do Sul |
| 4 | Sudeste | São Paulo, Rio de Janeiro, Espírito Santo e Minas Gerais |
| 5 | Centro-Oeste | Mato Grosso, Mato Grosso do Sul, Goiás e Distrito Federal |

Fonte: Elaborada pelo autor

Gráfico 14 - Distribuição de *shoppers* nas categorias de estado




Fonte: Elaborado pelo autor

5.5.2.9. Modelagem de *sellers* cadastrados

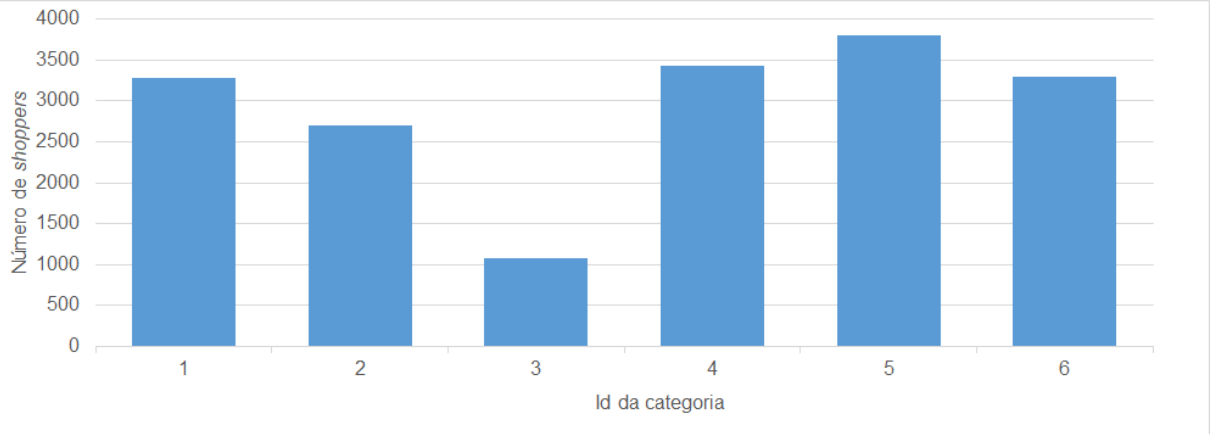
O número de *sellers* cadastrado para cada *shopper* também é tratado como uma variável categórica dado que só pode assumir a forma de um número natural. As categorias criadas de acordo com o método de redução da cardinalidade são apresentadas na Tabela 17 e, no Gráfico 15, é possível perceber quantos *shoppers* se enquadram dentro de cada categoria.

Tabela 17 - Categorias da *feature sellers* cadastrados

| ***Id* da categoria para a *feature* número de *sellers* cadastrados** | **Limite inferior (número de *sellers*)** | **Limite superior (número de *sellers*)** |
|---|---|---|
| 1 | 0 | 2 |
| 2 | 2 | 10 |
| 3 | 10 | 12 |
| 4 | 12 | 15 |
| 5 | 15 | 17 |
| 6 | 17 | ∞ |

Fonte: Elaborada pelo autor

Gráfico 15 - Distribuição de *shoppers* nas categorias de *sellers* cadastrados




Fonte: Elaborado pelo autor

## 5.6. PARÂMETROS DAS FUNÇÕES EM *PYSPARK*

As funções utilizadas para aplicar o modelo de *Random Forest* aos dados da Empresa X pertencem a uma biblioteca do *Python* chamada de *sklearn*. Serão usadas duas funções nativas desta biblioteca: *RandomForestClassifier*, *train_test_split* e *metrics*. A primeira função se vale das árvores de decisão para aplicar a classificação dos dados, enquanto que a segunda serve para segmentar a base de dados inicial em dados de teste e dados de treino. A terceira função, por sua vez, é usada para definir a importância das *features* e a precisão do modelo

gerado. Estas funções, bem como seus respectivos parâmetros serão detalhados a seguir.

### 5.6.1. A função *train_test_split*

Esta função é responsável por garantir que o processo de aprendizado do modelo de *Random Forest* seja supervisionado, ou seja, uma parte dos dados que servirão de entrada para o modelo será separada em **dados de treino** e outra parte, em **dados de teste**. Os dados de treino são formados pelas variáveis independentes e pela variável de saída *b_isChurn* que alimentarão o modelo de RF. Assim, as árvores de decisão, através do método de *bootstrap aggregation*, serão formadas pelos dados de treino.

Uma vez treinado o modelo, os dados de teste serão usados para medir a precisão do modelo. Deste modo, oculta-se do modelo as variáveis de saída conhecidas dos dados de teste e permite-se que o modelo de RF classifique estes dados em termos de *churn* ou não *churn*. Por fim, comparam-se as classificações do modelo treinado com os valores reais das variáveis de saída dos dados de teste.

Esta função apresenta apresenta apenas dois parâmetros que serão usados para dividir os dados do modelo:

- ***test_size*:** este parâmetro é um número racional entre 0 e 1 e representa a porcentagem dos dados iniciais que deve ser usada como teste. Assim, por exemplo, quando *test_size* é igual a 0.2, 20% dos dados iniciais serão usadas para teste e, consequentemente, 80% dos dados será usada para treino.

### 5.6.2. A função *RandomForestClassifier*

Esta função receberá os dados divididos em treino e teste pela *train_test_split* e aplicará o método de *bagging* para criar as AD's e classificar os dados. Os parâmetros desta função utilizados no presente trabalho são:

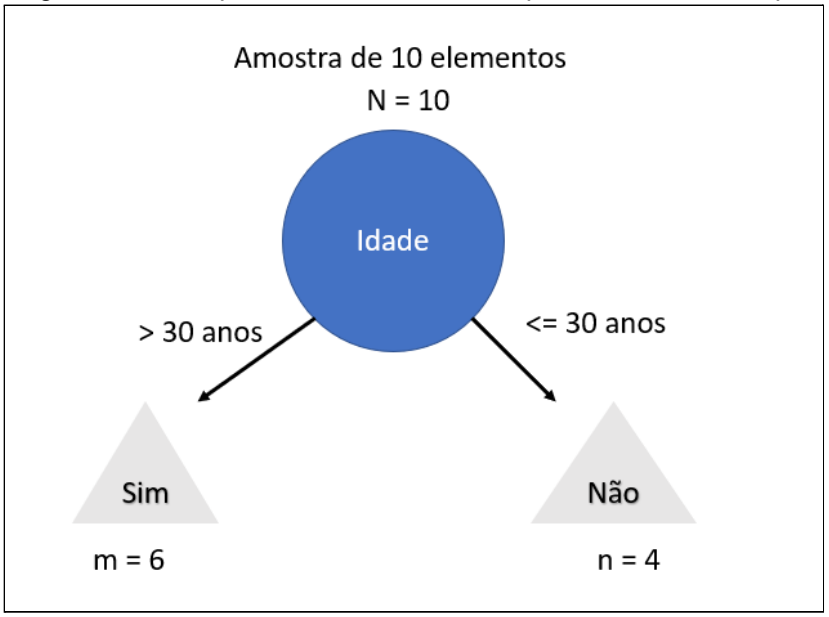
- ***n_estimators*:** representa o número de AD's que o modelo usará para classificar os dados.Este parâmetro é um número inteiro maior que 0;
- ***criterion*:** o critério usado pelo modelo de RF para segmentar os dados de maneira ótima em cada AD. Pode assumir um valor entre *gini*, que

encontra a AD ótima através do índice de Gini ou *entropy*, que constrói a AD ótima usando a entropia dos nós da AD;

- ***max_depth*:** representa a profundidade máxima de cada AD do modelo, ou seja, o número máximo de nós que criam novas subdivisões dentro de cada AD;
- ***max_features*:** este parâmetro define o número máximo de *features* usado para cada AD. Pode assumir um entre três valores: *sqrt* ou *log*2. O primeiro valor determina que o número máximo de *features* em cada AD é igual à raiz quadrada do número de *features* do modelo. O segundo valor determina que o número máximo de *features* em cada AD equivale ao logaritmo na base 2 do número de *features* do modelo.

Caso o critério de seleção da AD ótima seja igual a ***entropy***, a AD ótima usada pelo modelo de RF será aquela com o maior ganho de informação usando o índice de Gini. A Figura 11 apresenta um exemplo de nó em uma AD para o qual será calculada sua entropia. Será usada como exemplo uma amostra de 10 elementos que foram divididos segundo o critério “idade” em dois grupos de acordo sua propensão a responder ou não uma determinada comunicação.

Figura 11 - Exemplo de Árvore de Decisão para cálculo de entropia



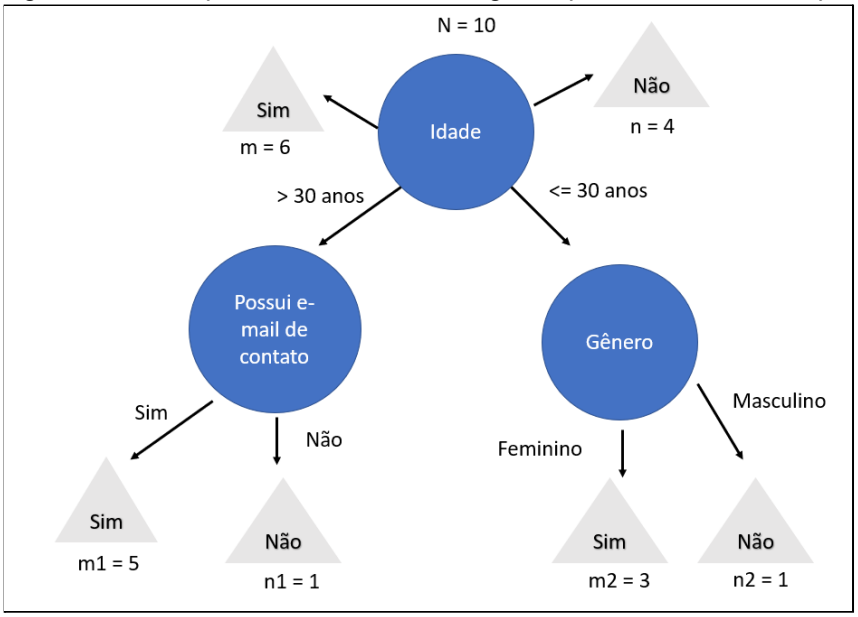

Fonte: Elaborada pelo autor

Assim, o cálculo da entropia do nó da Figura 11 é dada por:

$$H(s) = -\frac{6}{10}.log_2(\frac{6}{10}) - \frac{4}{10}.log_2(\frac{4}{10}) = 0,970951$$

A Figura 12 retoma o exemplo da propensão de um cliente responder a uma comunicação, mas traz além disso, outras segmentações. No caso, o gênero do cliente e a posse ou não de e-mail também são incluídos na AD como variáveis independentes para compor a AD cujo ganho de informação deseja-se encontrar.

Figura 12 - Exemplo de AD com três categorias para cálculo de entropia




Fonte: Elaborada pelo autor

Desde modo, para a AD da Figura 12, primeiramente seria preciso calcular as três entropias associadas a cada um dos nós como sendo:

$$Nó\ raiz:\ H_o(s) = -\frac{6}{10}.log_2(\frac{6}{10}) - \frac{4}{10}.log_2(\frac{4}{10}) = 0,970951$$

$$Nó\ interno\ de\ "email":\ H_1(s) = -\frac{5}{6}.log_2(\frac{5}{6}) - \frac{1}{6}.log_2(\frac{1}{6}) = 0,650022$$

$$Nó\ interno\ de\ "gênero":\ H_2(s) = -\frac{3}{4}.log_2(\frac{3}{4}) - \frac{1}{4}.log_2(\frac{1}{4}) = 0,811278$$

Uma vez calculadas as entropias dos nós, por fim, pode-se encontrar o ganho de informação desta AD hipotética *I* de acordo com a fórmula:

$$I = H_o(s) - \frac{m_1+n_1}{N}H_1(s) - \frac{m_2+n_2}{N}H_2(s) \quad \therefore$$

$$I = 0,970951 - 0,390013 - 0,324511 = 0,256427$$

Caso o parâmetro *criterion* seja igual a *gini*, o ganho de informação usado para definir a AD ótima dentro do modelo de RF será baseado no índice de Gini. Deste modo, retomando o caso exposto na Figura 11, o cálculo do índice de Gini para o nó evidenciado seria dado por:

$$Gini(s) = 1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2 = 0,48$$

Assim, para calcular o ganho de informação de Gini para o caso exposto na Figura 12, por exemplo, seria necessário calcular os índices de Gini dos três nós segundo as seguintes passagens:

$$Nó\ raiz:\ Gini_o(s) = 1 - \left[\left(\frac{6}{10}\right)^2 + \left(\frac{4}{10}\right)^2\right] = 0,48$$

$$Nó\ interno\ de\ "email":\ Gini_1(s) = 1 - \left[\left(\frac{5}{6}\right)^2 + \left(\frac{1}{6}\right)^2\right] = 0,277778$$

$$Nó\ interno\ de\ "gênero":\ Gini_2(s) = 1 - \left[\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right] = 0,375$$

A partir dos índices de cada nó, o ganho de informação da AD presente na Figura 12 seria:

$$I_{Gini} = Gini_o(s) - \frac{m_1+n_1}{N}Gini_1(s) - \frac{m_2+n_2}{N}Gini_2(s) \quad \therefore$$

$$I_{Gini} = 0,48 - 0,1666668 - 0,15 = 0,163333$$

### 5.6.3. A função *metrics*

Esta função é usada para medir a precisão do modelo de RF gerado, além de definir a importância das *features* que alimentaram o modelo. A importância das *features* nesta função em específico é calculada através do Decréscimo Médio de

Impureza. Em outras palavras, uma *feature* muito importante representa a variável independente do modelo que, caso fosse retirada de todas as AD's que a usaram para categorizar a amostra do *bootstrap*, tornaria estas árvores mais impuras.

Uma *feature* pouco importante, por sua vez, caso fosse retirada de alguma AD que a utiliza, pouco impactaria na variação da pureza desta árvore. A função *metrics* da biblioteca *sklearn* apresenta a importância das *features* em uma escala normalizada de modo que a soma das importâncias sempre seja igual a 100%. Assim, o valor absoluto da importância em si é irrelevante. O mais importante é analisar o valor da importância de uma *feature* quando comparado com os outros valores das importâncias das outras *features*.

## 6. RESULTADOS

Para escolher os valores mais convenientes para cada um dos parâmetros apresentados nas seções anteriores, o presente trabalho propõe variar os valores de cada um dos parâmetros e, para cada combinação de diferentes valores, medir a precisão do modelo gerado. O modelo com maior precisão, bem como seus parâmetros, serão escolhidos para basear o algoritmo de *Random Forest* e gerar a importância das *features*. A Tabela 18 mostra os 10 melhores resultados das iterações dos valores dos parâmetros.

Tabela 18 - Combinações de parâmetros com as dez maiores precisões de modelo

| ***test_size*** | ***n_estimators*** | ***max_features*** | ***criterion*** | ***max_depth*** | **precisão do modelo** |
|---|---|---|---|---|---|
| 10% | 200 | log2 | gini | 13 | 78,27% |
| 10% | 400 | sqrt | gini | 13 | 78,21% |
| 20% | 800 | sqrt | gini | 16 | 78,04% |
| 20% | 800 | log2 | gini | 16 | 77,72% |
| 10% | 800 | log2 | entropy | 20 | 77,70% |
| 20% | 200 | log2 | entropy | 13 | 77,70% |
| 20% | 800 | log2 | entropy | 13 | 77,64% |
| 10% | 400 | log2 | entropy | 20 | 77,59% |
| 20% | 600 | sqrt | gini | 13 | 77,55% |
| 10% | 200 | sqrt | gini | 20 | 77,53% |

Fonte: Elaborada pelo autor

Observando os dados da primeira linha da Tabela 18, o modelo de RF utilizado para classificar os dados de *churn* da Empresa X deve ser formado por 200 árvores de decisão. Cada AD irá selecionar as *features* pelo método de *bootstrap* com um limite máximo de *features* delimitado pelo logaritmo na base 2 do total de *features*. Como o total de *features* deste modelo é 12, cada AD será formada por, no máximo, 4 *features*.

Além disso, o melhor critério para a escolha das 200 Árvores de Decisão ótimas é baseado no índice de Gini e a profundidade máxima de cada uma das 200 AD's deve ser 13. Por fim, o modelo será alimentado com dados de 17577 *shoppers* distintos. Deste total de dados, 90% será usado para treinar o modelo. Sob estas condições, o modelo usado terá uma precisão de aproximadamente 78%, isto é, quando pedido para classificar um *churn* dentro dos dados de teste, o modelo acerta 78% dos casos. A Figura 13 mostra o código usado para gerar a Tabela 18, bem como os possíveis valores de cada parâmetro.

Figura 13 - Código usado para definir os parâmetros do modelo de RF

```
# Import train_test_split function
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics



## Generate list of features
listFeatures = [elem for elem in dfInput.columns if '_feature' in elem]
X = dfInput[listFeatures]  # Features
y = dfInput['b_isChurn']  # Labels


dfParameters = pd.DataFrame()
for size in [0.1, 0.2, 0.3, 0.4]:
  for estimator in [200, 400, 600, 800]:
    for maxFeatures in ['sqrt', 'log2']:
      for criter in ['gini', 'entropy']:
        for maxDepth in [13, 16, 20]:

          # Split dataset into training set and test set
          X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=size)

          #Create a Gaussian Classifier
          clf=RandomForestClassifier(n_estimators=estimator,max_features = maxFeatures)

          #Train the model using the training sets y_pred=clf.predict(X_test)
          clf.fit(X_train,y_train)

          # Predic the test data
          y_pred=clf.predict(X_test)

          dfParameters = dfParameters.append([[size,
                                               estimator,
                                               maxFeatures,
                                               criter,
                                               maxDepth,
                                               metrics.accuracy_score(y_test, y_pred)]])

```

Fonte: Elaborada pelo autor

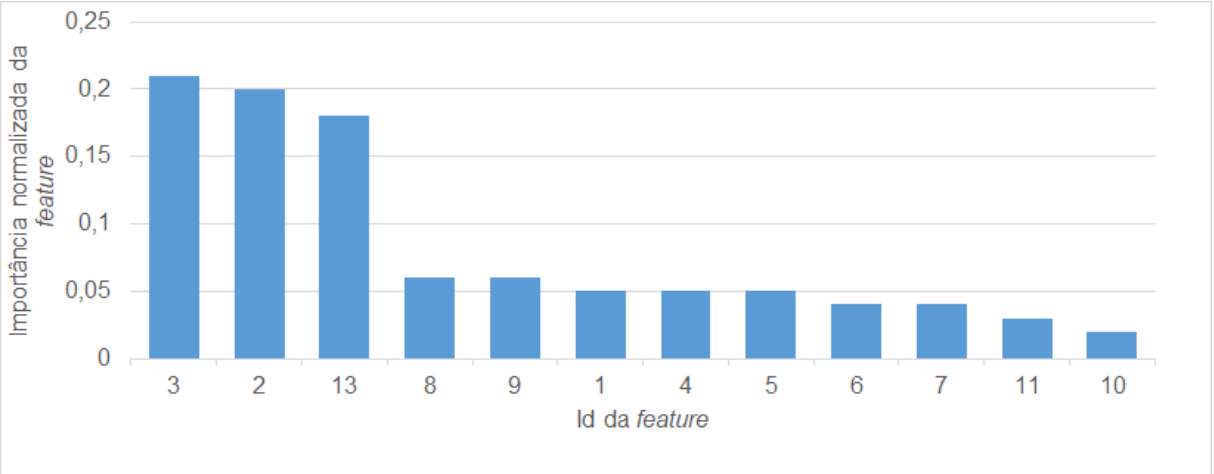
A Tabela 19 e o Gráfico 16 mostram a importância relativa de cada *feature* gerada pelo modelo de RF com 200 AD's e precisão de 78%.

Tabela 19 - Importância das *features*

| Id da *feature* | Nome da *feature* | Importância normalizada da *feature* |
|---|---|---|
| 1 | GMV total | 0,05 |
| 2 | *Status* do primeiro pedido | 0,20 |
| 3 | Meio de pagamento do primeiro pedido | 0,21 |
| 4 | *Status* do último pedido | 0,05 |
| 5 | Meio de pagamento do último pedido | 0,05 |
| 6 | GMV médio por pedido | 0,04 |
| 7 | Estado | 0,04 |
| 8 | Número de acessos à plataforma por dia | 0,06 |
| 9 | Número total de carrinhos abandonados | 0,06 |
| 10 | Número de *sellers* cadastrados | 0,02 |
| 11 | Número de pedidos cancelados | 0,03 |
| 13 | Frequência de compra | 0,18 |

Fonte: Elaborada pelo autor

Gráfico 16 - Importância das *features* em ordem decrescente




Fonte: Elaborado pelo autor

Ao observar o Gráfico 16, é possível identificar que *features* mais importantes para o modelo de RF construído são: o meio de pagamento do primeiro pedido, o *status* do primeiro pedido e a frequência de compra do *shopper*. As outras *features*, quando comparadas com as três *features* mais importantes, possuem uma importância menos expressiva.

Identificar as *features* mais importantes, por si só, pouco diz sobre o caso de *churn* dos clientes na LV. Para entender, por fim, a causa deste *churn*, é preciso interpretar as árvores de decisão construídas neste modelo. Cabe ressaltar que, inclusive, o modelo de *Random Forest* foi escolhido frente a outros modelos pela simplicidade de interpretação dos resultados.

Deste modo, sabendo que as AD's são construídas de maneira a classificar a amostra de dados de maneira ótima, isto é, reduzindo ao máximo sua impureza, o método usado para interpretar a importância das *features* é:

I. Selecionar uma entre as 200 árvores de decisão usadas pelo modelo;
II. Identificar para esta árvore, os *n* nós que representam uma entre as três *features* mais importantes;
III. Para cada um dos *n* nós, identificar quais são os nós que se conectam diretamente com uma folha que representa *churn* e o respectivo valor da *feature* associado a esta folha;
IV. Repetir o procedimento para todas as AD's do modelo

O acesso a cada uma das árvores de decisão do modelo de RF é realizado através do método *sklearn.tree* da mesma biblioteca *sklearn* usada para definir os

parâmetros do modelo. O resultado da interpretação das 200 AD's sob a ótica das três *features* é mostrado na Tabela 20.

Tabela 20 - Interpretação das *features* mais importantes

| **Nome da *feature*** | **Número de árvores de decisão que usaram a *feature*** | **Número de nós da *feature* ligados a folhas de *churn*** | **Valor da categoria da *feature* na separação entre o nó e a folha** |
|---|---|---|---|
| Meio de pagamento do primeiro pedido | 115 | 356 | 1 |
| Meio de pagamento do primeiro pedido | 115 | 24 | 0 |
| *Status* do primeiro pedido | 154 | 15 | 1 |
| *Status* do primeiro pedido | 154 | 391 | 0 |
| Frequência de compra | 172 | 251 | 0 |
| Frequência de compra | 172 | 54 | 1 |
| Frequência de compra | 172 | 85 | 2 |
| Frequência de compra | 172 | 65 | 3 |
| Frequência de compra | 172 | 44 | 4 |
| Frequência de compra | 172 | 15 | 5 |
| Frequência de compra | 172 | 12 | 6 |

Fonte: Elaborada pelo autor

A análise da Tabela 12 evidencia que, entre as três *features* mais importantes para o *churn* dos *shoppers* na LV, o algoritmo de RF gerou 356 nós ligados a folhas

de *churn* para a categoria de valor 1 (compra feita via cartão de crédito) da *feature* meio de pagamento do primeiro pedido, 391 nós associados a folhas de *churn* para a categoria 0 (pedido cancelado) da *feature* de *status* do primeiro pedido e 251 nós ligados a folhas de *churn* para a categoria 0 (primeira compra) da *feature* frequência de compra. Em outras palavras, a interpretação do modelo de RF permite traçar o perfil principal do *shopper* que desiste da LV como sendo **um *shopper* cuja primeira compra na LV é realizada via cartão de crédito e, posteriormente, cancelada**.

Desde a criação do *squad* de análise de dados, a hipótese da Empresa X acerca do *churn* dos usuários estava baseada no número de *sellers* cadastrados no perfil de cada *shopper*. A Empresa X tinha a suposição de que, para cada *shopper*, quanto mais *sellers* cadastrados, mais produtos o dado *shopper* era capaz de enxergar nos *banners* da LV, o que poderia fazer com que a navegação do usuário se tornasse pouco fluída, levando o *shopper* a um *churn*.

O resultado do modelo de RF aponta, entretanto, para outro contexto: *shoppers* que procuram usar a LV como uma alternativa para comprar produtos dos *sellers* usando cartão de crédito como meio de pagamento. Esta condição de pagamento não é oferecida para os canais *offline* dos *sellers* com preços tão atrativos quanto na LV. Assim, o *shopper* atraído para a plataforma, ao tentar realizar sua primeira compra via cartão de crédito e ver este pedido ser cancelado, desiste de utilizar a plataforma ao ter sua compra cancelada.

Este perfil de *churn* foi levado para os líderes da Empresa X em meados de novembro de 2021, período no qual este trabalho foi concluído. Esta evidência fez com que os líderes poupassem recursos que, até então, estavam alocados na melhoria do fluxo de navegação do usuário voltada para o número de *sellers* cadastrados. Estes recursos poupados, majoritariamente compostos por desenvolvedores de *software*, foram realocados em um novo *squad* cujo principal objetivo era entender e melhorar a experiência do usuário da LV em termos de meios de pagamento. Por isso, o novo *squad* criado recebeu o nome de ***squad* de meios de pagamento**.

Desde a criação deste *squad*, no dia 11 de novembro de 2021, até a data de elaboração deste trabalho, o *squad* de meios de pagamentos já identificou um problema essencial no fluxo de informações entre o *shopper* e o *seller* em uma compra via cartão de crédito na LV. Segundo as descobertas do *squad*, quando uma

compra via cartão de crédito é realizada, o *seller* cujos produtos são transacionados precisa receber da LV um arquivo com as informações daquele pedido. Com este arquivo, o *seller*, utilizando seu sistema de informação interno, valida se a compra pode de fato ser concluída com base em um critério:

I. **O *shopper* possui crédito na carteira do dado *seller*:** esta validação não é intermediada pela LV. Aqui, o *seller* verifica dados de inadimplência do *shopper* e decide se valida ou não o pedido via cartão de crédito.

Atualmente, a informação enviada da LV para o *seller* é realizada por uma integração automática existente entre cada um dos *sellers* e a plataforma da LV. As análises iniciais do *squad* de meios de pagamento mostraram que cada *seller* possui requisitos específicos para o recebimento desta informação. Deste modo, a maneira atual com que a integração é realizada pressupõe que cada *seller* deva receber a informação de compras via cartão de crédito no mesmo formato, o que difere da realidade. Assim, atualmente, muitas informações de pedidos realizados via cartão de crédito são perdidas na integração da LV com o *seller*, gerando pedidos com erro.

Deste modo, o *squad* de meios de pagamento já identificou um potencial ponto a ser melhorado a fim de reduzir os erros nos pedidos realizados via cartão de crédito e, atualmente, desdobrou este problema em tarefas que serão realizadas ao longo do primeiro trimestre de 2022.

## 7. CONCLUSÃO

No ano de 2021, especialmente após a pandemia de COVID-19, os *e-commerces* tornaram-se cada vez mais presentes nas vidas das pessoas. Assim, o ser humano passou a visitar cada vez mais os *marketplaces* de produtos ou serviços diversos. Neste contexto, as empresas que controlam estes *marketplaces* têm buscado diversas maneiras de entender os diferentes perfis de seus usuários a fim de melhorar sua experiência nas plataformas virtuais.

Assim, a Empresa X, uma *startup* cujo portfólio de produtos contém, entre outros, um *marketplace* focado em materiais de construção, realizou em novembro de 2020 uma pesquisa qualitativa com uma amostra de 450 usuários da plataforma buscando entender suas dores. A pesquisa foi baseada no método de *Net Promoter Score* e seus resultados indicaram que a maioria dos usuários que responderam à pesquisa não recomendariam a plataforma virtual da Empresa X a um conhecido.

Além deste cenário, a Empresa X notou que, em média, desde julho de 2020 até março de 2021, 24% dos usuários que compravam no *marketplace* em um dado mês não voltavam a comprar no mês seguinte. A Empresa X deu a este evento o nome de desistência ou *churn* e criou, em março de 2021, uma força-tarefa para identificar as causas desta desistência.

Inicialmente, a força-tarefa, da qual o autor deste trabalho fez parte, analisou a métrica atual de desistência em si a qual levava em consideração apenas o fato de um cliente ter comprado na plataforma em um mês e não comprado no mês seguinte sem considerar, entretanto, que dentro da base de clientes pode haver diferentes perfis de compra.

A força-tarefa, também chamada de *squad* de análise de dados, analisou que cada cliente da plataforma possui uma frequência de compra no *marketplace* distinta. Assim, o *squad* criou categorias de frequência de compra baseadas em dias úteis entre compras. Estas categorias englobam desde compras semanais até compras semestrais e são atribuídas a cada um dos clientes da plataforma.

Deste modo, o *squad* propôs, para cada categoria de frequência de compra, um limite máximo de dias úteis que o cliente daquela categoria pode ficar sem comprar na plataforma sem ser considerado um desistente. Assim, a métrica de *churn* passou a avaliar a frequência de compra do usuário na plataforma, refletindo de maneira mais fiel a base de clientes da Empresa X.

Com a nova métrica de *churn* em mãos, a força-tarefa escolheu um entre três métodos de classificação de *Machine Learning* para identificar as causas do *churn*. O método escolhido foi *Random Forest* por conta de sua facilidade em ser interpretado quando comparado com os métodos de Regressão Logística e Redes Neurais Artificiais. Então, foram selecionados dados de aproximadamente 17500 usuários distintos para alimentar o modelo.

Estes dados foram modelados a fim de reduzir a cardinalidade das variáveis contínuas e evitar que o modelo fosse enviesado. Uma vez modeladas, as informações tornaram-se 13 *features* distintas. Os parâmetros do modelo foram escolhidos com base em um processo iterativo que selecionou a combinação de parâmetros que gerasse o modelo com maior precisão. Neste caso, o modelo de Random Forest criado obteve uma precisão de 78%.

O resultado da importância das *features* para o modelo de *churn* apontou como principal causa do *churn* as primeiras compras realizadas na plataforma via cartão de crédito que não foram finalizadas com sucesso. Este resultado fez com que a empresa melhor entendesse o comportamento de seu usuário que, majoritariamente, vê o *marketplace* como uma alternativa para comprar produtos com preço atrativo via cartão de crédito, o que muitas vezes não ocorre nos canais de vendas *offline* dentro dos quais o usuário está acostumado a comprar.

Após o resultado deste trabalho, a Empresa X criou uma nova força-tarefa focada em entender e resolver os problemas associados a meios de pagamento no *marketplace*. Esta força-tarefa, em 15 dias de operação, notou que a maioria dos pedidos cancelados por conta de problemas em meios de pagamento, ocorrem por falha na integração entre a Empresa X e os fornecedores do *marketplace*. Deste modo, a Empresa X pretende priorizar suas tarefas no primeiro trimestre de 2022 a fim de solucionar este problema.

O presente trabalho auxiliou a Empresa X a entender melhor o perfil de seu usuário além de, através dos resultados do modelo de *Machine Learning*, fornecer insumos para que a empresa possa alocar seus recursos de maneira eficiente e focada em reduzir o *churn*. Cabe ressaltar que, antes do resultado deste trabalho, a maior parte dos recursos da Empresa X estava focada em tarefas técnicas de fluxo de navegação do usuário que não davam visibilidade à empresa de problemas associados a meios de pagamento.

Além do conhecimento adquirido pelo autor do presente trabalho em temas de *Machine Learning*, o desenvolvimento deste relatório ressaltou a importância da revisão contínua de métricas a fim de validar se condizem com a realidade dentro do qual uma empresa é inserida. Assim, espera-se que este trabalho possa estimular o uso de *Machine Learning* como ferramenta auxiliar para que se conheça o comportamento e os vieses de usuários de plataformas virtuais.

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

ARTIFICIAL Neural Networks: Biological Motivation. *In*: MITCHELL, Tom M. Machine Learning. [*S. l.*]: McGraw-Hill Science/Engineering/Math, 1997. cap. 4, p. 82. ISBN 0070428077.

ASCARZA, Eva. Retention futility: Targeting high risk customers might be ineffective. Journal of Marketing Research, [*S. l.*], ago. 2017.

BONACCORSO, Giuseppe. Machine Learning Algorithms: A reference guide to popular algorithms for data science and machine learning. Birmingham, UK: Packt Publishing Ltd., 2017. 336 p. ISBN 978-1-78588-962-2.

BREIMAN, Leo. Random Forests. Random Forests, Berkeley, CA 94720, 2001.

CHRISTOFF, Chris. How To Easily Slow Down Customer Churn. *In*: **Forbes**. 26 jun. 2020. Disponível em: https://www.forbes.com/sites/theyec/2020/06/26/how-to-easily-slow-down-customer-churn/?sh=abf2f9d459ae. Acesso em: 17 jul. 2021.

CORPORATE FINANCE INSTITUTE. “Gross Merchandise Value (GMV): The total amount of a company‘s sales over a specified period.” [*S. l.*], ca. 2020. Disponível em: https://corporatefinanceinstitute.com/resources/knowledge/finance/gross-merchandise-value-gmv/. Acesso em: 10 nov. 2021.

DALL'AGNOL, Laisa. Construção é setor que mais perdeu receita em vendas online em 2021. In: **Veja.** [S. l.], 19 nov. 2021. Disponível em: https://veja.abril.com.br/blog/radar/construcao-e-setor-que-mais-perdeu-receita-em-vendas-online-em-2021/. Acesso em: 1 dez. 2021.

DREISEITL, Stephan; OHNO-MACHADO, Lucila. Logistic regression and artificial neural network classification models: a methodology review. Journal of

Biomedical Informatics, 35 (5-6), p. 352-359, 7 fev. 2003. DOI 10.1016/S1532-0464(03)00034-0. Disponível em: www.sciencedirect.com. Acesso em: 20 nov. 2021.

E-COMMERCE BRASIL. E-commerce no Brasil bate recorde e atinge R$ 53 bilhões no 1º semestre, mostra Ebit|Nielsen. *In*: **E-commerce Brasil**. [*S. l.*], 11 ago. 2021. Disponível em: https://www.ecommercebrasil.com.br/noticias/e-commerce-no-brasil-bate-recorde-e-atinge-r-53-bilhoes-ebit-nielsen-webshoppers/. Acesso em: 1 dez. 2021.

FIGUEIRA, Cleonis V. MODELOS DE REGRESSÃO LOGÍSTICA. Orientador: Prof. Dra. Sílvia Regina Costa Lopes. 2006. 149 p. Dissertação (Mestrado em Ciência da Computação) - Instituto de Matemática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006.

GONZALEZ, Leandro de A. Regressão Logística e suas Aplicações. Orientador: Prof. Dr. Ivo José da Cunha Serra. 2018. 46 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Tecnológicas, Universidade Federal do Maranhão, São Luís, 2018.

GOUVEIA, Rosimar. Média Aritmética. *In*: TodaMatéria. [*S. l.*], 2021. Disponível em: https://www.todamateria.com.br/media/. Acesso em: 10 nov. 2021.

GRISAFFE, Douglas B. Questions About The Ultimate Question: Conceptual Considerations In Evaluating Reichheld's Net Promoter Score (NPS). Journal of Consumer Satisfaction, DIssatisfaction and Complaining Behavior. Vol. 20, (2007): 36-53. Disponível em: https://www.proquest.com/openview/0e0fc91a969ae53f2e87f6db8f79c815/1?pq-origsite=gscholar&cbl=46531. Acesso em: 20/11/2021.

HATA, Itamar. Classificadores de Alta Interpretabilidade e de Alta Precisão. Orientador: Prof. Adriano Alonso Veloso. 2013. 82 p. Dissertação (Mestrado em

Ciência da Computação) - Instituto de Ciências Exatas, Universidade Federal de Minas Gerais, Belo Horizonte, 2013.

HATA, Itamar *et al*. Learning Accurate and Interpretable Classifiers Using Optimal Multi-Criteria Rules. Journal of International and Data Management, [*S. l.*], ano 2013, v. 4, n. 3, p. 204-219, out. 2013.

INTRODUCTION: Perspectives and Issues in Machine Learning. *In*: MITCHELL, Tom M. Machine Learning. [*S. l.*]: McGraw-Hill Science/Engineering/Math, 1997. ISBN 0070428077.

INTRODUÇÃO: Âmbito e Objectivo da Estatística. In: SANTOS, Carla. Estatística Descritiva: Manual de Auto-Aprendizagem. 3. ed. Lisboa: Edições Sílabo, 2018. cap. 1, p. 15. ISBN 978-972-618-968-8.

JAMALALDIN, Seyed *et al*. Application of artificial neural networks to predict compressive strength of high strength concrete. **International Journal of the Physical Sciences**, [*S. l.*], ano 2011, v. 6, n. 5, p. 975-981, 4 mar. 2011. DOI 10.5897/IJPS11.023. Disponível em: http://www.academicjournals.org/IJPS. Acesso em: 18 jul. 2021.

KATELARIS L., THEMISTOCLEOUS M. (2017) Predicting Customer Churn: Customer Behavior Forecasting for Subscription-Based Organizations. In: **Themistocleous M., Morabito V. (eds) Information Systems. EMCIS 2017. Lecture Notes in Business Information Processing, vol 299. Springer, Cham**. https://doi.org/10.1007/978-3-319-65930-5_11.

LAUDON, Kenneth C.; LAUDON, Jane P. Sistemas de Informação Gerenciais. 7. ed. São Paulo: Pearson Prentice Hall, 2007. 452 p. ISBN 978-85-7605-089-6.

MAIMON, Oded; ROKACH, Lior. Data Mining and Knowledge Discovery Handbook. 2. ed. New York: Springer, 2010. 1285 p. ISBN 978-0-387-09822-7.

MEDIDAS-RESUMO: Medidas de Posição. *In*: MORETTIN, Pedro A.; BUSSAB, Wilton de O. Estatística Básica. 5. ed. São Paulo: Saraiva, 2004. cap. 3, p. 35-36. ISBN 85-02-03497-9.

MENDONÇA, Herbert Garcia. E-commerce. IPTEC - Revista Inovação, Projetos e Tecnologias, [*s. l.*], ano 2016, v. 4, n. 2, p. 240-251, dez 2016.

MORETTIN, Pedro A.; BUSSAB, Wilton de O. Estatística Básica. 5. ed. São Paulo: Editora Saraiva, 2004. 526 p. ISBN 85-02-03497-9.

RESENDE, Hellen D.; COIMBRA, Lucas A.; DE PAULA, Murilo R. Revisão Bibliográfica: Redes Neurais Artificiais. *In*: RESENDE, Hellen D.; COIMBRA, Lucas A.; DE PAULA, Murilo R. REDES NEURAIS ARTIFICIAIS PARA A ESTIMATIVA DA RESISTÊNCIA E CONSISTÊNCIA DE CONCRETOS. 2018. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2018. p. 47-53.

SHARMA, Anuj; PANIGRAHI, Prabin. A Neural Network based Approach for Predicting Customer Churn in Cellular Network Services. **International Journal of Computer Applications** (0975-8887), [*S. l.*], ano 2011, v. 27, n. 11, p. 26-31, ago. 2011.

SILVA, Marcos Noé Pedro da. "Média ponderada"; *Brasil Escola*. Disponível em: https://brasilescola.uol.com.br/matematica/media-ponderada.htm. Acesso em 10 de novembro de 2021.

STROBL, Carolin *et al*. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics, München, Germany, 8:25, p. 1-21, 25 jan. 2007. Disponível em: http://www.biomedcentral.com/1471-2105/8/25. Acesso em: 20 nov. 2021.