Clientes Pagantes vs Trial: Explorando a Identificação em um Ambiente SaaS

Igor Lemos Vicente
Universidade Federal da Fronteira Sul
Campus Chapecó
Chapecó – SC – Brasil
igor94@gmail.com

Denio Duarte
Universidade Federal da Fronteira Sul
Campus Chapecó
Chapecó – SC – Brasil
duarte@uffs.edu.br

Guilherme Dal Bianco
Universidade Federal da Fronteira Sul
Campus Chapecó
Chapecó – SC – Brasil
guilherme.dalbianco@uffs.edu.br

ABSTRACT

Software as a Service (SaaS) is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted, generally, in the web. In this software delivery model, users have a period for freely testing the software before buying it, called trial. This work aims to extract features from a specific SaaS product (named *Belasis*) to build a classification model to identify whether or not a trial customer becomes a payer one. To accomplish that we use three classification algorithms: *Support Vector Machine*, *K-Nearest Neighbours* and *Random Forest*. We conduct some experiments using some existing and new features from *Belasis*, and the results are promissing. Based on F_1*-score* metric, *Random Forest* gets the best results.

Keywords

Classification; Free Trial; Software-as-a-Service; Supervised Learning

CCS Concepts

•Computing methodologies → Machine learning approaches;

1. INTRODUÇÃO

Atualmente, o acesso à Internet é cada vez mais simples e com custo mais acessível. O uso da Internet para compartilhar, transmitir e acessar dados faz parte do dia-a-dia das empresas e usuários domésticos. A expansão da Internet trouxe uma nova classe de serviço computacional: a computação nas nuvens (*cloud computing*). Dentre os vários usos da computação nas nuvens, pode-se citar Software como um Serviço (em inglês, *Software-as-a-Service* - SaaS). Este modelo de entrega de sistemas computacional revolucionou as empresas desenvolvedoras de *software* bem como os usuários de sistemas de informação [16].

O modelo de entrega de um sistema no formato SaaS possibilita o uso sem precisar de configuração e manutenção de infraestrutura e máquinas para hospedagem do servidor da aplicação pelo usuário [14]. Por se encontrar acessível na Internet, o usuário necessita apenas de uma conexão e um navegador *web* para utilizar o serviço desejado. Outra característica dessa forma de entrega de produto é o pagamento: o usuário aluga o serviço através de pagamentos mensais. Essa forma de contrato permite que a empresa tenha uma renda mensal com o serviço, além de permitir que o usuário cancele o uso de forma mais flexível.

Muitas empresas desenvolvedoras, então, migraram suas soluções para o SaaS, o que cria um aumento na oferta. Esse aumento fez com que empresas criassem estratégias para atrair clientes para conhecer e utilizar seus produtos. Uma das estratégias mais utilizadas é a chamada uso *Trial*, ou seja, o produto é oferecido por um período determinado com todas as funcionalidades para o usuário identificar se a solução vem ao encontro de suas necessidades. Com essa estratégia, a empresa pode identificar os usuários *Trials* e aplicar alguma estratégia de *marketing* para torná-los efetivos [6].

Nesse contexto, este trabalho tem como objetivo criar modelos de aprendizado de máquina que identifiquem quais clientes *Trials* estão mais propensos para se tornarem efetivos. Identificando tais clientes, a empresa desenvolvedora pode criar estratégias de *marketing* mais efetivas para atrair usuários para o seu produto.

O estudo de caso utiliza um ambiente SaaS, chamado de *Belasis*[1], que permite a gestão de empresas que prestam serviços focados principalmente em salões de beleza, barbearias, esmalterias, clínicas e spas. O *Belasis* permite a criação de uma conta com acesso ao sistema para uso gratuito por um período de 7 dias com todas as funcionalidades. Passado o período *Trial*, o usuário, para continuar o uso, deve assinar um dos planos disponibilizados pela aplicação. Caso não o faça, o sistema fica bloqueado, impedindo-o de realizar qualquer outra ação que não seja a efetivação da assinatura. Os planos ofertados variam de preço conforme as funcionalidades oferecidas.

Para a empresa desenvolvedora do produto, é importante que um usuário em período de teste efetive a assinatura de algum dos planos, tornando-se, assim, um cliente com assinatura ativa. Algumas medidas são tomadas para este fim, uma delas é a venda ativa por telefone. Esse contato

[1] `www.belasis.com.br`

demanda um uso substancial de recursos humanos, tendo em vista que muitos clientes *Trials* não têm interesse na assinatura.

Ao prever quais usuários são mais propensos à efetivação da assinatura, a empresa passa a usar menos recursos humanos para contatos que se mostram infrutíferos e passa a ter mais disponibilidade de esforços de seus colaboradores para a execução de outras tarefas. Tal distinção, porém, mostra-se uma tarefa de execução complexa pela falta de informações sobre o cliente em potencial. Uma abordagem apoiada nos dados gerados pelo usuário durante o uso do sistema no período de teste poderia ser usada para análise de comportamento, assim, modelos baseados em aprendizado de máquina podem ser estratégicos para a empresa.

Para conduzir este trabalho de identificação de clientes mais propensos a se tornarem efetivos, três algoritmos de aprendizado de máquina foram testados durante a fase de execução do projeto: *K-Nearest Neighbours*, *Random Forest* e *Support Vector Machine*. Os três algoritmos são validados na etapa do projeto com base nas métricas precisão, revocação e F_1-*score*.

A aplicação dos algoritmos foi realizada em um conjunto de atributos extraídos da base de dados do produto *Belasis*. Os atributos extraídos representam o comportamento do usuário no sistema durante o período *Trial* de sete dias. Os experimentos mostram que os modelos podem ser úteis para a empresa criar estratégias de marketing mais efetivas para fidelizar clientes *Trials*. Dentre os algoritmos utilizados, a *Random Forest* foi que obteve o melhor resultado utilizando a métrica F_1-*score*.

O restante do artigo está organizado da seguinte forma: a próxima seção apresenta algumas definições importantes para o entendimento deste trabalho: a empresa alvo, aprendizado de máquina e métricas. Em seguida, alguns trabalhos relacionados são apresentados. A Seção 4 apresenta o projeto e os resultados dos experimentos. Finalmente, a Seção 5 apresenta conclusão e os possíveis trabalhos futuros. Os Apêndices A e B apresentam os atributos utilizados para a construção dos modelos.

2. REFERENCIAL TEÓRICO

Esta seção apresenta dois conceitos importantes para o entendimento da proposta deste trabalho: o ambiente SaaS utilizado como estudo de caso e conceitos de aprendizado de máquina. Ambos são apresentados brevemente a seguir.

2.1 Belasis

Criada em 2015, a *Belasis* é uma empresa que tem em seu portfólio um sistema para gestão de empresas que prestam serviços. Embora atenda algumas regras de negócio gerais, a empresa tem como foco os segmentos de salões de beleza, barbearias, esmalterias, clínicas e spas. A entrega da solução da empresa é feita por um SaaS homônimo (*i.e.*, o produto é chamado *Belasis*) em que os usuários que desejam ter acesso às funcionalidades devem ter apenas conexão com a internet e um navegador *web*.

Ao se cadastrar no sistema, o usuário tem sete dias de teste grátis. Após esse tempo, o usuário deve contratar algum plano e efetivar a assinatura para continuar com o uso. Os planos da empresa são oferecidos de acordo com as funcionalidades que estão presentes neles.

O objetivo do trabalho foi definido com base nos dados de uso dos usuários do sistema *Belasis*. A empresa contribuiu para o presente trabalho disponibilizando a base de dados que é usada para a construção dos conjuntos de dados. Os dados de um usuário são baseados no uso das funcionalidades que o sistema oferece. O uso, por sua vez, é representado pelos registros de modelos criados no banco de dados, enquanto o usuário usa as funcionalidades.

As funcionalidades do sistema são criadas para atender principalmente três necessidades dos salões de beleza. A primeira é a da agenda, que serve para controlar os horários de serviços marcados. Um agendamento pode ser criado por uma pessoa que tem acesso ao sistema. Além dos usuários do sistema, os seus clientes também podem agendar por meio de aplicativos para celulares. A segunda principal funcionalidade é a comanda, nela se registra o consumo e serviços prestados ao cliente durante sua permanência no estabelecimento. Ao final do atendimento e no pagamento das pendências, o usuário do sistema fatura a comanda (isto é, grava o valor do pagamento, as formas de pagamento, descontos, entre outros). A terceira principal funcionalidade é, na verdade, um conjunto de outras funcionalidades, agrupadas no módulo chamado de financeiro. Neste módulo, é possível gravar despesas e receitas (inclusive aquelas de origem na comanda), realizar conferência de caixa, gerar relatórios, entre outras funcionalidades.

2.2 Aprendizado de Máquina e Métricas de Avaliação

O aprendizado de máquina é uma sub-área da inteligência artificial cujo o objetivo é criar modelos de predição através da aplicação de algoritmos sobre um conjunto de dados previamente conhecido (conjunto de treinamento). Os modelos são criados através da generalização do conjunto de treinamento. Esta generalização permite predizer novos valores utilizando dados não vistos durante o treinamento [9].

As duas principais classes de algoritmos de aprendizado de máquina são supervisionados e não supervisionados. Os algoritmos supervisionados são aqueles que utilizam conjuntos de treinamento que possuem rótulos, ou seja, os rótulos guiam o algoritmo na generalização do modelo. Já os não supervisionados utilizam conjunto de dados sem rótulos e a tarefa é encontrar grupos nos conjuntos como, por exemplo, *clusters*.

Para a criação dos modelos da proposta deste trabalho, o conjunto de dados possui um rótulo indicando se o cliente efetivou ou não a assinatura, assim, este trabalho se encaixa na classe dos algoritmos supervisionados. Sendo o rótulo discreto (*i.e.*, dois valores possíveis), o tipo de aprendizado supervisionado é classificação. Conforme apresentado anteriormente, este trabalho utiliza três classificadores para criar o modelo de predição [10, 1]: *Support Vector Machine*, *K-Nearest Neighbours* and *Random Forest*.

A criação de um modelo de predição envolve vários passos e um dos mais importantes é a avaliação do resultado comparando as classes preditas com as classes verdadeiras. Esta avaliação é feita através do uso de métricas. As métricas se baseiam nos resultados de predição de uma das classes que pode ser categorizado de quatro formas, conforme a matriz de confusão apresentada na Tabela 1. A categoria *Verdadeiro Positivo* (VP) é a situação em que a classe foi corretamente predita como positiva. A categoria *Verdadeiro Negativo* (VN), por sua vez, é a situação em que a classe foi corretamente predita como negativa. Ao classificar um exemplo que é negativo como positivo, tem-se a categoria de

Falso Positivo (FP). A categoria de *Falso Negativo* (FN) se dá quando classifica-se um exemplo positivo como negativo.

Table 1: Matriz de confusão

	Positivo real	Negativo real
Positivo predito	TP	FP
Negativo predito	FN	TN

As principais métricas utilizadas para a validação de modelos de classificação são baseadas na contagem de classes previstas [7] e categorizadas na matriz de confusão. A mais tradicional delas, a acurácia, pode ser definida como todos os acertos sobre todos os exemplos e é dada pela Equação 1. A acurácia é a métrica recomendada quando o número de classes do conjunto de dados é balanceado [7]. Se um dado conjunto de dados, por exemplo, possui 96% de exemplos positivos e um modelo retorna apenas exemplos positivos, este modelo terá uma acurácia de 96%. Porém, este mesmo modelo, apesar de uma ótima acurácia, não é capaz de identificar um exemplo negativo, portanto, é um modelo pobre.

Quando o conjunto de dados é desbalanceado outras métricas são recomendadas, como a precisão (Equação 2), a revocação (Equação 3) e a F_1-*score* (Equação 4), sendo esta última a média harmônica entre a precisão e a revocação.

$$A = \frac{VP + VN}{VP + VN + FP + FN} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{4}$$

Intuitivamente, a precisão é usada quando é necessária assertividade ao encontrar os exemplos positivos. A revocação, por sua vez, é usada quando é necessária assertividade em não deixar de identificar nenhum exemplo positivo. Os modelos construídos neste trabalho serão avaliados utilizando essas três métricas, sendo que a F_1-*score* é a escolhida para indicar o melhor modelo pois representa a média harmônica entre a precisão e a revocação.

3. TRABALHOS RELACIONADOS

A maioria dos trabalhos que apresentam abordagens para implementar alguma tarefa de previsão em SaaS focam na retenção de clientes pagantes [15, 13, 12, 11]. Apenas a abordagem apresentada em [17] lida com clientes que utilizam serviços de armazenamento na nuvem de forma gratuita e se tornarão pagantes, sendo o mais similar a proposta apresentada aqui.

Os trabalhos de [12] e [11] utilizam algoritmos de aprendizado de máquina para identificar clientes *churns* (*desertores* em português), *i.e.*, clientes pagantes que por alguma razão encerram o uso do produto. O primeiro trabalho aplica quatro algoritmos de aprendizado de máquina para identificar possíveis candidatos a *churn*: redes neurais recorrentes (algoritmo *Long Short-term Memory*– LSTM), redes neurais convolucionais (*Convolutional Neural Network* – para CNN), *Support Vector Machine* (SVM) e *Random Forest*. Já o segundo trabalho, além de utilizar também SVM, aplicou árvores de decisão para a criação de modelo de previsão.

O trabalho de [12] utilizou um conjunto de dados disponibilizado por uma empresa finlandesa *Smartly.io*, que oferece soluções em *marketing* para plataformas como *Facebook* ou *Pinterest*. Após experimentos, a média de acurácia obtida com os algoritmos foi de em torno 75% mas com um F_1-*score* abaixo de 50%. Isso ocorre pois as classes do conjunto de dados não são balanceadas, ou seja, existem mais exemplos de clientes não-desertores do que de desertores.

O trabalho de [11] também trata o problema de prever possíveis clientes que se tornarão *churn*. A empresa do estudo, segundo o autor, é uma provedora de *software* hospedado na nuvem (ou seja, um SaaS). Não é explicitado o tipo de serviço que a empresa oferece, apenas é citado três produtos que são definidos com base em seus preços: o produto de custo baixo (*low-price*, abreviado para LP); custo médio (*middle-price*, abreviado para MD); e custo alto (*high-price*, abreviado para HP).

O modelo de predição foi criado utilizando dois métodos de aprendizado de máquina: árvore de decisão (implementação do algoritmo C4.5) e o *Support Vector Machine*. Ambos os algoritmos foram usados para classificar os exemplos do conjunto de dados usados. Além disso, o SVM tem uma funcionalidade que é possível calcular qual a probabilidade de um exemplo pertencer a uma determinada classe. Essa funcionalidade foi explorada pelo autor para a criação de um distribuição de frequência, que é usada no cálculo de retorno financeiro esperado para um cliente. Como resultado final, o SVM foi o melhor modelo para prever clientes que optam por serviço LP, o C4.5 foi o melhor para MD e ambos empataram estatisticamente para HP.

No trabalho de [15] é apresentado dois novos algoritmos de mineração de dados para idetificar *churns*: AntMiner+ (que usa Ant Colony Optimization) e outro chamado Active Learning Based Approach (ALBA). Ambos são utilizados para extrair regras de associação. O conjunto utilizado foi de uma operadora de telecomunicação sem fio disponível no repositório *KDD Library*. Os resultados obtidos relativos à media de acertos foi superior a 90% sendo que o número de regras de classificação criadas ficaram entre 10 e 70, mas a maioria ficou abaixo de 30.

Em [13] é proposta uma abordagem para identificar clientes que não se tornarão pagantes utilizando dois passos: clusterização e árvores de decisão. O conjunto de dados é de uma operadora de telefonia pré-paga. Foram criados quatro *clusters* baseados nas distância em dias das ligações. Em seguida, foram criados mais 8 atributos baseados nos *clusters* e árvores de decisão foram aplicadas em cada *cluster*. Os experimentos levaram a dois resultados: o percentual de identificação de *churns* em cada *cluster* (77.8%, 66.7%, 30% e 45.5% , respectivamente) e a identificação de atributos que contribuem mais para a identificação de *churns*: frequência de uso, minutos de uso, número de ligações realizadas e número de ligações recebidas.

Finalmente, o trabalho mais próximo da proposta aqui apresentada usa dados de um serviço na nuvem para armazenamento de dados [17]. A empresa oferece o serviço gratuitamente limitando a quantidade de dados armazenada e de forma paga. Os autores enviaram um questionário para usuários do *Google Drive* e obtiveram 181 respostas de usuários não pagantes. O objetivo dos autores foi encontrar os valores que levariam os clientes com licença gratuita a ser

tornarem pagantes, auxiliando fornecedores destes serviços serem mais assertivos nas promoções de licenças pagantes.

Percebe-se, pelos trabalhos aqui citados, que a maioria cria modelos para identificar clientes que após efetivarem suas assinaturas abandonam o produto. Apenas o trabalho de [15] propõe a identificação da razão do abandono utilizando regras de associação que na maioria das vezes se tornam complexas pelo número de testes (condições) realizadas para se encontrar a classe. Os outros trabalhos [13, 12, 11] aplicam algoritmos de aprendizado de máquina que através do uso de métricas padrão para classificação identificam a probabilidade de abandono dos clientes. Apesar da similaridade de previsão, a proposta aqui apresentada difere apenas no momento da identificação: enquanto os trabalhos relacionados tratam de clientes assinantes que abandonam um produto assinado, esta proposta trata de clientes que estão testando um sistema e podem ou não, após o fim do período de teste, abandonar. A maior diferença entre as duas abordagens é em relação ao conjunto de dados, enquanto no primeiro caso ele já está consolidado, ou seja, possui informações reais do uso do sistema, esta proposta trabalha com um conjunto de dados em que as transações presentes são menos consolidadas. Isso ocorre por alguns motivos: o cliente ainda não possui total domínio do produto, a insegurança do cliente em inserir seus dados confidenciais e a adaptação da rotina do cliente em relação aos processos oferecidos pelo produto.

4. EXPERIMENTOS

O primeiro passo do projeto dos experimentos foi a definição do período de tempo em que os dados dos clientes estariam contidos. Isso porque as funcionalidades do sistema, tal como o modelo de negócio, oferta e *marketing* que a empresa utiliza só tiveram uma certa estabilidade em relação às alterações recentemente. O período escolhido, então, foi de um ano e seis meses, contando a partir do início de 2018.

Um usuário, ao entrar no sistema, passa a ter 7 dias para testá-lo. Cada um desses dias apresenta uma nova oportunidade para a predição de assinatura, por isso foram criados 7 conjuntos de dados. Cada conjunto de dados D, tal que $1 \leqslant D \leqslant 7$, representa o D-ésimo dia de teste dos usuários.

A construção dos conjuntos de dados teve início na obtenção da base de dados de clientes. Os dados foram inseridos numa instância do banco de dados *MySQL* hospedada na máquina que foi utilizada para a execução dos experimentos. A partir dessa base de dados, foram projetados os atributos que retornaram valores que descrevem o uso do sistema pelos clientes.

Os resultados das projeções foram agrupados por usuário e por dia, constituindo os conjuntos de dados para os experimentos. Esse agrupamento é possível pois a data de criação de cada registro encontra-se gravada na base de dados. Disponibilizadas, então, a data de criação de conta e a data de criação de cada registro é possível saber quais são aqueles registros que foram criados no n-ésimo dia da conta do usuário, tal que $1 \leq n \leq 7$. Esse agrupamento acontece pois há a relativização dos dias de teste de um usuário, isto é, o dia 1 de um usuário A pode não ser no mesmo dia 1 de um usuário B. Para a execução do trabalho, só importa quantas comandas (quantidade de atividades realizadas pelo usuário) foram criadas no primeiro dia de uso da conta de um usuário, por exemplo, independente da data deste dia.

Table 2: Características dos conjuntos de dados

Nro de atributos originais	36
Nro de atributos gerados	87
Nro total de atributos	123
Nro total de exemplos	3.373
Nro de exemplos positivos	275 (8,15%)
Nro de exemplos negativos	3.098 (91,85%)

A partir dos atributos extraídos do conjunto de dados, outros 87 novos foram criados: 28 correspondem às médias de valores de alguns atributos correspondendo à movimentação dos dias anteriores, 28 são medianas criadas no mesmo formato das médias, 10 são de razões de registros ou valores sobre o número total de clientes, 3 são de razões de registros ou valores sobre o número total de comandas e 18 são de razões de registros ou valores sobre o número total de profissionais. Os Apêndice A e B apresentam a relação de atributos originais e gerados no conjunto de dados, respectivamente. A Tabela 2 apresenta a descrição do conjunto de dados utilizado. Perceba que é um conjunto em que a distribuição das classes positivas e negativas é desbalanceada.

4.1 Descrição dos atributos

A Tabela 3 apresenta um extrato de valores de alguns atributos no conjunto de dados utilizado. A primeira coluna da tabela apresenta o nome dos atributos. Da segunda coluna em diante, o D_n, representando o n-ésimo dia de conta.

Na Tabela 3, os atributos originais são apenas *comandas* (quantidade de atividades realizadas pelo usuário) e *venda_total* (pagamentos recebidos no período). Os outros atributos correspondem àqueles gerados para auxiliar na criação do modelo. Observe que no primeiro dia, os atributos que correspondem à media e à mediana estão zerados pois não existem registros anteriores para os cálculos. A alternativa de simplesmente não usar os atributos de média e mediana no conjunto do primeiro dia para remover a redundância foi descartada para tornar mais simples a execução dos algoritmos de aprendizado de máquinas utilizados.

Os atributos *comandas_d_por_cliente_t* e *venda_total_d_por_cliente_t* são os atributos de razão sobre o número total de clientes. No primeiro atributo é possível observar a letra d depois de comandas e a letra t depois de cliente. Essas letras sinalizam que o cálculo da razão está sendo feito com as comandas criadas no dia (d) sobre os todos os clientes registrados (t), conforme na equação abaixo:

$$comandas\_d\_por\_cliente\_t = \frac{\text{Quantidade comandas dia}}{\text{Quantidade total clientes}}$$

A hipótese para a criação dessas razões é que elas descrevem o relacionamento entre as funcionalidades cadastradas no sistema. Isto ajudará o modelo a diferenciar um usuário que cria vários registros de um objeto apenas para testar o sistema (e geralmente cria várias comandas para um mesmo cliente, por exemplo) de outro que cria vários registros por já estar usando o sistema (várias comandas para vários clientes). Para os outros atributos que simbolizam razões a lógica é a mesma, com a exceção do atributo *venda_total_d_por_comanda_d*, que tem como divisor um valor que foi registrado apenas no dia (por isso o d depois de comanda).

Table 3: Exemplos de alguns atributos de valores originais e gerados para um usuário

atributo	D_1	D_2	D_3	D_4	D_5	D_6	D_7
comandas	5	34	0	26	49	47	0
venda_total	422,0	2995,0	0,0	2748,0	4065,0	3664,0	0,0
comandas_media	0	19,5	13,0	16,25	22,8	26,83	23,0
venda_total_media	0	1708,5	1139,0	1541,25	2046,0	2315,67	1984,86
comandas_mediana	0	19,5	5	15,5	26	30,0	26
venda_total_mediana	0	1708,5	422,0	1585,0	2748,0	2871,5	2748,0
comandas_d_por_cliente_t	0,5	0,34	0,0	0,17	0,29	0,25	0,0
venda_total_d_por_cliente_t	42,2	29,95	0,0	18,44	23,77	19,7	0,0
venda_total_d_por_comanda_d	84,4	88,09	0	105,69	82,96	77,96	0
comandas_d_por_profissional_t	0,83	5,67	0,0	4,33	8,17	7,83	0,0
venda_total_d_por_profissional_t	70,33	499,17	0,0	458,0	677,5	610,67	0,0

4.2 Criação e validação dos modelos

Após a criação do conjuntos de dados, utilizou-se o pacote *SelectKBest* da biblioteca *scikit-learn* para atribuir notas de importância a cada um dos atributos e ordená-los dos mais informativos para os menos informativos. O pacote citado utilizou a função χ^2 (Qui-quadrado) [5] para calcular a importância dos atributos. A Figura 1 mostra os 10 melhores atributos de cada conjunto de dados. Na figura, o eixo x representa os dias, ou seja, cada conjunto de dados. O eixo y, por sua vez, representa a posição do atributo no ordenamento, que segue a ordem decrescente de informatividade. Cada quadrado no gráfico mostra a posição do atributo que está associada a uma cor. As linhas pontilhadas mostram a movimentação dos atributos no ordenamento. Com base nessa figura, pode-se perceber os seguintes comportamentos dos atributos quanto à importância no modelo de predição:

- Os atributos *movimentacao_saida* e *movimentacao_entrada* aparecem no top-10 de todos os conjuntos de dados. Ou seja, tais atributos representam as movimentações principais do sistema pelo usuário que é um indicador de quanto o sistema está sendo utilizado;
- *movimentacao_saida_media* aparece no top-3 dos conjuntos de dados a partir de segundo dia. Isto devido ao primeiro dia não possuir estatísticas de dias anteriores;
- Os atributos *clientes_d_por_profissional_t*, *servicos_produtos*, *venda_total_d_por_cliente_t*, *clientes* e *venda_total_d_por_comanda_d* se apresentam no topo apenas no primeiro dia. Tais atributos devem ser importantes para os dias posteriores porém os atributos criados a partir de estatísticas de outros atributos se tornam mais importante a partir de segundo dia; e
- A maioria dos atributos top-10 que aparecem a partir de segundo dia utilizam estatísticas dos atributos originais, ou seja, os novos atributos gerados têm um papel importante na construção do modelo.

As constatações acima não seriam possíveis analisando atributo por atributo manualmente mas após serem gerados fica mais claro a razão da importância dos mesmos.

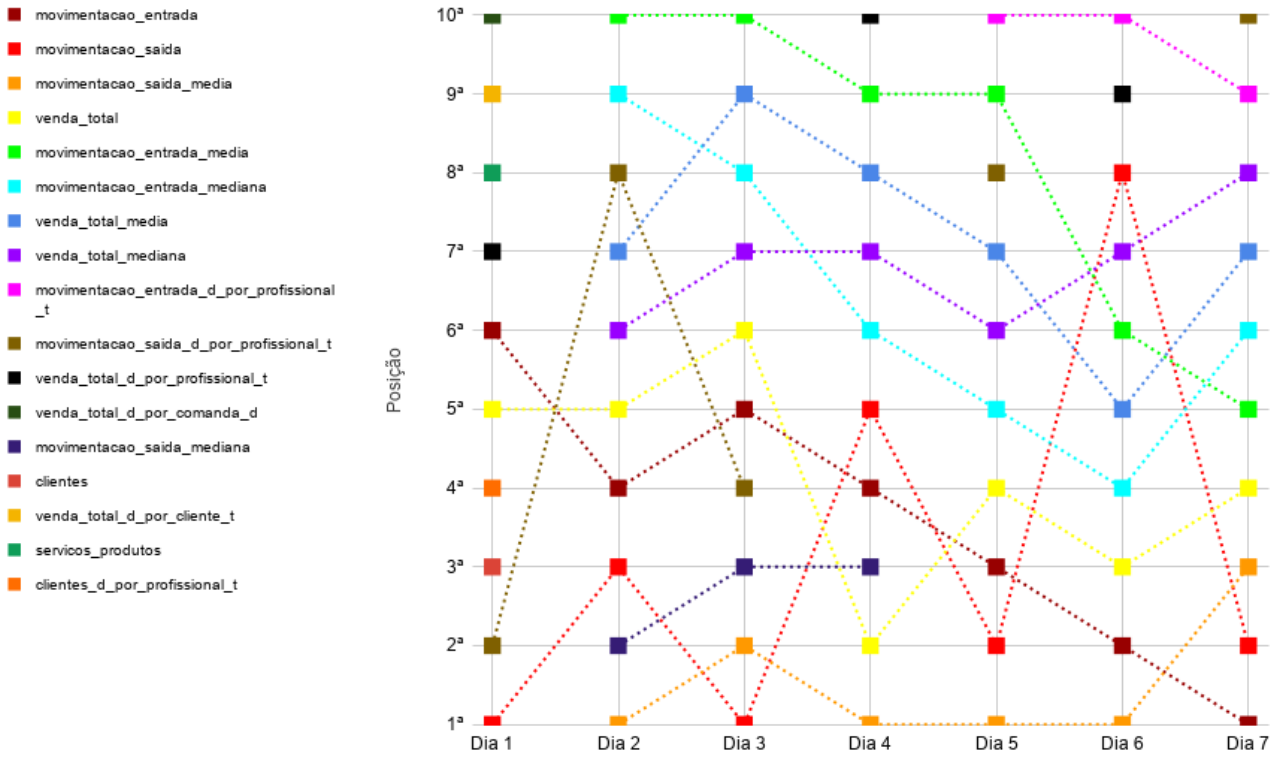
Com os melhores atributos ordenados do mais informativo para o menos informativo, foram executados os algoritmos classificadores para identificar quais as melhores combinações de atributos para cada conjunto de dados considerando a métrica F_1-*score*.

Levando em conta o contexto atual do domínio da aplicação, a métrica que mais condiz com as regras de negócio é a F_1-*score*. Isso porque uma boa precisão significa menos tempo desperdiçado contatando clientes que não virarão pagantes e uma boa revocação significa mais clientes que se tornarão pagantes sendo atendidos. No inverso, uma baixa precisão significa mais tempo desperdiçado contatando clientes que não se tornarão pagantes e uma baixa revocação significa menos clientes que podem se tornar pagantes terão um atendimento personalizado. Por bons resultados em ambas as métricas beneficiarem, de certa forma, igualmente a empresa, a métrica da média harmônica foi escolhida para ser usada na criação e validação dos modelos.

Definida a métrica de avaliação dos algoritmos, um novo experimento foi executado para encontrar qual o algoritmo de classificação que obteve a melhor avaliação e qual o melhor número de atributos para gerar o modelo. Para o primeiro caso, conforme afirmado anteriormente, foi utilizada a F_1-*score*, para o segundo caso foi adotada a seguinte abordagem: (*i*) os atributos são classificados por importância do mais importante (primeiro) ao menos importante (123º), (*ii*) os modelos são gerados com o primeiro atributo e avaliados, com os primeiro e segundo atributos e avaliados e assim por diante e (*iii*) os modelos gerados para cada conjunto de dados são classificados baseados nas avaliações. O Algoritmo 1 apresenta os passos para encontrar o melhor classificador e os melhores atributos para a construção do modelo. O algoritmo possui três laços simples: o mais externo que seleciona os sete conjunto de dados que representam os dias, o segundo entre as Linhas 3 e 11 para combinar os atributos para construir o modelo e o laço mais interno, entre as Linhas 6 e 10, que cria os modelos para cada algoritmo considerado utilizando 60% do conjunto de dados para treino e 40% para teste. Já a Figura 2 apresenta pictoricamente os passos apresentados no Algoritmo 1, sendo que o último passo é a escolha dos melhores modelos×atributos por dia.

Na comparação final dos resultados, o *Random Forest* teve o melhor desempenho em relação aos outros dois algoritmos no experimento. A média da diferença do *Random Forest* para o *K-Nearest Neighbours* e para o *Support Vector*

Figure 1: 10 melhores atributos por dia



Algorithm 1: Algoritmo para criação de modelos para validação dos algoritmos.

```
1  para d de 1 até 7 faça
2      dataset ← carrega conjunto de dados do dia d;
3      para k de 1 até 123 faça
4          X ← dataset com k melhores features;
5          treino, teste ← (X * 0.6, X * 0.4);
6          para cada m em [RF, SVC, KNN] faça
7              modelo ← m.train(treino);
8              resultado ← modelo.fit(teste);
9              salva resultado;
10         fim
11     fim
12 fim
```

Machine foi de 27,91% e 14,64%, respectivamente. A partir disso, então, o *Random Forest* foi escolhido como o classificador final para os experimentos. A Figura 3 apresenta os melhores resultados por dia após a execução do Algoritmo 1. Perceba que, conforme descrito, *Random Forest* supera em termos de F_1-*score* todos os outros algoritmos. Foi utilizado o método de segmentação por estratificação para a criação do conjuntos de treino e teste. Este método garante a confiança no resultado final das métricas pois os conjuntos são separados em lotes com o mesmo balanceamento das classes, evitando o viés e a variação nos dados (bias e variance, respectivamente). Pode-se perceber, também, pela diferença nos resultados (*e.g.*, 14,64 foi a menor diferença) que a afirmação *Random Forest* teve o melhor desempenho é correta.

Finalmente, a Tabela 4 apresenta as melhores quantidade de atributos para cada dia da semana (conjunto de dados). Perceba que, para o sexto dia, a melhor combinação é de 114 atributos, já para o terceiro dia é de apenas 38. Esses números indicam a quantidade de melhores atributos por ordem de importância que foram utilizadas para alcançar os resultados obtidos para o algoritmo *Random Forest*. Estes valores também serão utilizados no experimento final para a construção de um *Random Forest* com hiperparâmetros otimizados.

4.3 Experimento final

A etapa anterior dos experimentos teve como objetivos: (*i*) identificar entre os três algoritmos aquele que teve o melhor desempenho na métrica F_1-*score* (*Random Forest*), (*ii*) classificar os atributos pré-existentes e os propostos por or-

Figure 2: Sequencia de Passos para Encontrar o Melhor Modelo de Predição.

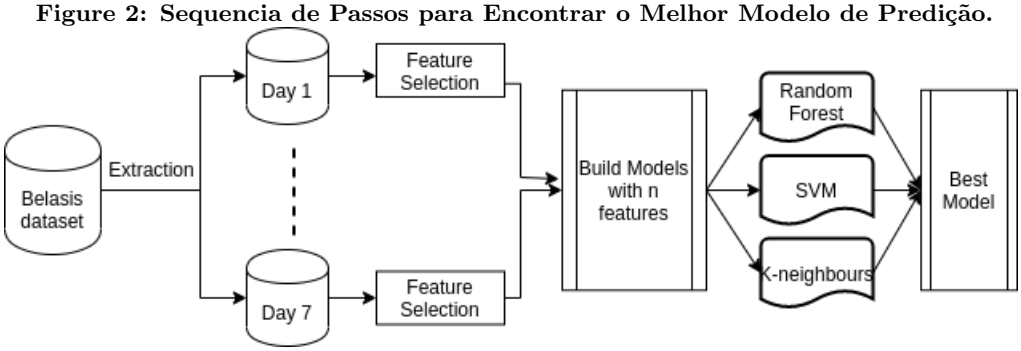


Figure 3: Melhores desempenhos por dia de cada algoritmo.

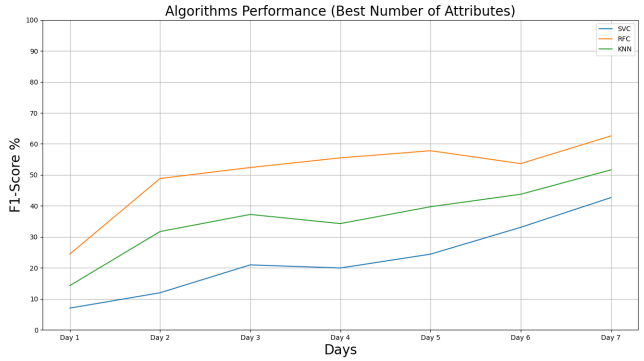


Table 4: Melhores número de atributos por conjunto de dados utilizando o *Random Forest*.

Dia	Número de melhores atributos
1	74
2	77
3	38
4	66
5	73
6	114
7	53

dem de importância, e (*iii*) identificar quais combinação de atributos tiverem o melhor desempenho com o modelo *Random Forest*. Já nesta seção, o objetivo é identificar os melhores valores para os hiperparâmetros para o *Random Forest* e finalmente criar o melhor modelo de predição.

Os hiperparâmetros considerados foram: *n_estimator, criterion, min_weight_fraction_leaf, max_features, min_min_impurity_decrease, bootstrap e class_weight.* Sendo utilizado o pacote `GridSearchCV` da biblioteca *scikit-learn* para a identificação dos melhores valores.

A Tabela 5 apresenta os melhores valores encontrados para cada conjunto de dados. O número de subárvores utilizado (*n_estimator*), por exemplo, foi de 50 a 200, sendo que 200 foi o valor mais frequente nos modelos. Para a maioria dos conjunto de dados, a função de ganho de informação (*criterion*) com melhor desempenho para o modelo foi a *gini*.

A Figura 4 apresenta o resultado final para as métricas de F_1-*score*, precisão e revocação. A precisão e a revocação foram utilizadas neste experimento para auxiliar no entendimento dos resultados encontrados. Pode-se perceber que, com exceção do dia 6, todos os dias possuem um F_1-*score* igual ou melhor ao dia anterior. A explicação intuitiva para o melhoramento incremental do modelo pode ser a de que as ações de um usuário nos primeiros dias podem não condizer com as ações que ele faria em um uso contínuo do sistema (em produção). Por estar no começo do período de teste, o cliente pode estar em uma fase mais exploratória e menos processual.

Ainda sobre a Figura 4, pode-se perceber que o único dia em que a revocação ficou maior do que a probabilidade randômica (50% de chance de um usuário se tornar ativo) foi o quarto dia. Porém, neste mesmo dia a precisão também teve o segundo pior desempenho que, consequentemente, diminuiu o resultado da F_1-*score*. Não foi possível levantar nenhuma hipótese para isso ter ocorrido apenas no quarto dia.

O melhor resultado para a métrica F_1-*score* e para a precisão foi no dia 7. Pelas características de alguns atributos, que derivam seu valor de informações de dias anteriores (como a razão de comandas criadas por profissionais totais cadastrados), pode-se cogitar que essa influência ajude o algoritmo fornecendo um conjunto de dados mais descritivo do usuário.

Para a métrica de precisão, um valor acima de 70% foi alcançado em 3 dos 7 conjuntos de dados utilizados. Já para a métrica de revocação, nenhum do 7 conjuntos de dados utilizados possibilitou um resultado maior que a probabilidade randômica (50%) de acerto. Um modelo com tais resultados representa na prática um nível bom de certeza em relação às predições positivas, mas um nível de certeza ruim em relação às predições negativas.

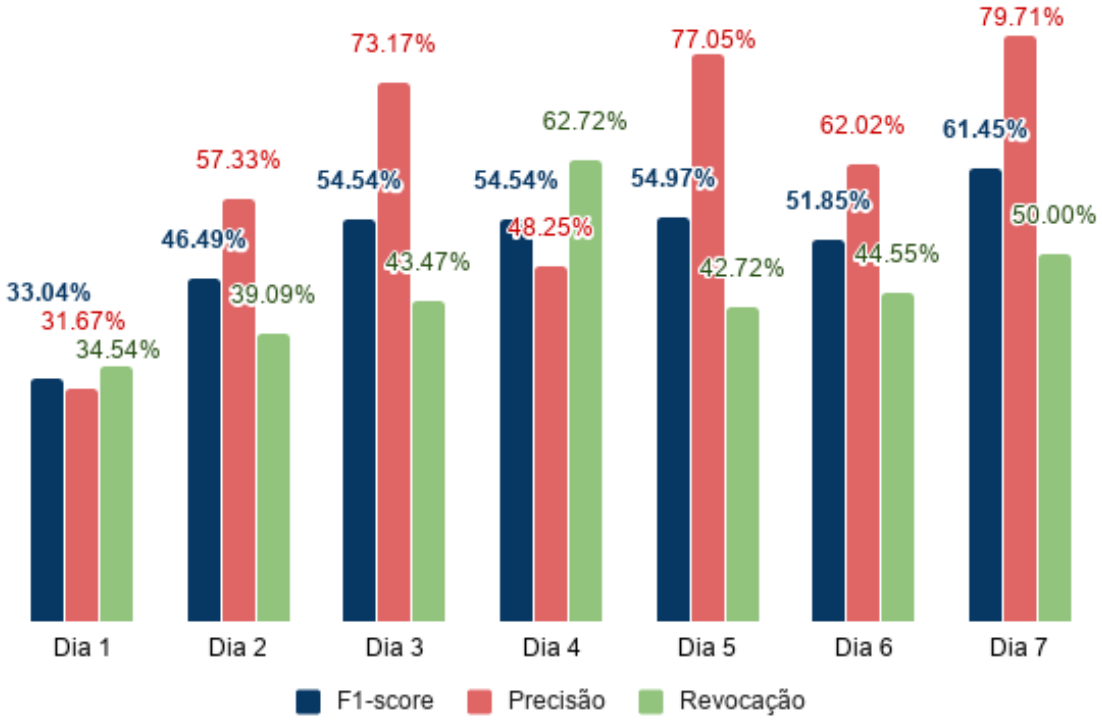
5. CONCLUSÃO

Este trabalho teve como objetivo propor novos atributos e criar um modelo de predição para usuários *Trial* de um produto SaaS a fim de melhorar o grau de sucesso do setor de vendas em transformar os usuários *Trial* em pagantes. Foram testados três algoritmos de aprendizado de máquina e o *Random Forest* foi o que obteve o melhor desempenho segundo a métrica F_1-*score*. A partir dos 36 atributos originais foram propostos mais 87, totalizando 123 atributos. O *Random Forest* teve um maior êxito no conjunto de dados que represente o sétimo dia. Isso era esperado, pois no sé-

Table 5: Melhores combinações de hiperparâmetros para cada dia

Hiperparâmetros	Valores por dia						
	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7
n_estimators	200	200	100	50	100	200	50
criterion	gini	gini	gini	gini	entropy	gini	gini
min_weight_fraction_leaf	0	0	0	.2	0	0	0
max_features	None	auto	sqrt	sqrt	None	auto	sqrt
min_impurity_decrease	0	.1	.1	.1	0	0	0
bootstrap	Sim	Sim	Sim	Sim	Sim	Sim	Sim
class_weight	None	balanced	balanced_subsample	balanced	None	None	None

Figure 4: Melhores métricas alcançadas do experimento separadas por dia



timo dia o usuário está mais familiarizado com o produto e as estatísticas dos atributos criados estão mais consistentes.

O modelo alcançou uma boa precisão, ou seja, os usuários classificados como possíveis assinantes se tornarão realmente assinantes. Para a empresa, o benefício disso é ter menos gasto de recursos humanos com falso-positivos que provavelmente não se tornarão assinantes. Mesmo sendo definida a F_1-*score* como métrica principal para o trabalho, o resultado com uma boa precisão é um ponto a favor do uso de aprendizado de máquina para a predição de clientes.

Por fim, baseando-se nos resultados, uma estratégia de vendas pode ser criada tal que o contato com o cliente aconteça no 3º ou no 5º dia de vida de sua conta. Isso se dá pois o algoritmo teve uma maior precisão nesses dias. Embora o 7º dia possua a maior precisão dos dias, um usuário nessa fase já pode ser atendido por um setor que lida com clientes já ativos. Isso quer dizer que a venda pró-ativa faz mais sentido quando é feita antes do término do *Trial*. É válido, portanto, concluir que o objetivo geral foi em parte alcançado.

Como trabalhos futuros pode-se sugerir: (*i*) utilizar técnicas de correlação entre os dias para a criação de novos atributos, (*ii*) adicionar outros algoritmos de aprendizado de máquina nos experimentos, (*iii*) criar outras combinação de dias, como por exemplo, conjunto de dados correspondente a dois ou mais dias, e (*iv*) considerando o comportamento do conjunto de dados como séries temporais, aplicar abordagens voltadas para os problemas de sazonalidade, tendência e ciclo [8, 2].

6. REFERENCES

[1] C. C. Aggarwal. *Data classification: algorithms and applications*. CRC press, 2014.

[2] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny. An empirical comparison of machine

learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621, 2010.

[3] D. e. G. BRASIL, Ministério do Planejamento. Gabinete do ministro. portaria nº 468, de 22 de dezembro de 2017. *Diário Oficial da União*, Brasília, DF, 26 dez:983, 2017.

[4] D. e. G. BRASIL, Ministério do Planejamento. Gabinete do ministro. portaria nº 442, de 27 de dezembro de 2018. *Diário Oficial da União*, Brasília, DF, 27 dez:517, 2018.

[5] F. B. Bryant and A. Satorra. Principles and practice of scaled difference chi-square testing. *Structural Equation Modeling: A Multidisciplinary Journal*, 19(3):372–398, 2012.

[6] H. Datta, B. Foubert, and H. J. Van Heerde. The challenge of retaining customers acquired with free trials. *Journal of Marketing Research*, 52(2):217–234, 2015.

[7] D. Duarte and N. Ståhl. Machine learning: a concise overview. In *Data Science in Practice*, pages 27–58. Springer, 2019.

[8] R. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software, Articles*, 27(3):1–22, 2008.

[9] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[10] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[11] N. Prasasti, M. Okada, K. Kanamori, and H. Ohwada. Customer lifetime value and defection possibility prediction model using machine learning: An application to a cloud-based software company. In *Asian Conference on Intelligent Information and Database Systems*, pages 62–71. Springer, 2014.

[12] A. Rautio. Churn prediction in saas using machine learning. Master's thesis, Tampere University - Faculty of Management and Business, 2019.

[13] A. Tamaddoni Jahromi, M. M. Sepehri, B. Teimourpour, and S. Choobdar. Modeling customer churn in a non-contractual setting: the case of telecommunications service providers. *Journal of Strategic Marketing*, 18(7):587–598, 2010.

[14] W. Tsai, X. Bai, and Y. Huang. Software-as-a-service (SaaS): perspectives and challenges. *Science China Information Sciences*, 57(5):1–15, 2014.

[15] W. Verbeke, D. Martens, C. Mues, and B. Baesens. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert systems with applications*, 38(3):2354–2364, 2011.

[16] B. Waters. Software as a service: A look at the customer benefits. *Journal of Digital Asset Management*, 1(1):32–39, 2005.

[17] J. K. Yan and R. Wakefield. Cloud storage services: Converting the free-trial user to a paid subscriber. In *Thirty Sixth International Conference on Information Systems*. Association for Information Systems, 2015.

APPENDIX

A. ATRIBUTOS ORIGINAIS DO CONJUNTO DE DADOS

- **agendamentos**, **agendamentos_cancelados**, **agendamentos_confirmados**, **agendamentos_nao_confirmados**: número de agendamentos cadastrados no sistema. O primeiro atributo indica o número total de agendamentos e os outros 3 o número por *status* do agendamento. O *status* "cancelado"é usado quando o cliente cancela previamente o agendamento. O *status* "não confirmado"representa um agendamento que ainda não foi confirmado pelo cliente e o "confirmado"representa um agendamento que já foi confirmado. Essa confirmação pode ser feitas de várias formas e não entram no escopo do sistema.
- **anamneses**: número de anamneses criadas no sistema. Uma anamnese precisa, necessariamente, estar atrelada a um cliente. A anamnese é um formulário dinâmico usado para entrevistas com os clientes e pacientes. Atributo não considerado na construção dos modelos.
- **avaliacoes**: número de avaliações feitas pelos clientes dos estabelecimentos. Uma avaliação está atrelada a um serviço realizado e cadastrado em uma comanda. Uma comanda pode ter vários serviços, portanto, uma comanda pode ter várias avaliações.
- **campanhas**: campanhas são formas de divulgar promoções, anúncios ou qualquer outra coisa que o usuário queira mostrar no *website* personalizado da empresa ou enviar por meio de mensagens de texto para seus clientes. Atributo transformado em número que representa a quantidade de campanhas realizadas.
- **categorias**: categorias de produtos. Servem para definir comissões por categoria ou apenas para agrupar e organizar a lista de produtos. Atributo não considerado na construção dos modelos.
- **clientes**: cadastro dos clientes das empresas. Consta, nesse cadastro, uma grande gama de informações do cliente, que vão desde e-mail, nome, endereço a dados mais específicos como dependentes (ligação entre um cliente e outro), pontos do programa de fidelidade etc. Atributo não considerado na construção dos modelos.
- **comandas** e **comandas_finalizadas**: a comanda registrada a presença do cliente no que tange ao consumo e serviços realizados. Uma comanda finalizada é uma comanda que já foi paga pelo cliente e o usuário deu baixa no sistema, atrelando pagamentos e exibindo-os no módulo do financeiro.
- **compras**: registros de compras realizadas pelo estabelecimento. São também uma forma de dar entrada de um produto no estoque.
- **compras_total**: o valor total pago nas compras registradas no sistema.
- **cores_agendamentos**: cores personalizadas para os agendamentos. As cores personalizadas servem para dar visibilidade de um agendamento com base em seu status ou o que ele significa. Um exemplo para isso é quando um cliente tem que voltar a um salão para fazer o "retrabalho". Nessa situação, o cliente em questão tem que ser tratado com um tato diferente, e por

isso a cor no agendamento serve como um sinal visual para aqueles profissionais que o atenderão. Atributo não considerado na construção dos modelos.

- **despesas**: quantidade de registros de despesas criados. Uma despesa é uma saída monetária. Tipos de despesas incluem pagamento de salários, comissões, produtos, etc.
- **dia_segunda_feira**, **dia_terca_feira**, **dia_quarta_feira**, **dia_quinta_feira**, **dia_sexta_feira** e **dia_sabado**, **dia_domingo**: atributos *booleanos* que diz qual o dia da semana do *n*-ésimo dia do usuário do exemplo. Por exemplo, se é analisado o conjunto de dados do dia 3 e o 3º dia de teste de um usuário caiu na data 16 de novembro de 2019, então o atributo *dia_sabado* terá como valor "Verdadeiro" enquanto os outros 6 atributos terão como valor "Falso". A ideia desse atributo é levar em conta se o uso do sistema por parte do usuário ser maior ou menor pode ser justificado pelo dia da semana (menos uso no domingo e mais uso nas sextas-feira, por exemplo).
- **fechamento_caixa**: quantidade de fechamento de caixa realizado pelo usuário. Um fechamento de caixa é uma conferência de caixa. O fluxo de uso normalmente é iniciado no começo do turno do estabelecimento com a abertura do caixa e, ao final, é feito uma conferência para checar se o dinheiro físico no caixa é igual ao registrado no sistema. Daí, então, o caixa é fechado e um registro é criado.
- **feriado**: atributo que diz se o dia da semana do *n*-ésimo dia do usuário do exemplo é feriado. Por exemplo, se o 3º dia de teste de um usuário caiu no dia 25/12/2018, então esse atributo terá valor verdadeiro. A definição das datas de feriado está em [3] e [4].
- **fornecedores**: quantidade de fornecedores cadastrados no sistema. Um registro é uma descrição de fornecedor. Pode ser usado nas compras para registro de entrada de produtos.
- **marcas**: número de marcas de produtos. Serve apenas para organização e agrupamento de produtos. Atributo não considerado na construção dos modelos.
- **movimentacao_entrada** e **movimentacao_saida**: valor monetário movimentado. Grava o que foi recebido e o que foi enviado, respectivamente. A *movimentacao_saida* diz respeito aos valores das *despesas* cadastradas. A *movimentacao_entrada* diz respeito aos recebimentos cadastrados.
- **pacotes** e **pacotes_finalizados**: pacotes são produtos ou serviços que são comprados antecipadamente ao seu uso. Um exemplo disso é quando um cliente de uma empresa vai ao salão para cortar o cabelo todo mês. O estabelecimento pode vender, então, 12 cortes (um ano de corte, portante) por um preço menor, de uma vez. A partir daí, o cliente não precisa pagar o corte enquanto este pacote comprado tiver saldo suficiente. Um pacote pode ser simplesmente salvo ou faturado. Ao ser faturado, ele se torna um pacote finalizado.
- **pacotes_pre_definidos**: ao vender um pacote, o usuário precisa escolher quais são os produtos que serão vendidos manualmente. Um pacote pré definido é uma facilidade para o usuário que permite o preenchimento automático de pacotes com definições pré-existentes.
- **profissionais**: número de profissionais cadastrados no sistema. O registro do profissional é usado para a criação de agendamentos, comandas, registro de pagamento de comissões, etc.
- **recebimentos**: recebimentos são transações em que o beneficiado é o estabelecimento. Pode ser um pagamento por um serviço prestado, por exemplo.
- **servicos_produtos**: quantidade de produtos ou serviços cadastrados. Um registro de serviço é necessário para a criação de um agendamento, de uma comanda, entre outros.
- **vales**: registros de pagamentos adiantados a um profissional. Está relacionado com uma despesa, já que ao pagar o profissional, o estabelecimento gera uma saída de dinheiro (a despesa, no caso).
- **venda_total**: valor total de venda cadastrado. É a soma dos valores de todas as comandas criadas.

B. LISTA DE ATRIBUTOS GERADOS

1. agendamentos_cancelados_d_por_profissional_t
2. agendamentos_cancelados_media
3. agendamentos_cancelados_mediana
4. agendamentos_confirmados_d_por_cliente_t
5. agendamentos_confirmados_d_por_cliente_t
6. agendamentos_confirmados_d_por_profissional_t
7. agendamentos_confirmados_media
8. agendamentos_confirmados_mediana
9. agendamentos_d_por_cliente_t
10. agendamentos_d_por_profissional_t
11. agendamentos_media
12. agendamentos_mediana
13. agendamentos_nao_confirmados_d_por_profissional_t
14. agendamentos_nao_confirmados_media
15. agendamentos_nao_confirmados_mediana
16. anamneses_media
17. anamneses_mediana
18. avaliacoes_d_por_comanda_d
19. avaliacoes_d_por_profissional_t
20. avaliacoes_media
21. avaliacoes_mediana
22. campanhas_media
23. campanhas_mediana
24. categorias_media
25. categorias_mediana
26. clientes_d_por_profissional_t
27. clientes_media
28. clientes_mediana
29. comandas_d_por_cliente_t
30. comandas_d_por_profissional_t

31. comandas_finalizadas
32. comandas_finalizadas_d_por_cliente_t
33. comandas_finalizadas_d_por_comanda_d
34. comandas_finalizadas_d_por_profissional_t
35. comandas_finalizadas_media
36. comandas_finalizadas_mediana
37. comandas_media
38. comandas_mediana
39. compras_media
40. compras_mediana
41. compras_total
42. compras_total_media
43. compras_total_mediana
44. despesas_d_por_profissional_t
45. despesas_media
46. despesas_mediana
47. fechamento_caixa_media
48. fechamento_caixa_mediana
49. fornecedores_media
50. fornecedores_mediana
51. marcas_media
52. marcas_mediana
53. movimentacao_entrada_d_por_cliente_t
54. movimentacao_entrada_d_por_profissional_t
55. movimentacao_entrada_media
56. movimentacao_entrada_mediana
57. movimentacao_saida_d_por_profissional_t
58. movimentacao_saida_media
59. movimentacao_saida_mediana
60. pacotes_d_por_cliente_t
61. pacotes_d_por_profissional_t
62. pacotes_finalizados_d_por_cliente_t
63. pacotes_finalizados_d_por_profissional_t
64. pacotes_finalizados_media
65. pacotes_finalizados_mediana
66. pacotes_media
67. pacotes_mediana
68. pacotes_pre_definidos_d_por_profissional_t
69. pacotes_pre_definidos_media
70. pacotes_pre_definidos_mediana
71. profissionais_media
72. profissionais_mediana
73. recebimentos_d_por_cliente_t
74. recebimentos_d_por_profissional_t
75. recebimentos_media
76. recebimentos_mediana
77. servicos_produtos_d_por_profissional_t
78. servicos_produtos_media
79. servicos_produtos_mediana
80. vales_d_por_profissional_t
81. vales_media
82. vales_mediana
83. venda_total_d_por_cliente_t
84. venda_total_d_por_comanda_d
85. venda_total_d_por_profissional_t
86. venda_total_media
87. venda_total_mediana