Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Estudo de comportamento de usuários de um app: transição de Freemium para Premium

Alex Nascimento Souza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Prof.ª Dr.ª Genaina Nunes Rodrigues

Brasília
2021

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

:

Banca examinadora composta por:

Prof.ª Dr.ª Genaina Nunes Rodrigues (Orientadora) — CIC/UnB
Prof. Dr. Luís Paulo Faina Garcia — CIC/UnB
Prof. Dr. Vinícius Ruela Pereira Borges — CIC/UnB

CIP — Catalogação Internacional na Publicação

Souza, Alex Nascimento.

Estudo de comportamento de usuários de um app: transição de Freemium para Premium / Alex Nascimento Souza. Brasília : UnB, 2021. 87 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2021.

1. comportamento, 2. usuário, 3. freemium, 4. premium,
5. aprendizagem de máquina, 6. clusterização

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Estudo de comportamento de usuários de um app: transição de Freemium para Premium

Alex Nascimento Souza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof.ª Dr.ª Genaina Nunes Rodrigues (Orientadora)
CIC/UnB

Prof. Dr. Luís Paulo Faina Garcia
CIC/UnB

Prof. Dr. Vinícius Ruela Pereira Borges
CIC/UnB

do Bacharelado em Ciência da Computação

Brasília, 25 de Maio de 2021

Dedicatória

Dedico esse trabalho ao meu pai, que sempre sonhou ter um filho formado em uma universidade pública e sempre me incentivou a estudar para atingir meus objetivos, à minha mãe, que sempre me apoiou em todas as decisões e fez de tudo para que eu chegasse aqui e ao meu irmão que foi um dos grandes mentores da minha vida. Não posso esquecer dos meus amigos, que me fazem ser mais forte a cada dia. Sem eles, simplesmente não sou.

Eu amo vocês.

Agradeço à todos os professores que fizeram parte da minha trajetória, em especial a orientadora Prof.ª Dr.ª Genaina Nunes Rodrigues por todo o suporte na faculdade, desde matérias, papos descontraídos, projetos de pesquisa até o trabalho de conclusão de curso. Uma das professoras mais humanas que conheci na minha caminhada pela UnB.

Aproveito para agradecer a todos que contribuíram direta ou indiretamente para a realização do presente trabalho.

Resumo

A análise de comportamentos de usuários na transição de um modelo Freeemium para um modelo Premium tem sido cada vez mais enfatizada por empresas que adotam esse modelo de negócio. Entender o comportamento do usuário em uma plataforma permite com que perfis diferentes sejam identificados viabilizando, assim, que produtos e serviços mais alinhados com esses perfis sejam criados. Este trabalho tem o intuito de realizar um estudo de comportamento de usuários de um aplicativo com fins sociais, combinando sua base de dados com métodos de Mineração de Dados e Aprendizagem de Máquina para clusterizar os usuários e entender quais atributos são mais relevantes para a clusterização utilizando uma abordagem de aprendizagem supervisionada combinada com uma abordagem de aprendizagem não supervisionada. Além disso, o trabalho tem como objetivo entender qual segmento gerado pela clusterização contêm mais usuários assinantes, permitindo a criação de segmentos específicos para esses usuários.

Palavras-chave: comportamento, usuário, freemium, premium, aprendizagem de máquina, clusterização

Sumário

Lista de Figuras

Capítulo 1

Introdução

O surgimento de smartphones no início do século, mais precisamente o lançamento do iPhone em 2007 [10], possibilitou com que vários usuários experimentassem carregar computadores que cabiam em seus respectivos bolsos. Além da vantagem em relação às suas dimensões físicas, os smartphones revolucionaram a indústria de *software* ao evidenciar o uso de aplicações *mobile*, conhecidas como *apps*. A intensa popularização desse tipo de dispositivo implicou diretamente na rápida adoção e implementação de tecnologias mobile, atigindo a marca de um milhão de apps desenvolvidos ao final do ano de 2011 [25].

Paralelamente, o crescimento tecnológico advindo da massiva utilização de aplicativos permitiu com que várias empresas reformulassem ou gerassem seus modelos de negócio, isto é, a forma com que essas capturam, criam e entregam valor. Sendo assim, foi necessário estabelecer canais de comunicação com seus clientes pelo meio *mobile*, assim alavancando suas respectivas receitas [18]. Atualmente, a plataforma de *streaming* musical *Spotify*, o app de comunicação *Whatsapp* e o serviço de transporte *Uber* enfatizam a predominância na utilização de *apps* na indústria atual.

No que tange à modelos de negócio, empresas optam por operar com um modelo *freemium*, que é uma variação moderna do modelo de assinaturas. No modelo de assinaturas, a empresa concede serviços ou produtos ao usuário a partir de um pagamento periódico feito por ele, geralmente mensal. Ao passo que o modelo *freemium* consiste em atuar no mercado utilizando dois produtos ou serviços ou a combinação de ambos [24]. Nessa combinação, um dos serviços ou produtos é oferecido de forma gratuita, ao passo que o segundo é vendido por um preço específico. Vale a pena ressaltar, também, que o produto ou serviço vendido por um preço específico possui uma qualidade maior em relação ao gratuito, além de conter mais funcionalidades.

Modelos de negócio como esse geram uma imensa quantidade de dados, seja por meio dos dados cadastrados no banco de dados da aplicação, seja por ações específicas que os usuários executam durante a utilização do *app*. Sendo assim, torna-se necessária uma solução para armazenar e processar tais dados. Além disso, a utilização de técnicas de aprendizagem de máquina e mineração de dados para entender melhor o comportamento dos usuários pode se tornar uma importante vantagem competitiva frente aos pares no mercado. Essa prática possibilita que sejam geradas ideias baseadas em dados para novos serviços, além de atingir ramos específicos de consumidores por meio de incentivos de *marketing*.

Para um tema tão relevante como o de compreensão de comportamento de usuários em modelos de negócios *Freemium*, como proposto neste trabalho, bases sólidas de conhecimentos foram pesquisadas e utilizadas. Como a proposta é entender o comportamento dos usuários agroupando-os em grupos semelhantes a partir da utilização de um aplicativo, conceitos relativos à clusterização e análise de dados foram abordados. Dessa forma, foi possível mapear cada cluster e avaliar os conjuntos gerados utilizando a metodologia apresentada em [23].

1.1 Problema de pesquisa

Apesar da disponibilidade dos dados, muitas empresas não sabem utilizá-los de forma a entender o comportamento dos usuários e converter o maior número de usuários possíveis do modelo *Freemium* para o modelo *Premium*. Um estudo realizado pela KPMG Capital com mais de 130 executivos de finanças e inteligência de mercado mostrou que 85% dos envolvidos no estudo afirmar enfrentar desafios para implementar soluções corretas a partir dos dados gerados, além de encontrarem dificuldades para interpretar os dados existentes [16]. Assim, sofrem da falta de informações baseadas em dados ao não identificar quais os fatores mais relevantes para essa conversão e como eventualmente medí-los.

Mesmo com os dados disponíveis, ainda é necessário contar com um profissional que entenda o modelo de negócio por trás da empresa a fim de facilitar a compreensão dos dados e possibilitar que estes sejam explorados da maneira correta. Posto isso, o principal problema de pesquisa é justamente observar e entender quais comportamentos do uso do aplicativo indicam que um usuário do modelo *Free* pode migrar para o modelo *Premium*, elecando quais ações dentro do aplicativo são mais relevantes.

1.2 Objetivo

O objetivo principal desse trabalho é desenvolver um modelo preditivo que utilize técnicas de mineração de dados e aprendizagem de máquina a fim de entender quais comportamentos de usuários de um aplicativo os impulsionam a tornarem-se usuários *Premium*, aplicando uma variação da metodologia apresentada no artigo *A Study of App User Behaviours: Transitions From Freemium to Premium* [23].

O aplicativo usado como base no estudo será nomeado como Aplicativo Social em razões de confidencialidade e de proteção de dados e tem como missão tornar o ato de doação cada vez mais comum no mundo com uma abordagem interativa. O Aplicativo Social permite com que usuário coletem moedas por meio de várias interações, como ler notícias de impacto social ou realizar atividades diárias dentro do aplicativo. Essas moedas podem ser então destinadas à diversas ONGs brasileiras. Um usuário torna-se *premium* a partir do momento que este realiza uma assinatura mensal de um pacote de moedas para realizar ainda mais doações.

A abordagem é baseada em três etapas principais: (1) extração dos dados por meio de consultas SQL à ferramentas específicas utilizadas pelo negócio; (2) pré-processamento dos dados, que consiste na limpeza, organização e estruturação dos dados; e (3) agrupamentoxz e avaliação dos rótulos estabelecidos para cada cluster. Os dados utilizados nesse estudo pelo Aplicativo Social são resultantes de ações de usuários do App, que são

gravados em formas de eventos. Eventos são estruturas de dados compostas por uma única palavra de até 32 caracteres e que são armazenadas em um banco de dados específico. Sendo assim, esses dados são coletados e pré-processados para, posteriormente, serem agrupados e classificados. O Capítulo 4 exemplifica as etapas do processo.

1.3 Justificativa

A importância de se tomar decisões baseadas em dados já é um conceito consolidado, e fazer isso de forma replicável e automatizada é essencial para garantir o bom andamento de um negócio [19].

O estudo do comportamento de usuários se justifica devido à alta competitividade no mercado, sobretudo o valor entregue ao cliente, que neste caso é o usuário do sistema, considerando quais são as características desse público-alvo e como realizar entregas mais alinhadas às demanadas que eles geram. Assim, é possível trabalhar de maneira mais assertiva a rentabilidade do negócio. Para o estudo de caso em questão, existe, ainda, o fato de que para cada usuário que torna-se *Premium*, a quantia doada para as fundações aumenta gradativamente.

Além disso, o modelo utilizado no presente trabalho pode ser explorado de maneiras diferentes, de tal forma que o objetivo seja, por exemplo, compreender o comportamento de usuários em relação ao seu tempo de permanência no app ou abarcar o nível de satisfação dos usuários.

O processo de desenvolvimento do modelo construído, tão bem quanto os conceitos que envolvem sua construção são mostrados no Capítulo 2, como parte da fundamentação teórica deste trabalho. Conceitos estes que foram utilizados para a implementação do modelo apresentado nesta pesquisa.

1.4 Contribuições

A contribuição do presente trabalho consiste em um modelo de aprendizagem de máquina que combina as abordagens supervisionadas e não supervisionadas para analisar o comportamento de usuários em relação à transição do modelo *Freemium* para *Premium* de um aplicativo.

A análise do comportamento é realizada de tal modo que os usuários são agrupados de acordo com as ações realizadas durante a utilização do aplicativo. Por fim, o modelo ressalta quais atributos são os mais relevantes para a formação dos agrupamentos.

1.5 Organização do trabalho

Os demais capítulos desta monografia estão organizados da seguinte forma: O Capítulo 2 aborda a fundamentação teórica relativa aos conceitos utilizados neste trabalho. As bases apresentadas neste capítulo são utilizadas para a implementação do modelo proposto.

No Capítulo 3 é realizada uma revisão da literatura, mostrando conceitos e ideias semelhantes ao que é apresentado neste trabalho, e que são utilizados como referência.

O Capítulo 4 apresenta o processo de criação do modelo, assim como descrito na seção 1.2. O processo é dividido em etapas e apresentado em formato de diagrama. Além disso, dado que o processo de desenvolvimento é inerente ao estudo de caso, o capítulo explicita o que foi utilizado como objeto de estudo.

No Capítulo 5 são discutidos os resultados experimentais, tão bem quanto a validação do modelo levando em consideração as métricas escolhidas para avaliar sua performance.

Por fim, o Capítulo 6 apresenta a conclusão do trabalho.

Capítulo 2

Fundamentação Teórica

2.1 Eventos

Eventos são estruturas de dados compostas por uma única palavra de até 32 caracteres e que são armazenadas em um banco de dados específico. Os eventos representam quaisquer insights que estão acontecendo em uma aplicação, seja ações de usuários, eventos do sistema ou erros.

A depender da ferramenta utilizada, existem eventos que são disparados automaticamente, ao passo que é possível também personalizá-los a ponto de mudar seus nomes e adicionar parâmetros de contexto. Um exemplo de evento é o seguinte trecho de código:

```
1    analytics.logEvent('select_content', {
2        content_type: 'image',
3        content_id: 'P12453',
4    });
```

No código exemplificado acima, o evento "*select_ content*" é disparado com os parâmetros "*content_ type*" e "*content_ id*". Vale a pena ressaltar que os eventos recebem somente *strings* como parâmetros de contexto.

Sendo assim, os eventos disparados pelos usuários do Aplicativo Social serão recuperados de um banco de dados. A partir dos eventos recuperados, serão gerados atributos utilizados como entrada para o modelo proposto nesta pesquisa.

2.2 Aprendizagem Supervisionada e Não-Supervisionada

O conceito de aprendizagem de máquina está intrinsecamente ligado com o conceito de mineração de dados. Sendo assim, faz-se importante ressaltar este conceito.

A mineração de dados é definida como um conjunto específico de métodos e algoritmos visando unicamente a extração de padrões de uma base de dados [13]. Após a extração de padrões, é possível levantar informações relevantes de uma maneira entendível para humanos.

A mineração de dados pode ser dividida em dois tipos. O primeiro tipo, conhecido como detecção de padrões, consiste em procurar por regularidades ou anomalias no conjunto de dados. O segundo tipo, conhecido como construção de modelos, consiste em resumir grandes partes de dados de uma forma conveniente [12].

Dentro do segundo tipo já citado, está o conceito de aprendizagem de máquina. Aprendizagem de máquina (do inglês, *Machine Learning*) é a área que investiga como o computador consegue aprender, ou melhorar sua performance, baseado em dados [11]. A área de Aprendizagem de Máquina desenvolve programas computacionais que aprendem e reconhecem padrões complexos automaticamente, além de fazer decisões inteligentes a partir do estudo feito nesses dados [11].

Para a execução de um projeto de aprendizagem de máquina, é necessário ir além dos dados disponíveis, levando em consideração qual o problema que está tentando ser resolvido e em qual das três categorias de aprendizagem de máquina o problema se encaixa [11]:

1. Aprendizagem supervisionada: os rótulos que serão previstas para cada elemento do conjunto de dados são previamente conhecidas. Um exemplo de aprendizagem de máquina supervisionada é a categorização de e-mails, separando entre e-mails que são relevantes para o usuário e e-mails considerados *spam*.

2. Aprendizagem não supervisionada: não há rótulos previamente conhecidos para as instâncias do conjunto de dados. Assim, o algoritmo é responsável por estabelecer padrões e identificar as classes. Um sistema de recomendação pode ser tomado como exemplo para uma aprendizagem de máquina não supervisionada, uma vez que é necessário descobrir o gosto do usuário à medida que as recomendações são propostas.

3. Aprendizagem por reforço: é uma mistura entre aprendizagem supervisionada e não supervisionada. Um algoritmo de apredizagem por reforço é avisado quando a classe é prevista de maneira errada, mas não é avisado como prever a classe corretamente. Os avisos recebidos pelo algoritmo são oriundos do ambiente que ele está inserido. A aprendizagem de máquina por reforço pode ser exemplificada no estudo de carros autônomos.

O problema proposto nesta pesquisa se encaixa tanto na categoria de aprendizagem não supervisionada, uma vez que determinamos os clusters a partir dos atributos utilizados como entrada para o modelo sem quaisquer características prévias e quanto na categoria de aprendizagem supervisionada, uma vez que realizamos previsões acerca dos rótulos já gerados pelo processo de agrupamento.

2.2.1 Classificação

Classificação é o processo no qual o modelo ou classificador é construido para prever os rótulos de dados, que são escolhas de uma lista pré-definida de possibilidades. A classificação dos dados consiste em duas etapas: aprendizagem (ou etapa de treino) e classificação (ou etapa de teste) [11].

Durante a etapa de treinamento, onde o modelo é construído, o algoritmo de classificação utiliza um conjunto de treino composto por tuplas e seus respectivos rótulos. A i-ésima tupla x_i consiste em um vetor de atributos de n dimensões:

$$x_i = (x_1, x_2, ..., x_n) \tag{2.1}$$

Onde x_n representa o n-ésimo atributo da i-ésima tupla. O conjunto de todas as tuplas x do conjunto de dados é chamado de X, além disso, faz-se importante ressaltar que cada tupla x possui uma classe pré-definida.

Por fim, o conjunto de rótulos a serem previstos é chamado de Y. A depender do conjunto Y, a classificação pode ser considerada binária, caso Y assuma dois valores ou multiclasse caso Y assuma mais de dois valores.

Assim, no escopo deste trabalho, a classificação será multiclasse, uma vez que o rótulo a ser previsto é o rótulo gerado pelo processo de clusterização.

Métricas de Avaliação de Modelos

Para avaliar se a classificação resultante do modelo contém resultados satisfatórios são usadas métricas de performance. Entre as métricas utilizadas nessa pesquisa estão Precisão, Sensibilidade e F-Score.

A métrica de Precisão mede quantas instâncias previstas como positiva são, de fato, positivas e é referenciada pela equação 2.2. Para uma classificação multiclasse, a precisão é dada pela seguinte equação [30], em que l é o número de rótulos possíveis, tp é o total de positivos verdadeiros e fp é o total de falsos positivos, todos obtidos após a indução do modelo:

$$Precision = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fp_i)} \tag{2.2}$$

A métrica de Sensibilidade (ou *Recall*) é descrita pela equação 2.3 mede quantas instâncias positivas foram corretamente rotuladas pelas previsões positivas, ou seja, das instâncias que eram positivas quantas foram detectadas. Para a classificação multiclasse, a sensibilidade é dada pela equação abaixo [30], onde l é o número de rótulos possíveis, tp é o total de positivos verdadeiros e fn é o total de falsos negativos:, obtidos após a indução do modelo

$$Recall = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fn_i)} \tag{2.3}$$

Por sua vez, a métrica F-Score é dada pela equação 2.4 e representa a média harmônica entre as duas métricas mencionadas anteriormente e é descrita pela equação a seguir:

$$F\text{-}Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.4}$$

Por fim, uma forma visual de compreender a relação entre instâncias positivas e negativas é a matriz de confusão. A matriz de confusão é uma Tabela CM_{ij} de tamanho nxn, onde n é o número total de rótulos possíveis a serem previstos [11]. Cada entrada da matriz indica o número de rótulos i que foram classificados pelo algoritmo como o rótulo j.

Árvores de Decisão

Árvore de decisão é um modelo preditor hierárquico composto por regras de decisão que prevê a classe de uma instância x. A estrutura da árvore de decisão assemelha-se à

estrutura de um fluxograma em árvore, no qual cada nó não-folha representa um teste em um atributo, isto é, se atende ou não ao critério posto em questão. Cada ramificação corresponde à resposta de cada teste e cada nó folha contém um rótulo de classe [11].

Além disso, a árvore, durante sua montagem, utiliza a abordagem *top-down* particionando o conjunto original de instâncias (atributos) recursivamente em subconjuntos de tal forma a encontrar um modelo informativo e robusto de classificação [21]. Um exemplo visual de uma árvore de decisões pode ser observado na Figura 2.1 abaixo, que tem como exemplo um conjunto de dados de pétalas, em que cada nó da árvore representa a largura da pétala em centímetros. O objetivo dessa árvore é classificar uma flor de acordo com essa característica.

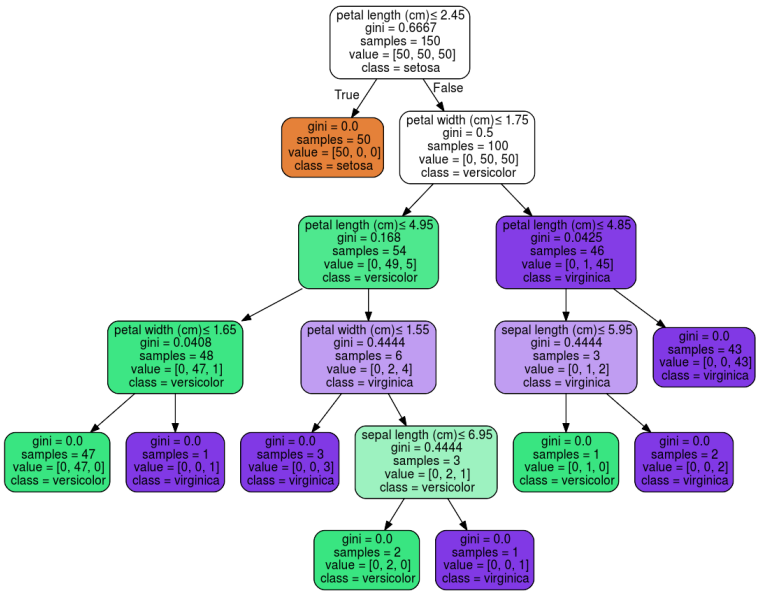

Figura 2.1: Estrutura de uma árvore de decisão [1]

Para melhor avaliar as divisões da árvore de decisão, o algoritmo utiliza um critério de pontuação para avaliar as partições da árvore. Dois dos critérios utilizados são ganho de informação e índice de Gini. O critério de ganho de informação, ou entropia, baseado no trabalho de Claude Shannon acerca da teoria da informação [28], consiste em escolher o atributo com maior ganho de informação a fim de minimizar a informação necessária para a partição dos nós e diminuir a impureza nesse processo. Assim, dado um nó t, a quantidade de informação $I(t)$ pode ser dada pela equação 2.5:

$$I(t) = -\sum_{i=1}^{m} (p_i) \log_2 (p_i) \tag{2.5}$$

onde, p_i é a porcentagem de cada classe das instâncias no nó t e m é a quantidade de rótulos distintos no conjunto de dados em análise. Ademais, o atributo a que maximiza o ganho de informação é selecionado para a divisão do nó t. O ganho de informação é calculado pela equação 2.6

$$G(a) = I(t) - I_a(t) \tag{2.6}$$

[1]Imagem acessada em Abril de 2021: https://scikit-learn.org/stable/modules/tree.html

onde a representa um atributo, $I(t)$ a quantidade de informação no nó t e $I_a(t)$ a informação esperada para classificar um nó t baseado no atributo a, definida pela equação 2.7:

$$I_a(t) = \sum_{j=1}^{v} \left(\frac{|N_j|}{|N|} \right) \times (I(N_j)) \tag{2.7}$$

onde v é a quantidade de partições geradas da divisão, $|N|$ representa o número de instâncias no nó t, $|N_j|$ é a quantidade de instâncias no j-ésimo nó depois da divisão pelo atributo a e $I(N_j)$ representa a quantidade de informação do j-ésimo nó.

O critério do índice de Gini mede a impureza, que é a diferença da distribuição de frequência das classes de um conjunto de instâncias D tal que

$$G(D) = 1 - \sum_{i=1}^{m} p_i^2 \tag{2.8}$$

em que m é a quantidade de rótulos, p_i é a probabilidade de que uma instância em D pertença a uma classe C_i e é estimada pela equação 2.9:

$$p_i = |C_{i,D}|/|D| \tag{2.9}$$

Na equação acima, $C_{i,D}$ representa a quantidade de rótulos C_i presentes no conjunto de instâncias D. Para divisões por atributos, o índice de Gini, onde D_1 e D_2 são subconjuntos de D formados a partir do atributo a é dado pela equação 2.10:

$$Gini_a(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \tag{2.10}$$

Assim, o atributo que minimizar a índice de Gini é escolhido para a divisão na árvore de decisão [11].

Florestas Aleatórias

O algoritmo de florestas aleatórias proposto por Leo Breiman [5] é um algoritmo do tipo *ensemble*. Esse paradigma consiste em combinar classificadores a fim de encontrar uma solução para o mesmo problema. Cada classificador vota e o rótulo da classe prevista é retornado de acordo combinação dos votos de cada classificador.

O algoritmo de florestas aleatórias combina a utilização de uma coleção de árvores de decisão, gerando, assim, o nome de "florestas". Deste modo, cada árvore depende dos valores de um vetor aleatório φ_k, que é uma amostra das características da instância escolhida de forma independente e com a mesma distribuição para todas as árvores da floresta [11].

Gradient Boosting Machine

Gradient Boosting Machine é outro algoritmo que utiliza o método de *ensemble* que combina múltiplas árvores de decisão para criar um modelo mais poderoso, além de utilizar a técnica de *Boosting*. A técnica de *boosting* consiste em atribuir pesos à cada instância treinada pelo modelo.

Assim, após o treino do classificador M_i, os pesos são atualizados permitindo com que o próximo classificador M_{i+1} corrija o erro cometido pelo classificador anterior. O processo é repetido até que o erro não sofra mudança ou o número limite de árvores tenha sido gerado [11].

De uma forma genérica, o objetivo do algoritmo é encontrar a função $F*(x)$ tal qual seu valor seja minimizado para uma função de perda $\psi(y, F(x))$, em que x é o conjunto de atributos e y é o conjunto de rótulos de classe [9], assim como demonstrado na equação 2.11:

$$F*(x) = \arg\min_{F(x)} \psi(y, F(x)) \tag{2.11}$$

A função de perda aplicada ao classificador M_i, para o contexto da pesquisa, é semelhante à utilizada para o algoritmo *AdaBoost (adaptative boosting)* e é descrita pela equação 2.12:

$$error(M_i) = \sum_{j=1}^{d} w_j \times err(X_j) \tag{2.12}$$

onde d é o número de instâncias, w é o peso aplicado à instância que foi classificada incorretamente e X_j é uma instância do conjunto de dados [11]. Caso a instância tenha sido classificada de maneira incorreta, $err(X_j)$ recebe o valor 1, caso contrário recebe 0. Por fim, para cada instância classificada corretamente, seu peso é multiplicado pela equação 2.13:

$$\frac{error(M_i)}{(1 - error(M_i))} \tag{2.13}$$

2.2.2 Agrupamento

Além dos algoritmos de classificação, o presente trabalho também conta com a utilização de algoritmos de agrupamento. Agrupamento é o processo que visa separar um conjunto finito de dados não rotulados em um conjunto discreto e também finito de dados [31], denominado *clusters*, ou "agrupamentos" em português.

O objetivo do agrupamento é dividir os dados de tal forma que os pontos de cada cluster sejam parecidos e estejam no mesmo cluster enquanto os pontos com características diferentes estejam em clusters separados [20]. Igualmente processos de classificação, a clusterização fornece um ponto para cada instância denominando o número do cluster desta, conforme ilustrado na Figura 2.2, onde os pontos vermelhos podem ser dados pelo rótulo 1 e os verdes pelo rótulo 2.

Neste trabalho, o algoritmo de agrupamento utilizado foi o algoritmo *K-Means*. Sendo assim, a próxima seção é responsável por explicar como ele funciona.

O algoritmo *K-Means* consiste em aplicar a técnica de clusterização de tal forma a procurar o centróide do cluster C_i como forma representativa dos dados, sendo assim, o algoritmo visa diminuir o critério conhecido como *inertia*, que é a soma dos quadrados da distância de cada ponto para seu centróide mais próximo.

[2]Imagem acessada em Abril de 2021: https://blogs.oracle.com/bigdata/k-means-clustering-machine-learning

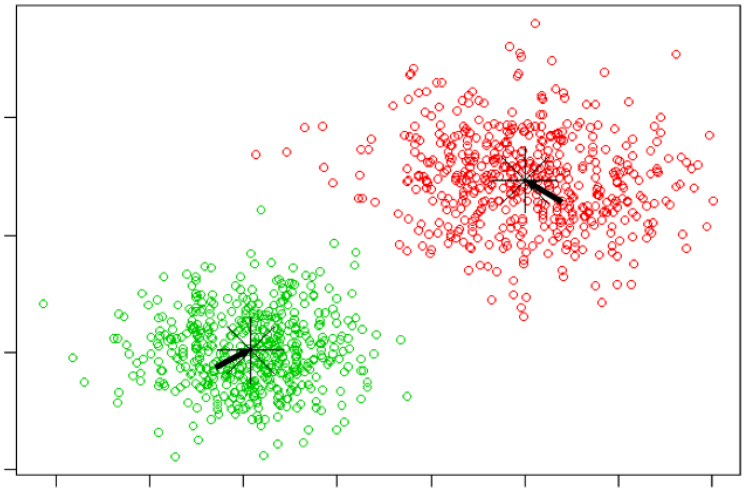

Figura 2.2: Processo de agrupamento para dois clusters [2]

O centróide de um cluster C_i pode ser definido pela média das distâncias dos pontos daquele cluster para o seu centro, onde a média de um objeto p para o centro c_i do cluster C_i é dada por $dist(p, c_i)$, em que $dist(x, y)$ é a distância Euclidiana entre os pontos x e y [11].

Além disso, é possível medir a taxa de erro de um cluster C_i, que consiste na soma do erro quadrático da distância entre os pontos que representam um objeto p no cluster C_i e seu centróide c_i:

$$\sum_{i=1}^{k} \sum_{p=C_i} dist(p, c_i)^2 \tag{2.14}$$

Por fim, o critério de inicialização utilizado é o *k-means++*, um método que busca minimizar a média quadrática da distância entre os pontos de um mesmo cluster, de tal forma que a convergência do algoritmo seja rápida [1].

Métrica de Avaliação

Dentre as métricas de avaliação disponíveis de clusterização, está a métrica de silhueta. A análise de silhueta proposta por *Peter J. Rousseeuw* é uma análise que mostra o quão bem os objetos estão dentro de cada cluster resultando na largura média da silhueta, que varia no intervalo $[-1, 1]$ [26]. Uma pontuação (largura média) de silhueta que está próxima de 1 indica que a amostra está longe dos clusters vizinhos, enquanto uma pontuação negativa indica que a amostra pode ter sido atribuída ao cluster errado.

Capítulo 3

Trabalhos Relacionados

3.1 Algoritmos de Aprendizagem de máquina

3.1.1 Prevendo decisões de compra em um jogo para celular

O estudo de usuários *Freemium* e *Premium* é um tema que está em constante amadurecimento por conta do seu impacto nos negócios atuais e não fica preso ao contexto de apps para empresas. Assim sendo, o artigo *Predicting Purchase Decisions in Mobile Free-to-Play Games* [29] motivado pelo fato de que prever características dos jogadores é uma técnica valiosa para cortar gastos e aumentar receita, propõe a utilização de dois modelos de aprendizagem de máquina.

O modelo de classificação consiste em classificar se um usuário irá realizar uma compra de um item dentro do jogo no futuro ou não para um conjunto de dados desbalanceado, no qual a maior parte é constituída por jogadores não pagantes. Para lidar com o desbalanceamento do conjunto de dados, a técnica SMOTE-NC (do inglês, *Syntethic Minority Over-sampling Technique-Nominal Continuous)* é adotada, gerando dados sínteticos para popular os dados considerados minorias no conjunto de dados analisado em questão.

Em seguida, para a classificação foram levados em consideração 3 comportamentos: acessos à plataforma (*logins*), rodadas jogadas e compras realizadas dentro do app. Além disso, foi levado em consideração três períodos de tempo de utilização: 1, 3 e 7 dias.

Adicionalmente, os algoritmos de classificação utilizados foram Árvore de Decisão, Florestas Aleatórias e Máquina de Vetores de Suporte (SVM). Assim, o algoritmo de máquina de vetores de suporte foi melhor avaliado em relação aos outros modelos quando utilizadas as métricas de sensibilidade (do inglês, *recall*), precisão e a pontuação F2 (do inglês, *F2-Score*).

Por fim, uma análise de importância dos atributos em relação ao classificador de Florestas Aleatórias foi induzida colocando como atributos mais relevantes o número de compras realizadas pelo usuário e a quantidade de dinheiro já gasto na plataforma.

Como conclusão, a pesquisa afirma que os modelos propostos geraram abordagens novas para a utilização do jogo do ponto de vista de experiência, além de evidenciar novos segmentos de marketing entre os usuários estudados. Sendo assim, o artigo em questão influencia este trabalho em dois pontos, um acerca da metodologia, o qual o autor utiliza o algoritmo de Florestas Randômicas e outro acerca da análise dos resultados, com a

análise da importância de atributos que pode trazer abordagens ainda mais interessantes do ponto de vista de negócio.

3.1.2 Prevendo rotatividade de clientes utilizando árvores de decisão em um sistema de serviço de celular

Um estudo análogo ao apresentado nesta pesquisa é dado pela previsão de rotatividade de usuários ou clientes em um determinado serviço. O estudo é análogo por tratar-se de uma análise de ações futuras de usuários ou clientes baseada em ações do passado.

Em [3], é proposto um modelo de previsão de rotatividade de clientes que utiliza o algoritmo de árvores de decisão, uma vez que é ressaltado que esse algoritmo possui um retrospecto positivo no que diz respeito à esse tipo de previsão.

Assim, o trabalho apresenta 3 atributos que são utilizados como entrada para o modelo: *frequência de uso*, que diz respeito ao número de ligações feitas pelo cliente, *esfera de influência*, que diz respeito ao número de ligações distintas feitas pelo cliente e, por fim, *minutos de uso*.

Uma análise em relação ao tempo é implementada para a robustez do modelo. A janela inicial observada corresponde à 180 dias, mas cortes de 10, 20, 30 e 60 dias são observadas. Além disso, é acrescentada uma penalidade ao modelo para os clientes previstos de maneira errada.

As métricas utilizadas para avaliação da performance do modelo são as métricas de sensibilidade (do inglês, *recall*), precisão e pontuação-F (do inglês, *F-score*).

Após a execução do modelo, o estudo conclui o valor das métricas de sensibilidade, precisão, pontuação-F são 0.95, 0.82 e 0.88, respectivamente, sendo o sub-período de 10 dias o mais performático. Os autores também ressaltam o fato de que o estudo é limitado devido à quantidade de dados disponíveis, mas que os resultados foram satisfatórios.

Apesar de ser realizado em um domínio diferente em relação à presente pesquisa, alguns pontos presentes nesse artigo puderam ser aproveitados nesse trabalho. A utilização do algoritmo de Árvore de Decisão no modelo é um dos pontos integrados, além da utilização das métricas de precisão, sensibilidade e *F-Score*.

3.2 Engenharia de atributos

3.2.1 Estudo de transição de usuários de um modelo Freemium para Premium

O artigo *Estudo de transição de usuários de um modelo Freemium para Premium* [23], que é utilizado como base para a presente pesquisa, buscar entender quais comportamentos dos usuários correspondem ou indicam uma transição do modelo *Freemium* para *Premium*, que são representados por dois aplicativos diferentes *"Sandbox"* e *"Real"*, respectivamente, por meio de um modelo de classificação supervisionada. Vale a pena ressaltar que os autores da pesquisa fizeram uma parceria com uma empresa de investimentos que detêm os aplicativos citados, os quais são os objetos de estudo. O artigo *Estudo de transição de usuários de um modelo Freemium para Premium* [23], que é utilizado como base para a presente pesquisa, buscar entender quais comportamentos dos usuários correspondem ou

indicam uma transição do modelo *Freemium* para *Premium*, que são representados por dois aplicativos diferentes *"Sandbox"* e *"Real"*, respectivamente, por meio de um modelo de classificação supervisionada. Vale a pena ressaltar que os autores da pesquisa fizeram uma parceria com uma empresa de investimentos que detêm os aplicativos citados, os quais são os objetos de estudo.

Os dados utilizados para o estudo foram recuperados do banco de dados em formato *csv* e possuem a estrutura de eventos, assim como mostrado no Capítulo 2. Ao total, são 208 tipos de eventos capturados, incluindo eventos triviais como os disparados ao abrir o aplicativo e fechá-lo.

Individualmente, cada evento não possui um significado relevante para o processo de aprendizagem de máquina e, sendo assim, é necessário trabalhá-los para que seja possível utilizá-los como entrada para o modelo. Portanto, os eventos foram transformados em um matriz na qual cada linha corresponde ao usuário e as colunas representam dois tipos de atributos em relação aos eventos: atributos de engajamento de usuário e atributos de eventos.

Os atributos de engajamento de usuário correspondem a atributos como o total de número de eventos, número de dias ativos na plataforma, média de eventos por dia ativo, etc. Ao passo que atributos de eventos correspondem a atributos como o número de dias entre a ocorrência de dois eventos específicos, dias entre a última e penúltima ocorrência do evento, etc. A pesquisa realiza a etapa de engenharia de atributos utilizando o próprio banco de dados da aplicação, resultando em aproximadamente 1000 atributos. Os atributos mais relevantes foram escolhidos a mão, resultando em 382 atributos.

Após o processo de engenharia de atributos, um agrupamento é executado sobre o conjunto de dados para melhor compreendê-lo. Dada as diferentes escalas dos atributos, uma etapa de normalização e padronização dos valores foi adicionada antes da clusterização.

Como o conjunto de dados em questão é desbalanceado (cerca de 0.85% dos usuários correspondem à usuários *Premium*), os autores da pesquisa propõem a estratificação do conjunto de dados em relação aos conjuntos de treino e teste na proporção de 90:10 e 50:50. Sendo assim, para cada um dos dois cenários propostos, quatro algoritmos são utilizados no modelo supervisionado: Árvore de Decisão, Floresta Aleatória, *Gradient Boosting Machine* (GBM) e *Support Vector Machines* (SVM), tendo como métricas de performance: Sensibilidade, Precisão, *F-Score* e AUC (do ingles, *Area Under the Receiver Operating Characteristics*).

Para fins de conclusão, a média das métricas é calculada para todos os cenários propostos e, assim, o algoritmo de Árvore de Decisão C5.0 foi o algoritmo com melhores números em relação à métrica de sensibilidade e pontuação F1, que, de acordo com [15], evidenciou-se como uma métrica adequada para problemas de classificação binários ou multiclasses.

O artigo descrito nessa seção foi utilizado como base para o estudo da presente pesquisa. Como maior influencia, tem-se o processo de engenharia de atributos acerca dos eventos utilizados como entrada para o modelo. Além disso, a etapa de clusterização, com o intuito de compreender ainda mais os usuários, foi inserida na etapa de aplicação dos algoritmos de aprendizagem de máquina, assim como explicado no Capítulo 4.

Por fim, detalhes presentes na etapa de estratificação presente neste artigo foram utilizadas como base para a presente pesquisa, como a razão da divisão entre dados de treino e teste e utilização da técnica de validação cruzada para evitar sobre-ajuste.

3.3 Clusterização

3.3.1 Intepretação de Clusters de Usuários em Rotatividade de Usuários

A clusterização de usuários para compreender seus comportamentos é o objeto de estudo de uma pesquisa realizada na empresa *Snap Inc*, anteriormente conhecida como *Snapchat Inc.*, que foca exclusivamente na análise de retenção do usuário do aplicativo *Snapchat* [32], que é um aplicativo multimídia de mensagens.

Foram coletados dados de 0.5 milhões de novos usuários cadastrados durante as duas primeiras semanas do mês de agosto de 2017, além de recuperar os dados de 40 milhões de usuários já cadastrados na plataforma. Os dez atributos gerados a partir do conjunto de dados observado são baseados nas principais atividades do usuário dentro do aplicativo.

Assim, a pesquisa objetiva gerar clusters interpretáveis para os novos usuários cadastrados baseados em seus comportamentos iniciais e a evolução dos padrões de comportamento quando o usuário interage com as funcionalidades dentro do aplicativo e com outros usuários. Vale ressaltar que a geração de grupos interpretáveis entre os usuários é crucial para o entendimento do comportamento deles, além de habilitar o design do aplicativo em questão, onde será possível tomar diferentes ações de acordo com os diferentes tipos de usuários.

O método utilizado no processo de clusterização foi dividido em quatro etapas, uma vez que para todas as etapas foi utilizado o algoritmo *K-means* e realizada uma análise de silhueta para decidir automaticamente o número K de clusters e distribuir os dados. A primeira etapa consistiu em realizar a clusterização para cada atributo separadamente, assim foi possível identificar usuários de acordo com cada atributo permitindo clusters interpretáveis. Em seguida, os atributos foram organizados de forma a combiná-los entre si mantendo sua interpretabilidade.

Vale a pena ressaltar que a pesquisa define que um usuário parou de utilizar o aplicativo quando não há quaisquer atividades no aplicativo na segunda semana após seu primeiro registro na plataforma. Sendo assim, a clusterização permitiu concluir seis perfis diferentes de usuários de acordo com a probabilidade de parar de utilizar a plataforma, como *All-stars*, usuário que utilizam a plataforma de maneira assídua e *Invitees*, usuários que são convidados para o aplicativo pelos amigos, mas que possuem uma chance alta de pararem de utilizar se não interagirem com eles. Dito isso, confirma-se uma das hipóteses iniciais do estudo, que auxiliaria em alavancar percepções valiosas acerca da retenção de usuários.

A avaliação da qualidade das divisões dos números de clusters do algoritmo *K-means* por meio da métrica de silhueta faz com que o número K de clusters seja definido de uma forma menos abstrata. Assim, essa abordagem foi utilizada como forma de avaliação do processo de clusterização na presente pesquisa.

3.3.2 Medindo contribuição de variáveis para clusterização

Para medir a contribuição de cada variável a fim de explicitar quais variáveis são mais relevantes para o processo de clusterização Ismaili *et al.* utilizam uma abordagem de aprendizagem de máquina supervisionada para propor uma forma de realizar essa avaliação [14].

A ideia da pesquisa é organizar as variáveis de acordo com suas contribuições para o processo de clusterização. A importância da variável é avaliada como sua capacidade de prever a associação de uma instância a um cluster específico. Para compreender essa importância, a ideia principal consiste em tornar os rótulos derivados do processo de clusterização em rótulos a serem previstos no modelo de aprendizagem supervisionado. Sendo assim, para cada variável é utilizado um algoritmo de classificação para prever o rótulo gerado pelo cluster.

Para medir a importância de cada variável, dois métodos de avaliação são levados em consideração: acurácia e Índice de Rand Ajustado (do inglês, *Adjusted Rand Index* - ARI). Além disso, como *benchmark* para avaliação dessa metodologia, os índices de Davies-Bouldin e SD foram utilizados [17].

Como objeto de estudo, foram utilizados três conjuntos de dados do repostitório UCI [4]: WINE, PIMA e WAVEFORM. A etapa de pré-processamento dos dados consistiu na padronização destes, ao passo que o algoritmo de clusterização utilizado foi o *K-Means*, utilizando o número de rótulos a serem previstos como o número de clusters K. Por fim, o algoritmo utilizado para prever os rótulos dos clusters foi a árvore de decisões. Como conclusão do estudo, a forma de avaliação proposta demonstrou competitividade em relação aos índices postos em *benchmark*.

Apesar de ser aplicado em um escopo diferente do apresentado como proposta neste trabalho, o processo para medir a contribuição de variáveis para clusterização foi utlizada como base para a etapa de aplicação dos algoritmos de aprendizagem de máquina supervisionada na presente pesquisa, além de implicar na etapa de análise de importância de atributos.

Capítulo 4

Visão Geral do Modelo

O modelo desenvolvido na presente pesquisa consistiu em analisar os eventos disparados pelos usuários durante a utilização do aplicativo e, com eles, agrupar os usuários de acordo com comportamentos semelhantes. Em seguida, os rótulos gerados para cada agrupamento são utilizados como os rótulos a serem previstos pelos algoritmos de aprendizagem de máquina como Árvore de Decisão, *Gradient Boosting Machine* e Florestas Aleatórias. Por fim, os agrupamentos são analisados a fim de encontrar a maior concentração de usuários Premium tendo como produto da pesquisa os identificadores desses usuários.

4.1 Metodologia

A pesquisa foi pautada na metodologia *Knowledge Discovery in Databases* (KDD), que tem como objetivo estabelecer um processo de identificar padrões potencialmente entendíveis, válidos e úteis nos dados [8]. Esse processo utiliza o banco de dados com uma etapa de pré-processamento ou sub-amostragem para aplicar métodos (algoritmos) de mineração de dados para enumerar os padrões contidos no conjunto de dados tão bem quanto avaliá-los. O processo KDD consiste em 9 etapas muito bem definidas, sendo elas

1. **Entendimento do Negócio**: Nessa etapa é necessário desenvolver um entendimento do domínio da aplicação, ou negócio, a fim de gerar o objetivo a partir da visão do cliente. A motivação e a contextualização do problema, tão bem quanto o negócio utilizado como caso de uso na presente pesquisa foram apresentados no Capítulo 1.
2. **Seleção dos dados**: Consiste em efetivamente selecionar e coletar os dados que serão trabalhados como entrada para o modelo.
3. **Pré-processamento dos Dados**: Realização de operações básicas em cima do conjunto de dados em estudo, como remoção de ruídos ou quaisquer informações que podem não ser úteis para o projeto. Também são levadas em consideração métodos utilizados para lidar com falta de dados.
4. **Redução/Transformação dos Dados**: é necessário encontrar atributos úteis para representar os dados, de tal forma que esses dependam do objetivo do projeto. Aqui

são usadas técnicas de transformação de dados para gerar novas representações para os dados.

5. **Identificação do Problema**: A etapa de identificação do problema consiste em concluir qual método de mineração de dados será utilizado na pesquisa, isto é, se é um problema de classificação, agrupamento, regressão, etc.
6. **Escolha dos Algoritmos de Mineração de Dados**: é necessário escolher os algoritmos de mineração de dados que serão utilizados no projeto. Dito isso, faz-se válido decidir também quais parâmetros serão utilizados como entradas para esses algoritmos. Os algoritmos utilizados na presente pesquisa foram Árvore de Decisão, Florestas Aleatórias e *Gradient Boosting Machine*, e já foram detalhados no Capítulo 2.
7. **Aplicação dos Algoritmos de Mineração de Dados**: aplicar os algoritmos de mineração de dados de forma a obter os padrões estabelecidos no início do projeto.
8. **Interpretação dos Resultados**: essa etapa avalia os padrões obtidos após a aplicação dos algoritmos de mineração de dados. Além disso, essa etapa também conta com visualizações dos padrões extraídos dos dados utilizados como entrada para o modelo.
9. **Consolidação do Trabalho**: A etapa de consolidação consiste em finalizar o projeto. Nessa etapa, o projeto é incorporado à um sistema já existente ou é documentado para futuras modificações. Essa etapa também inclui realizar uma análise crítica acerca das hipóteses iniciais.

A Seção 4.2 representa a etapa de seleção de dados, a Seção 4.3 engloba as etapas de pré-processamento e transformação dos dados e a Seção 4.3.5 representa a etapa de escolha dos algoritmos de mineração de dados do KDD. Por fim, as etapas de aplicação dos algoritmos, interpretação dos resultados e consolidação do trabalho são cobertas no Capítulo 5.

4.2 Seleção e Extração dos Dados

Os dados referentes aos eventos são providos pela ferramenta de análise da google *Google Analytics* integrado à plataforma *Firebase* [1]. Assim, com o auxílio da biblioteca do Firebase, é possível capturar até 500 eventos personalizados em um aplicativo. Os dados capturados ficam disponíveis em um painel no *console do Firebase* [2]. Para realizar análises personalizadas e unir os dados presentes no console, é necessário integrar o *Firebase* a outra ferramenta denominada *Big Query* [3]. Essa ferramenta permite com que sejam realizadas consultas mais complexas aos dados de tal forma que seja possível extrair os eventos contidos no console para arquivos *csv*.

A ferramenta *BigQuery* pode receber consultas em linguagem SQL, semelhante às utilizadas em bancos de dados estruturados. Vale a pena ressaltar que os eventos são

[1] https://firebase.google.com/docs/analytics
[2] https://console.firebase.google.com/
[3] https://cloud.google.com/bigquery

salvos em tabelas diferentes e únicas para cada dia. Sendo assim, para realizar a coleta dos eventos, foi necessário escrever a seguinte consulta:

```
1     SELECT
2         event_date,
3         event_timestamp,
4         event_name,
5         user_id,
6         device.operating_system,
7     FROM
8         `Aplicativo-Social.events_*`
9     WHERE
10        event_date between 'Data-1' and 'Data-2';
```

Resgatamos então a data do evento, seu nome, o identificador do usuário que disparou o evento em questão e qual o sistema operacional do seu celular, assim como mostram as linhas 2, 3, 4 e 5, respectivamente, do código acima exposto. O caracter especial " * " é utilizado para recuperar os eventos de todas as tabelas, dado que, como citado anteriormente, existe uma tabela para cada dia. Além disso, a ferramenta possui uma limitação a qual os dados são exportados em arquivos *csv* com no máximo 1GB cada e, por isso, precisam ser concatenados posteriormente. Vale a pena ressaltar que o período correspondente à análise presente nessa pesquisa equivale à 9 meses (Julho/2020 - Março/2021).

4.3 Pré-processamento dos dados

Para maior organização do fluxo de pré-processamento dos dados, essa etapa foi subdivida em 3 passos: (i) concatenação dos dados, (ii) limpeza dos dados e (iii) montagem do conjunto de dados.

4.3.1 Concatenação dos dados

Como anteriormente citado, os dados em formato *csv* exportados pela ferramenta *BigQuery* ultrapassam a limitação de tamanho de 1 GB pré-estabelecida pela própria ferramenta . Assim sendo, é necessário realizar a concatenação desses, como mostra a Figura 4.1. Assim, os arquivos *csv* são lidos e transformados em estruturas chamadas *Data Frames* [4], que nada mais são que tabelas. Os *Data Frames* são então concatenados respeitando as datas de disparo dos eventos para, enfim, serem salvos em um arquivo *csv* final.

4.3.2 Limpeza dos dados

O processo de limpeza de dados consiste em retirar quaisquer dados inconsistentes, como dados em branco ou nulos. Além disso, faz-se interessante nessa etapa transformar os tipos dos dados que iremos trabalhar, já que não temos controle do formato dos dados

[4]https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

[5]Imagem acessada em Dezembro de 2020: https://pandas.pydata.org/pandas-docs/stable/user$_guide/merging.html$

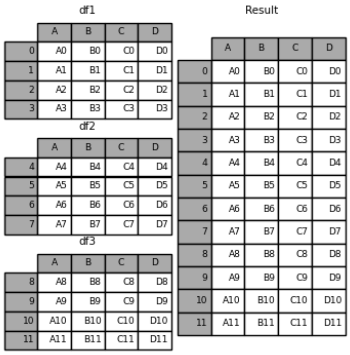

Figura 4.1: Concatenação de datasets [5]

ao exportá-los da ferramenta *BigQuery*. Posto isso, vale a pena citar o formato inicial dos dados, assim como mostra a Figura 4.2. O *Data Frame* conta com 5 colunas, que possuem seus valores deletalhados na tabela abaixo, uma vez que o tipo *Object* é dado para colunas que consistem em tipos misturados ou formadas puramente por uma sequência de caracteres:

event_ date	Object
event_ timestamp	Integer
event_ name	Object
user_ id	Float
operating_ system	Object

	event_date	event_timestamp	event_name	user_id	operating_system
0	20201212	1607743349731004	ProfileClickedProfile	150096.0	ANDROID
1	20201212	1607787781796000	ProfileClickedRedirectFromBonus	328625.0	ANDROID
2	20201212	1607746145914007	donation_tooltipProfile_view	229274.0	ANDROID
3	20201212	1607746186911000	ProfileClickedProfile	263811.0	ANDROID
4	20201212	1607790348321000	ProfileClickedRedirectFromBonus	141556.0	ANDROID

Figura 4.2: Formato inicial dos dados

Inicialmente são contabilizados as linhas que possuem a coluna *user_ id* nula, vazia ou composta de '0'. Assim, atribuindo o valor 1 para as linhas que possuem *user_ id* e 0 para as linhas que não possuem, é possível identificar no gráfico da Figura 4.3 um total de 3.829.934 linhas com valor de *user_ id* vazio ou nulo.

Em seguida, a próxima coluna a ser trabalhada é a referente ao sistema operacional: *operating_ system*. Ao identificar as linhas desta coluna, são observados três valores: "ANDROID", "IOS" e "WEB" com aproximadamente 50 milhões, 10 milhões e 3 milhões de ocorrências, respectivamente, de acordo com o gráfico apresentado na Figura 4.4. Por motivos de negócio, apenas as linhas com valor "ANDROID" serão consideradas na análise desta pesquisa. Assim, os valores nulos, branco, "IOS" e "WEB" são removidos do conjunto de dados. Vale a pena ressaltar que os valores dessa coluna não são únicos, isto é, dentre as 50 milhões de ocorrências estão eventos repetidos e que, por muitas vezes, foram disparados pelo mesmo usuário.

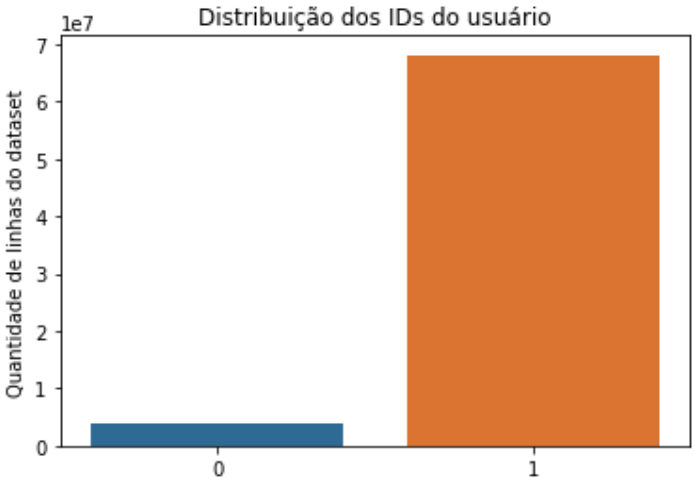

Figura 4.3: Distribuição dos IDs no conjunto de dados

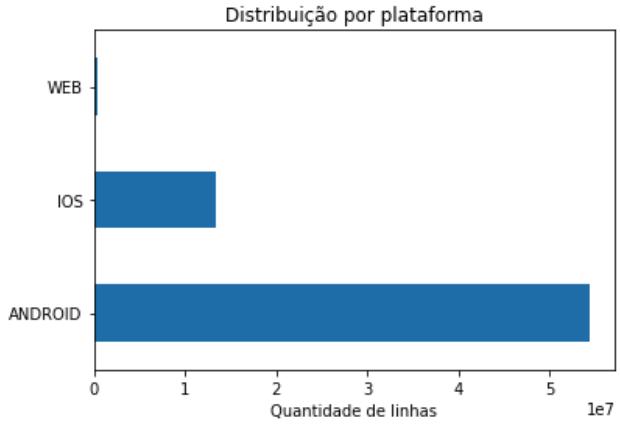

Figura 4.4: Distribuição das plataformas dos usuários no conjunto de dados

A partir da limpeza dos dados relacionados aos IDs e a plataforma dos usuários, é possível identificar que o número de usuário únicos presentes no conjunto de dados corresponde a um universo de mais de 50 mil usuários.

Por fim, são removidos os valores nulos e brancos da coluna *event_ name*. Além disso, são contabilizados quantos eventos distintos a coluna possui, totalizando 173 eventos. Após a limpeza, os dados são salvos em um arquivo *csv* separado.

4.3.3 Montagem dos dados

A montagem dos dados consiste, primeiramente, em converter o tipo da coluna *event_ date* de *object*, cuja definição foi citada na subseção de limpeza de dados, para *datetime*, por meio de uma função auxiliar da biblioteca *pandas*. Em seguida, é necessário formatar o conjunto de dados para que seja possível realizar o processo de engenharia de atributos detalhado na próxima seção, de tal forma que o conjunto seja dividido em dois conjuntos diferentes conforme mostrado na Figura 4.5.

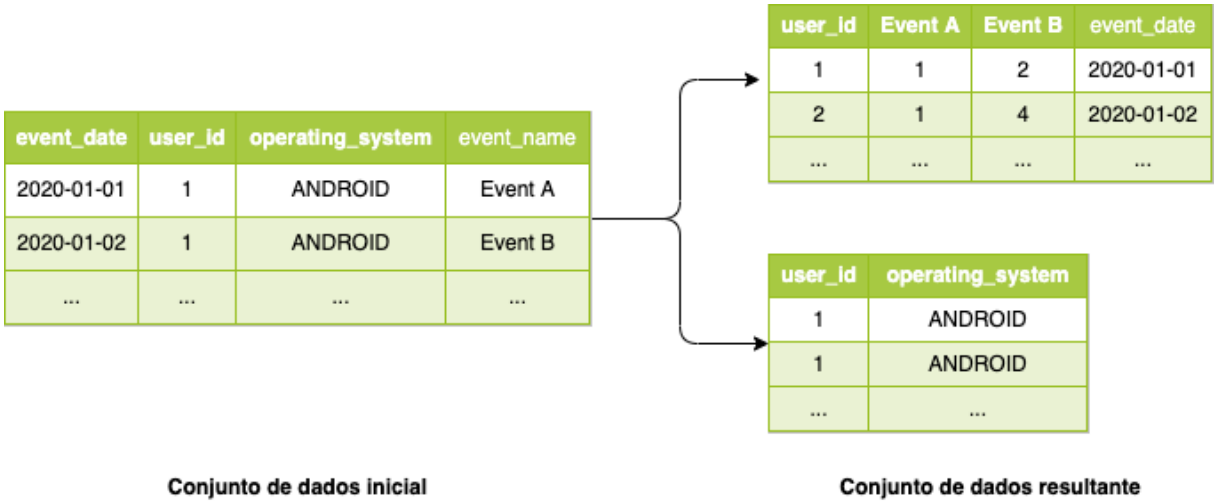

Figura 4.5: Conjunto de dados pré-processado

Como é possível observar, o primeiro subconjunto resultante consiste nos eventos separados por usuário por data em que foram disparados enquanto que o segundo subconjunto conta com os identificadores dos usuários e seus sistemas operacionais e será utilizado como base para montar o conjunto de dados final na etapa de transformação dos dados.

4.3.4 Transformação dos Dados

Os dados limpos permitem-nos seguir para a etapa de transformação dos dados, a qual possui como objetivo preparar os dados para que sejam utilizados como entrada para o modelo. Aqui aplica-se a engenharia de atributos, do inglês *Feature Engineering*, que é a tarefa de melhorar a performance de um modelo preditivo em um conjunto de dados por meio da transformação dos seus dados iniciais [22]. Esse processo é fundamental para que sejam geradas informações relevantes sobre o uso do aplicativo a partir do usuário, dado que os atributos utilizados em um projeto de mineração de dados são o fator mais importante para determinar o sucesso ou falha desse [7]. Portanto, combinando a metodologia para geração de atributos utilizada no artigo [23] com o conhecimento de domínio de profissionais envolvidos no negócio, 9 atributos para cada evento foram gerados para o conjunto de dados:

- **Total de Dias Ativos** : o número total de dias que o usuário tem ficado ativo no *app*.
- **Total de Dias Calendário** : o número total de dias passados desde que o usuário foi visto pela primeira e última vez no *app*.
- **Meia-vida dos Dias Ativos do Usuário** : O número de dias calendário passados até atingir metade dos dias ativos.
- **Total de Ocorrências do Evento** : o número total de ocorrências de um evento específico por usuário.

- **Meia-vida do Evento** : o número de dias levados até o usuário atingir a quantidade equivalente à metade dos eventos disparados. Por exemplo, se o usuário disparou um evento específico 10 vezes, o tempo passado desde o primeiro disparo até o quinto disparo representa a meia-vida desse evento.
- **Média de Eventos por Dias Ativos** : Total de ocorrências de um evento específico dividido pelo número total de dias ativos do usuário no *app*.
- **Média de Eventos por Dias Calendário** : Total de ocorrências de um evento específico dividido pelo número total de dias calendário do usuário no *app*.
- **Média de dias entre as ocorrências prévias de um evento** : Para cada evento, o número de dias entre a última vez que o evento foi disparado e a observação atual. A soma desses tempos é divida pelo número de vezes que esse evento foi disparado.
- **Dias entre a última e a penúltima ocorrência do evento** : É calculado o número de dias passados entre a última e a penúltima ocorrência para cada evento.

Assim, com os atributos já definidos, foram escolhidos 10 eventos levando em consideração quais eventos são mais relevantes para o objetivo do negócio, que é a compra de uma assinatura no aplicativo. Para cada evento tem-se 6 atributos, totalizando 60 atributos que, somados aos atributos dos usuários, isto é, dias calendário, dias ativos e meia-vida de dias ativos, totalizam 63 atributos. Por fim, o conjunto de dados final conta com 56235 instâncias.

4.3.5 Aplicação dos Algoritmos de Mineração de Dados

Na etapa de número 7 da metodologia utilizada na presente pesquisa está a aplicação dos algoritmos de mineração de dados. Assim, são usadas técnicas para implementar e aperfeiçoar os algoritmos escolhidos. Os experimentos foram executados por meio da linguagem *Python*[6] e das bibliotecas *Pandas*[7], utilizada na parte de exploração de dados, tão bem quanto suas análises, *Numpy*[8], uma biblioteca utilizada para trabalhar com funções matemáticas, *Seaborn*[9] para trabalhar com as visualizações e, por fim, *Scikit-learn*[10], que implementa diversos algoritmos de aprendizagem de máquina. Vale a pena ressaltar que as ferramentas escolhidas foram baseadas na aceitação dessas na comunidade de desenvolvimento de software e ciência de dados.

Agrupamentos

Primeiramente os atributos gerados na etapa de transformação dos dados são selecionados e utilizados como entrada para o algoritmo de agrupamentos *K-Means*. Como os atributos escolhidos possuem escalas diferentes, é necessário normalizá-los para que eles possuam o mesmo intervalo.

[6] https://www.python.org/
[7] https://pandas.pydata.org/
[8] https://numpy.org/
[9] https://seaborn.pydata.org/
[10] https://scikit-learn.org/stable/

Assim, uma análise de silhueta para 3 a 6 clusters é executada sobre o conjunto de dados a fim de identificar qual o número de clusters ideal para o problema em questão. A partir da análise de silhueta gerada, o agrupamento é executado pela última vez, agora com o número de agrupamentos ideal e os rótulos gerados durante esse processo são armazenados para serem utilizados como as classes a serem previstas na etapa de execução dos algoritmos de aprendizagem de máquina.

Amostragem Estratificada

Estratificação de amostras é um método que leva em consideração a existência de grupos não homogêneos dentro de uma população e produz amostras, onde as proporções desses grupos são mantidas [27].

Posto isso e como ressaltado anteriormente, a classificação dos rótulos pode ser dividida entre duas etapas: treinamento e teste. Assim, o conjunto de dados é dividido de forma estratificada em dados para treinamento e teste na proporção de 70:30, respectivamente, isto é, 70% dos dados são utilizados para treinamento, enquanto 30% são utilizados para teste.

Validação Cruzada

Durante a etapa de treinamento, é possível que o classificador apresente um bom desempenho, mas, durante a etapa de teste, é possível que este mesmo classificador apresente um desempenho ruim. Isso acontece por conta de um fenômeno denominado Sobre-ajuste, do inglês *overfitting* e que consiste em, de uma forma geral, utilizar modelos que são muito complexos para a quantidade de dados disponível [20]. Nesse caso, o modelo trabalha muito bem com as características contidas nos dados de treinamento, mas não é possível ser generalizado para novos dados, como os contidos no conjunto de teste.

Dado esse contexto, para evitar o cenário de sobre-ajuste, pode-se utilizar uma técnica denominada validação cruzada. A validação cruzada é um método utilizado para avaliar a generalização de um modelo preditivo, em que o conjunto de dados é dividido em subconjuntos mutuamente exclusivos e de tamanhos iguais que serão utilizados como entrada para o modelo [2].

Um dos algoritmos mais utilizados para realizar a validação cruzada é o algoritmo *K-Fold*, que divide o conjunto de dados aleatoriamente em k subconjuntos de tamanho aproximado. Para cada subconjunto, são divididas partições de treinamento e apenas uma partição de teste para ser utilizada na previsão do modelo. O processo de particionamento é repetido k vezes, assim como mostra a Figura 4.6. Para a presente pesquisa, o número de subconjuntos escolhido foi 10 e foi baseado na metodologia aplicada em [23].

Ajustes em hiperparâmetros e *Grid Search*

Cada algoritmo possui um conjunto de atributos que podem ser alterados antes de realizar uma previsão acerca de um conjunto de dados específico. À esse conjunto, é dado o nome de hiperparâmetros do algoritmo [20]. Uma vez que a escolha dos hiperparâmetros corretos para modelo pode afetar significativamente sua performance [6], faz-se necessário realizar esse processo de maneira cautelosa. Para isso, foi utilizada a técnica denominada de *Grid Search*.

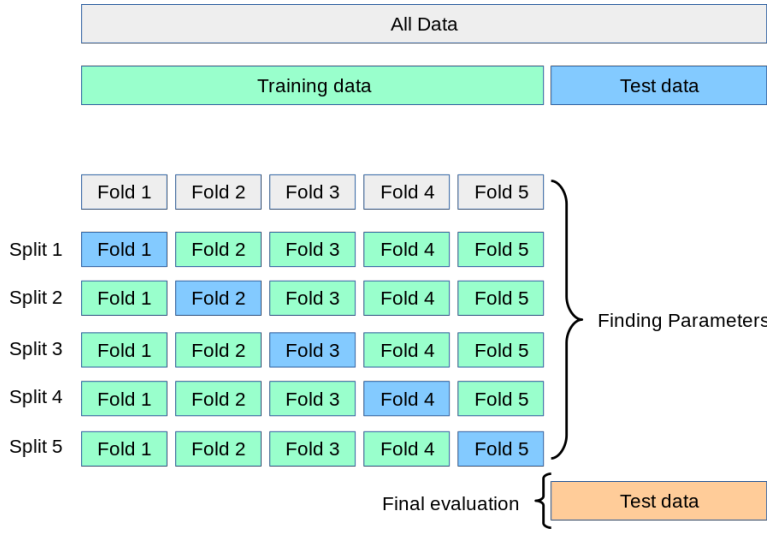

Figura 4.6: Validação Cruzada para dados de treino (*Training data*) e teste (*Test data*)

A técnica de *Grid Search* consiste em definir um conjunto de hiperparâmetros antes de realizar a previsão do modelo. Após a escolha desse conjunto, serão testadas todas as combinações possíveis entre os hiperparâmetros do conjunto de maneira exaustiva a fim de obter a melhor configuração para o modelo em questão. Os hiperparâmetros selecionados para os modelos utilizados na presente pesquisa encontram-se na Tabela 4.1 juntamente com seus valores e descrições. Vale a pena ressaltar que a etapa de Grid Search é aplicada sobre os dados do conjunto de teste.

Algoritmo	Hiperparâmetro	Descrição	Valores
Árvores de Decisão	- criterion - min_samples_split	- Função utilizada para medir a qualidade de divisão dos nós - Número mínimo de instâncias necessárias para realizar a divisão no nó interno	- 'gini', 'entropy' - 200, 300, 400, 500
Random Forests (Florestas Aleatórias)	- criterion - n_estimators	- Função utilizada para medir a qualidade de divisão dos nós - Número de árvores geradas na floresta	- 'gini', 'entropy' - 200, 300, 400, 500
Gradient Boosting Machine	- criterion - n_estimators	- Função utilizada para medir a qualidade da divisão dos nós - Número de estágios gerados para a execução do algoritmo	- 'friedman_mse', 'mse' - 200, 300, 400, 500

Tabela 4.1: Hiperparâmetros utilizados para os classificadores

Previsão dos modelos

Após a aplicação do *Grid Search* e do procedimento de validação cruzada com o algoritmo *K-fold*, os rótulos gerados pela clusterização são utilizados como os rótulos a serem previstos pelo modelo, enquanto os atributos gerados na Seção 4.3.4 são utilizados como entrada para a previsão do modelo.

No próximo capítulo, apresentaremos os resultados e análise dos mesmos diante da aplicação da metodologia explicada neste capítulo.

Capítulo 5

Resultados e Análises

Seguindo a metodologia KDD apresentada no início do Capítulo 4, a etapa de análise e interpretação dos resultados é dada pela etapa 8 da metodologia apresentada na Seção 4.1. Inicialmente são analisados os resultados da etapa de clusterização. Em seguida, são relembradas as métricas utilizadas para avaliação da performance dos modelos e apresentados os resultados juntamente com suas respectivas análises.

5.1 Clusterização

Assim como citado no Capítulo 4, a análise de silhueta foi utilizada para entender qual o melhor número de partições gerada pelo algoritmo de clusterização *K-Means* para o conjunto de dados em questão. Sendo assim, 5 valores distintos de K foram utilizados: 2, 3, 4, 5 e 6. Os valores foram anotados e descritos na Tabela 5.1.

Número de clusters	**Pontuação da Silhueta**
2	0.68403
3	0.51165
4	0.49210
5	0.51215
6	0.50557

Tabela 5.1: Pontuação da Silhueta para cada número de clusters

Sendo assim, analisando as pontuações apresentadas e levando em consideração que as pontuações são próximas, o número de clusters para uma escolha inicial é o 5, uma vez que quando K igual a 2, não é possível obter uma diversidade significativa para os dados. Com o número de clusters escolhido, é gerada uma análise gráfica acerca dos coeficientes da silhueta em relação aos rótulos gerados pelo algoritmo *K-Means* conforme mostra a Figura 5.1.

Nessa Figura, é possível observar os diferentes tamanhos dos clusters que, consequentemente, representa diferentes perfis de usuários de acordo com suas utilizações, como o cluster com rótulo 0, que indica usuários que não se engajaram tanto com a plataforma. Além disso, faz-se interessante identificar quantas instâncias pertecem a quais clusters, o que pode ser observado nos dados da Tabela 5.2. Com o número de instâncias de cada

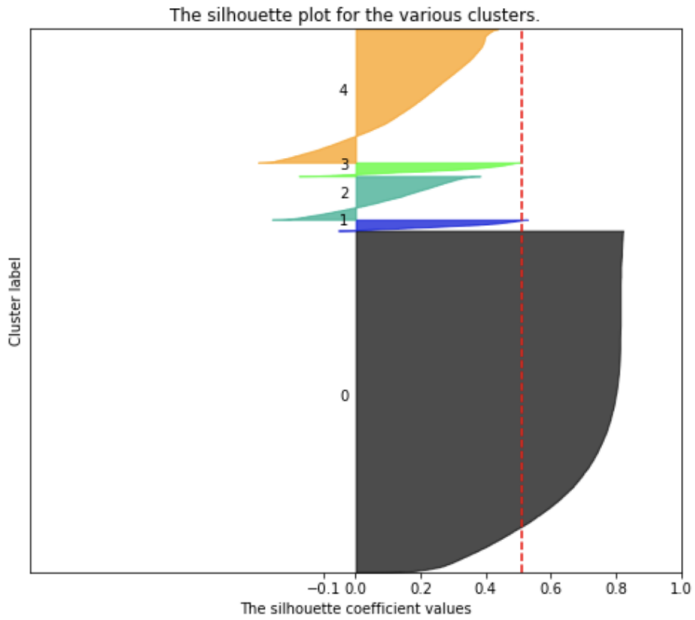

Figura 5.1: Gráfico da Silhueta para K = 5

cluster, é possível identificar o cluster que contém mais usuários assinantes e, assim, trabalhar de forma segmentada, uma vez que tem-se os identificadores dos usuários de cada cluster.

Rótulo do Cluster	**Quantidade de Instâncias**
0	41525
1	5864
2	2817
3	438
4	5591

Tabela 5.2: Número de instâncias por Cluster

5.2 Métricas de Desempenho

Assim que os experimentos foram executados, cada algoritmo foi analisado de acordo com as métricas descritas na Seção 2.2.1 no Capítulo 2. Para fins experimentais, cada algoritmo de previsão foi executado 10 vezes e o resultado final da métrica corresponde à média aritmética dos valores coletados.

Os resultados são apresentados na Tabela 5.3 e permitem concluir que o algoritmo que melhor desempenhou a identificação dos rótulos gerados pelo algoritmo de clusterização foi o algoritmo *Gradient Boosting Machine* (GBM) com as três métricas selecionadas tendo o maior resultado, sendo seu F1-Score equivalente a 0.98812. Vale ressaltar também o desempenho do algoritmo de florestas aleatórias que alcançou um F1-Score de 0.97923, o que foi próximo do resultado alcançado pelo GBM. Por fim, tem-se o algoritmo mais

simples dos selecionados, o algoritmo de Árvore de Decisão, que apresentou o F1-Score mais baixo dos analisados, somando 0.89293.

Algoritmo	**Sensibilidade (Recall)**	**Precisão**	**F1-Score**
Florestas Aleatórias	0.97936	0.97927	0.97923
Árvore de Decisão	0.89478	0.89406	0.89293
Gradient Boosting Machine	0.98815	0.98813	0.98812

Tabela 5.3: Métricas de Performance para os algoritmos selecionados.

Para entender mais a fundo quantas classes foram previstas de maneira correta e incorreta, utiliza-se a técnica de matriz de confusão.

Sendo assim, utilizando como base o algoritmo que melhor desempenhou entre os observados, *Gradient Boosting Machine*, é gerada uma matriz de confusão com o auxílio da biblioteca de visualizações de dados *Seaborn* [1], assim como mostra a Figura 5.2, onde o eixo y representa os rótulos previstos e o eixo x representa os rótulos reais. A soma da linha diagonal da matriz equivale ao total de instâncias do conjunto de dados de teste, que representa uma parte do conjunto de dados total, uma vez que esse foi dividido entre conjunto de treino e teste, assim como mencionado no Capítulo 4.

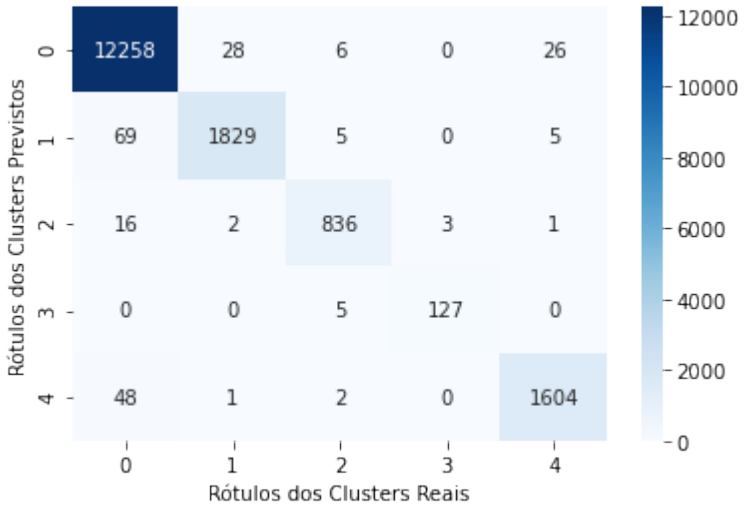

Figura 5.2: Matriz de Confusão referente ao algoritmo Gradient Boosting Machine

Pela matriz de confusão acima, é possível identificar que 69 instâncias que pertencem ao cluster 0 foram previstas como cluster 1, ao passo que 48 instâncias pertencentes ao cluster 0 foram previstas como cluster 4. Para o cluster 2, por exemplo, apenas 18 instâncias foram previstas com rótulos referentes à outros clusters. Dessa forma, o resultado da matriz de confusão mostra-se alinhado com o tamanho dos clusters, uma vez que o cluster 0, por ser o maior cluster, tende a conter mais previsões equivocadas.

[1] https://seaborn.pydata.org/

5.3 Análise de Importância de Atributos

A análise de importância de atributos consiste em identificar o quão importante cada atributo é em relação às decisões tomadas pelas árvores de decisão [20], que é utilizado como base tanto para o algoritmo de Florestas Aleatórias quanto para o algoritmo GBM.

Sendo assim, é possível realizar a análise de importância de atributos de algoritmos derivados de Árvores de Decisão a partir do atributo *feature_ importances_* presente nos algoritmos fornecidos pela biblioteca *sklearn*. O atributo em questão retorna um vetor com todos os atributos utilizados como entrada para o modelo com resultados entre 0 e 1, sendo 0 um atributo que não foi utilizado e 1 um atributo que preve perfeitamente o alvo, que neste caso é rótulo do cluster.

Para a presente análise, são gerados dois gráficos, um para o algoritmo que melhor desempenhou (GBM) e outro para o algoritmo de Florestas Aleatórias. Os gráficos contêm os 5 atributos que mais influenciaram na previsão dos rótulos do cluster.

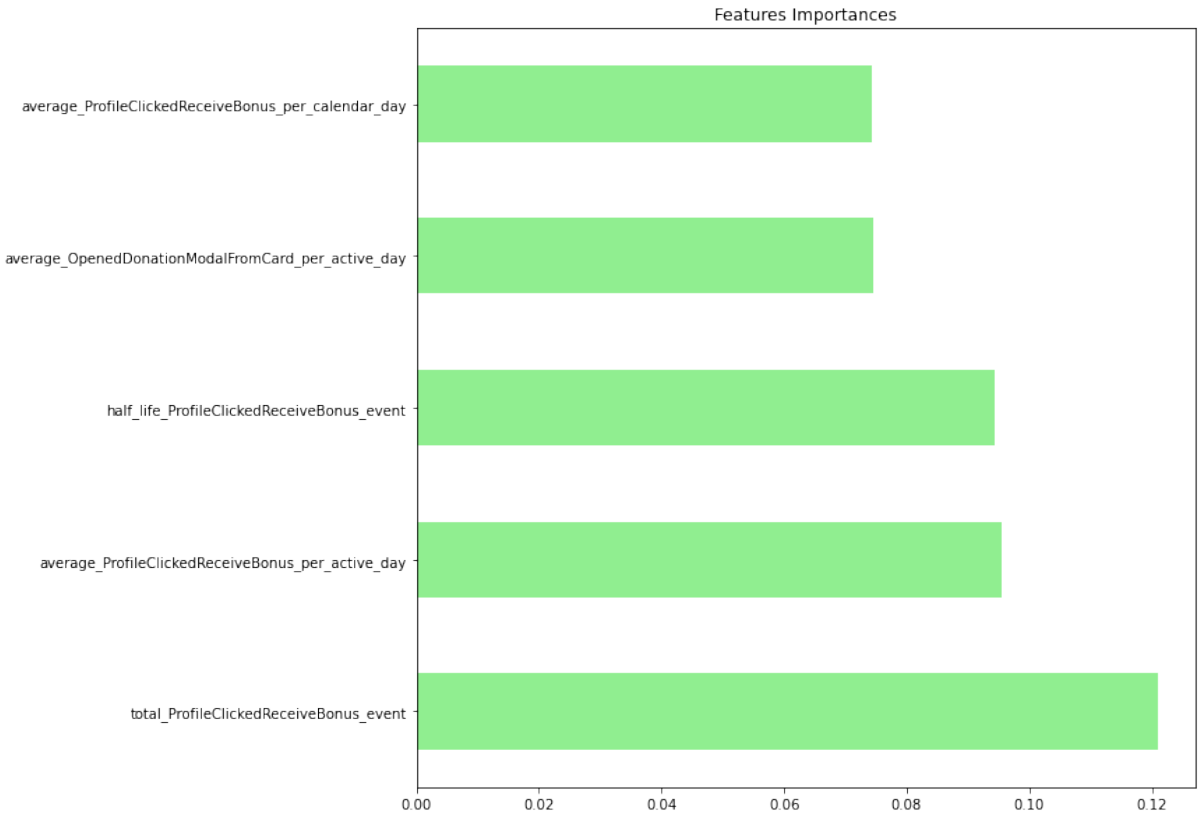

Figura 5.3: Gráfico de Importância de Atributos do algoritmo Gradient Boosting Machine

De acordo com a Figura 5.3, é possível concluir que os eventos "ProfileClickedReceive-Bonus"e "OpenedDonationModalFromCard"foram os mais relevantes para a classificação em questão, sendo o primeiro evento mais relevante com 4 variações: Meia-vida, média de disparos por dias ativos, média de disparos por dias calendário e total de disparos.

Assim, executando a mesma análise para o algoritmo de Florestas Aleatórias tem-se o gráfico da Figura 5.4.

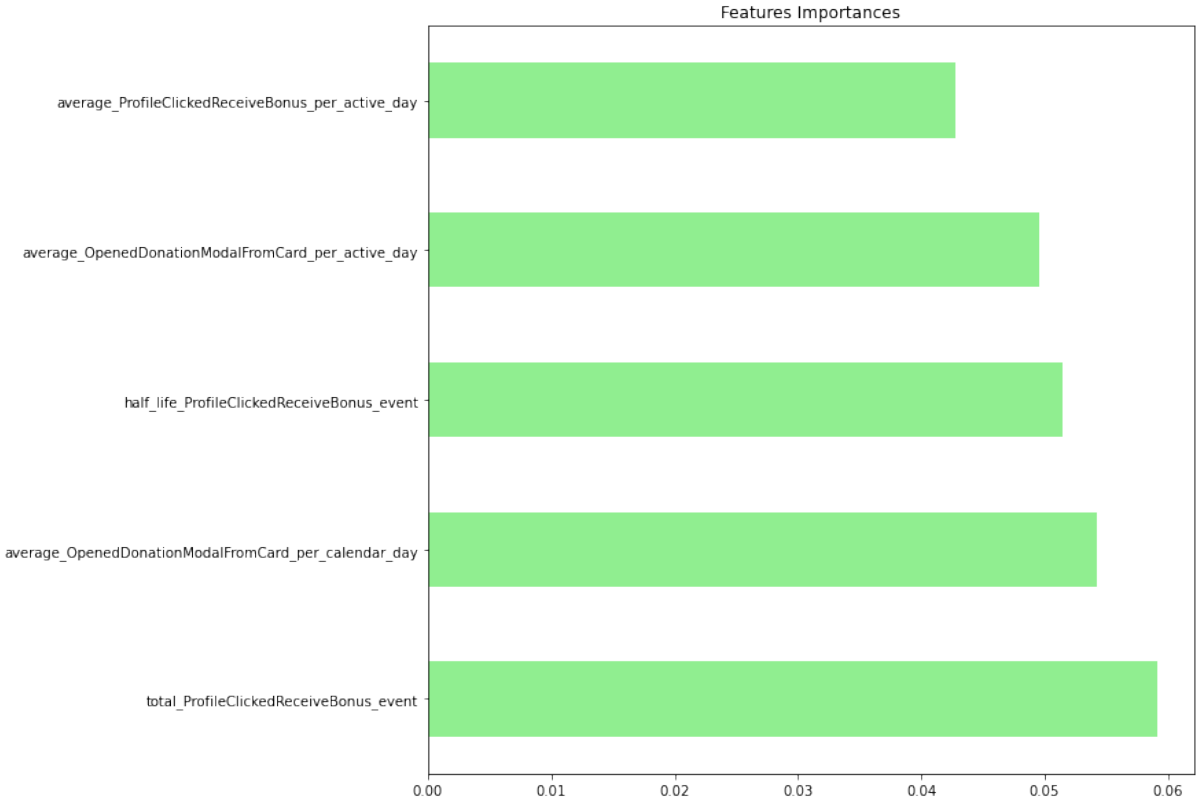

Figura 5.4: Gráfico de Importância de Atributos do algoritmo Florestas Aleatórias

Assim, é possível concluir que os mesmos eventos que figuraram na lista dos 5 atributos mais importantes do algoritmo *Gradient Boosting Machine* também foram relevantes para o algoritmo de Florestas Aleatórias. Contudo, para o algoritmo em questão, os atributos apresentaram valores inferiores. Por exemplo, o atributo mais relevante "total_ProfileClickedReceiveBonus_event"obteve pontuação de 0.05927, enquanto para o algoritmo GBM, o atributo mais relevante "total_ProfileClickedReceiveBonus_event"obteve pontuação de 0.01210.

5.4 Análise de Atributos por Cluster

A fim de compreender ainda mais os perfis de cada agrupamento, dois dos três atributos mais relevantes foram separados para uma análise por agrupamento. Com o auxílio da função *describe* da biblioteca *pandas*, dados estatísticos descritivos foram gerados para cada cluster e, assim, diagramas de caixa foram construídos.

Abaixo, as Figuras 5.5 e 5.6 representam os diagramas de caixa para os atributos total_ProfileClickedReceiveBonus_event e average_OpenedDonationModalFromCard_per_active_day, respectivamente.

As instâncias que foram agrupadas no cluster de número 1 apresentaram o comportamento mais claro em relação ao atributo total_ProfileClickedReceiveBonus_event. Com uma média aproximada de 98 eventos totais disparados, os usuários que são instâncias

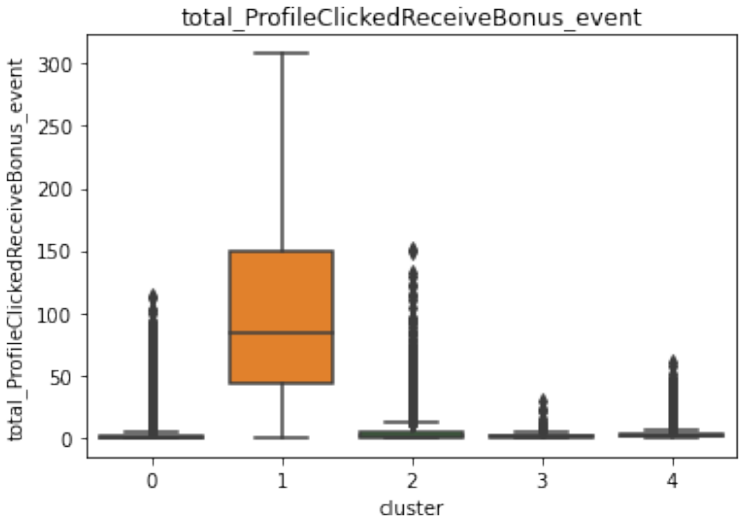

Figura 5.5: Análise de Diagrama de caixa para o evento total_ProfileClickedReceiveBonus_event

desse cluster, quando comparados aos clusters 0, 2, 3 e 4 possuem comportamento mais assíduo, uma vez que o evento em questão, por definição de negócio, tende a ser disparado diariamente. Sendo assim, os usuários desse cluster podem tender a realizar uma transição para o modelo *Premium* por conta do seu ritmo de utilização.

O próximo diagrama de caixa apresenta os usuários caracterizados como pertencentes ao cluster 4 com a maior média entre os demais clusters, além de conter o maior número de outliers em relação ao atributo em questão. Outro ponto importante a ser ressaltado é que os mínimos são comuns aos clusters 0, 2 e 3, mostrando que a média de disparos desse evento por dia ativo é próxima de 0. O cluster de número 1 também possui um comportamento notório em relação ao atributo analisado, uma vez que este contêm *outliers* mais próximos e a segunda maior mediana do gráfico.

Uma vez que o evento em questão (*OpenedDonationModalFromCard*) configura uma ação de acesso para a página de assinatura do modelo *Premium*, os clusters 1 e 4 podem ser trabalhados de forma isolada a fim de encontrar usuários com tendências à migrarem de serviço.

5.5 Análise de Usuários assinantes por cluster

Por fim, faz-se necessário para conclusão do estudo entender o cluster que contém mais assinantes para, assim, fornecer os IDs dos usuários desse (s) cluster (s) específico (s) de tal forma a realizar ações para que usuários com comportamentos parecidos sejam convertidos para o modelo *Premium*.

Sendo assim, utilizando o banco de dados da aplicação que foi usada como estudo de caso da presente pesquisa, foram recuperados todos os usuários que já fizeram alguma assinatura na plataforma até o dia 15 de março de 2021, tornando-se *Premium*. Ao total, 1241 usuários já realizaram algum tipo de assinatura.

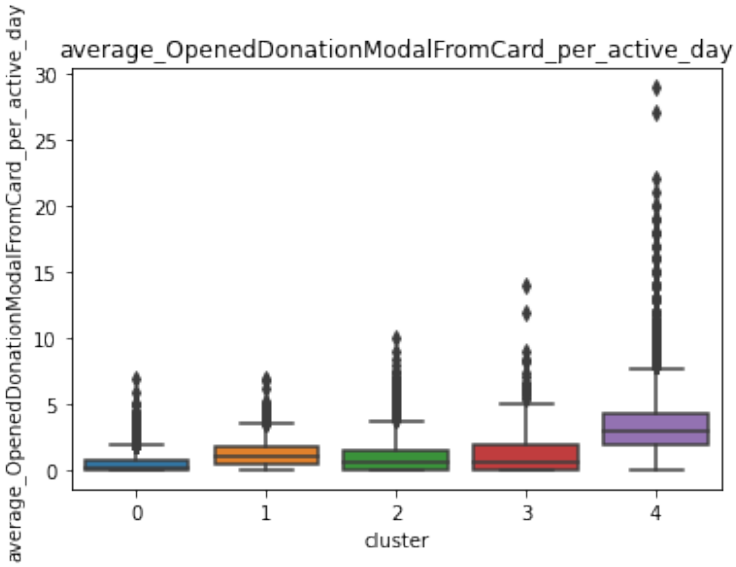

Figura 5.6: Análise de Diagrama de caixa para o evento average_OpenedDonationModalFromCard_per_active_day

Após atribuir um identificado aos 1241 usuários selecionados, foi possível identificar quais cluster possuem o maior número de assinantes tão bem quanto a razão entre usuários *Fremium* e *Premium*.

Rótulo do Cluster	**0**	1	2	3	4
É assinante?					
Não	39760	5202	2741	425	5428
Sim	414	622	65	7	133
Premium/Freemium (%)	1.0%	11.9%	2.3%	1.6%	2.4%

Tabela 5.4: Número de usuários Premium por Cluster

Observando a Tabela 5.4, é possível concluir que o cluster que possui mais assinantes em proporção em relação ao total de instâncias daquele agrupamento, é o cluster de número 1. As instâncias desses grupo apresentam comportamentos semelhantes ao de uso contínuo do aplicativo e, por isso, devem ser levadas em consideração para realizar quaisquer ações voltadas à conversão de usuários para o modelo *Premium*.

5.6 Discussão dos Resultados

A metodologia adotada foi produto da união de duas abordagens dentro da área de aprendizagem de máquina, sendo elas a abordagem supervisionada e não supervisionada. Do ponto de vista de implementação, a etapa de extração e concatenação dos dados é uma etapa que pode tornar-se complexa, uma vez que esta etapa depende de como os dados estão armazenados e como serão extraído para o formato utilizado pelo modelo proposto.

Ainda no que tange à complexidade, é possível afirmar que a etapa de seleção dos eventos mais relevantes para a aplicação deve ser levada em consideração. Para essa

etapa faz-se necessária a validação da escolha dos eventos com profissionais de negócio com experiência no domínio na aplicação.

Os resultados alcançados foram satisfatórios, uma vez que foi possível agrupar os usuários corretamente a partir de sua utilização, além de, sobretudo, identificar quais atributos mais influenciaram para esse agrupamento, disponibilizando possíveis novas abordagens do ponto de vista de produto para o negócio utilizado como estudo de caso no presente trabalho.

Apesar disso, uma lacuna encontrada que pode, e deve, ser trabalhada no futuro é a janela de observação dos eventos utilizados no modelo. Uma análise de safras para entender quanto tempo um usuário leva, em média, para realizar uma assinatura na plataforma pode ser utilizada como base para futuras versões do modelo.

As principais dificuldades encontradas durante o desenvolvimento do modelo tangem o negócio e a velocidade com que este pode mudar. Dado que o modelo está diretamente ligado com a regra do negócio por meio dos eventos, faz-se necessário recuperar os eventos mais recentes da plataforma constantemente e reavaliá-los em relação à sua relevância para objetivo do modelo. Para contornar tais problemas, o intervalo de tempo observado na pesquisa foi estudado para garantir pouca volatilidade em relação ao negócio. Além disso, é interessante tornar o modelo ainda mais modular de tal forma que seja prático modificar os eventos observados.

Por fim, por conta do tempo de execução do modelo, uma dificuldade ainda em aberto é automatizá-lo. Para isso, é necessário construir uma interface de consulta preferencialmente em *Python* e hospedá-la em uma máquina em um serviço de nuvem.

Capítulo 6

Conclusão

Como elucidado no capítulo 1, a conversão de usuário do modelo *Freemium* para o modelo *Premium* é um assunto que torna-se cada vez urgente para empresas que adotam esse modelo de negócio. Com base nisso, a presente pesquisa propôs um modelo com um esforço significativo na etapa de engenharia de atributos para agrupar usuários com mesmas características e identificar quais desses grupos possuem características que podem indicar potenciais usuários assinantes.

Além disso, uma técnica de aprendizagem supervisionada (previsões dos algoritmos) foi utilizada em conjunto com uma técnica de aprendizagem não supervisionada (clusterização) para compreender com mais detalhes o que mais influenciou no processo de seleção desses grupos gerados a partir da utilização dos usuário para com a plataforma.

O modelo mostrou resultados positivos em relação às suas previsões, mostrando que usuários podem ser agrupados em clusters específicos que representam suas características. Entretanto, um resultado ainda mais alinhado com a realidade pode ser obtido ao disponibilizar o modelo em produção, abastacendo-o com dados em tempo real.

Por fim, o modelo mostrou-se aberto a extensões, uma vez que é necessário compreender, a partir do objetivo a ser alcançado, quais eventos serão utilizados como entrada. Como exemplo seria possível aplicar a mesma técnica para o estudo de rotatividade de usuários na plataforma.

6.1 Trabalhos Futuros

A presente pesquisa teve como um de seus objetivos auxiliar a empresa utilizada como estudo de caso. Sendo assim, a empresa mostra-se otimista para desenvolver ainda mais as características apontadas por esse estudo.

Além disso, trabalhos de aprendizagem de máquina são extremamente ligados ao negócio. Uma vez que a empresa utilizada como estudo de caso na presente pesquisa passou, durante o tempo observado neste trabalho, por mudanças no modelo de negócio e em outras áreas que a tangem, o modelo será revisitado para observando as condições de negócio atuais.

Uma forma de melhorar ainda mais o modelo é utilizar formas de sobre amostragem para balancear o conjunto de dados com mais usuários *Premium*, como SMOTE (do inglês, *Synthetic Minority Oversampling Technique*). Por último, para adquirir ainda

mais aproximação com a realidade, o modelo será colocado em produção por parte do time de dados da empresa.

Referências

[1] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006. 11

[2] Daniel Berrar. Cross-validation. *Encyclopedia of bioinformatics and computational biology*, 1:542–545, 2019. 24

[3] Luo Bin, Shao Peiji, and Liu Juan. Customer churn prediction based on the decision tree in personal handyphone system service. In *2007 International Conference on Service Systems and Service Management*, pages 1–5. IEEE, 2007. 13

[4] Catherine Blake. Uci repository of machine learning databases. *http://www. ics. uci. edu/˜ mlearn/MLRepository. html*, 1998. 16

[5] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 9

[6] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015. 24

[7] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012. 22

[8] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, et al. Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88, 1996. 17

[9] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002. 10

[10] Gerard Goggin. Adapting the mobile phone: The iphone and its consumption. *Continuum*, 23(2):231–244, 2009. 1

[11] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems*, 5(4):83–124, 2011. 6, 7, 8, 9, 10, 11

[12] David J Hand and Niall M Adams. Data mining. *Wiley StatsRef: Statistics Reference Online*, pages 1–7, 2014. 5

[13] Jing He. Advances in data mining: History and future. In *2009 Third International Symposium on Intelligent Information Technology Application*, volume 1, pages 634–636. IEEE, 2009. 5

[14] Oumaima Alaoui Ismaili, Vincent Lemaire, and Antoine Cornuéjols. A supervised methodology to measure the variables contribution to a clustering. In *International Conference on Neural Information Processing*, pages 159–166. Springer, 2014. 15

[15] John D Kelleher, Brian Mac Namee, and Aoife D'arcy. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020. 14

[16] KPMG. Kpmg - going beyond data and analytics. Accessed: 2021-03-09. 2

[17] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *2010 IEEE international conference on data mining*, pages 911–916. IEEE, 2010. 16

[18] Douglas MacMillan, Peter Burrows, and Spencer E Ante. Inside the app economy. *Business Week*, 22:1–6, 2009. 1

[19] H Gilbert Miller and Peter Mork. From data to decisions: a value chain for big data. *It Professional*, 15(1):57–59, 2013. 3

[20] Andreas C Müller and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. "O'Reilly Media, Inc.", 2016. 10, 24, 29

[21] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6):275–285, 2004. 8

[22] Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana, Elias B Khalil, and Deepak S Turaga. Learning feature engineering for classification. In *IJCAI*, pages 2529–2535, 2017. 22

[23] David Nieborg. From premium to freemium: The political economy of the app. *Social, casual, and mobile games: The changing gaming landscape*, pages 225–240, 2016. 2, 13, 22, 24

[24] Nicolas Pujol. Freemium: attributes of an emerging business model. *Available at SSRN 1718663*, 2010. 1

[25] Thomas L Rakestraw, Rangamohan V Eunni, and Rammohan R Kasuganti. The mobile apps industry: A case study. *Journal of Business Cases and Applications*, 9:1, 2013. 1

[26] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 11

[27] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the stratification of multi-label data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–158. Springer, 2011. 24

[28] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001. 8

[29] Rafet Sifa, Fabian Hadiji, Julian Runge, Anders Drachen, Kristian Kersting, and Christian Bauckhage. Predicting purchase decisions in mobile free-to-play games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 11, 2015. 12

[30] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009. 7

[31] Rui Xu and Don Wunsch. *Clustering*, volume 10. John Wiley & Sons, 2008. 10

[32] Carl Yang, Xiaolin Shi, Luo Jie, and Jiawei Han. I know you'll be back: Interpretable new user clustering and churn prediction on a mobile social application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 914–922, 2018. 15