UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

VINÍCIUS FRAGA DE CASTRO

Auditoria em questionário de estudo epidemiológico utilizando os algoritmos *K-Means* e *FindCBLOF*

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Profª. Dra. Mariana Recamonde Mendoza

Porto Alegre
2019

RESUMO

Em estudos epidemiológicos, a qualidade dos questionários e dos dados coletados são fatores determinantes para a validade das conclusões. Nesse contexto, o Estudo Longitudinal de Saúde do Adulto (ELSA-Brasil) – o maior estudo epidemiológico em desenvolvimento na América Latina – adota variadas estratégias de controle de qualidade, conforme o tipo de questionário aplicado. Um dos questionários aplicados no ELSA-Brasil é o Formulário de Revisão Cardiovascular (FRC), cujo preenchimento é feito por um médico especialista, a partir da análise de diversos exames e registros de procedimentos realizados pelo participante do estudo, relacionados a eventos cardiovasculares. Como a entrada dos dados é manual, e muitos dos registros são em papel, esta atividade é propensa a erros. O FRC passa por um processo de auditoria, onde é realizada uma amostragem dos questionários, cujas respostas são conferidas manualmente na Plataforma Otus – plataforma tecnológica oferecida pela empresa Otus Solutions, para construção de questionários e gerenciamento de entrevistas – a fim de identificar possíveis erros de preenchimento.
Dessa forma, uma técnica capaz de auxiliar no processo de auditoria, processando o conjunto de dados do questionário e indicando quais potencialmente possuem erros de preenchimento, torna-se desejável. Uma limitação existente para a escolha da técnica a ser utilizada é a não disponibilidade dos registros de erros encontrados e corrigidos em auditorias anteriores. Assim, uma abordagem de aprendizado de máquina não supervisionado mostra-se adequada. O objetivo deste trabalho é aplicar o algoritmo de agrupamento K-means juntamente com o algoritmo de detecção de anomalias FindCBLOF ao conjunto de dados do questionário FRC, e analisar sua capacidade de identificar possíveis erros de preenchimento, através da introdução artificial de erros no conjunto de dados.
A partir da aplicação dos algoritmos citados ao conjunto de dados do questionário FRC, extraídos da Plataforma Otus, observa-se que quase metade dos erros introduzidos são detectados, para um tipo de questão. Para outros tipos, a capacidade de detecção é inferior. Conclui-se que mais estudos são necessários para identificar técnicas com possam auxiliar de maneira mais efetiva a auditoria dos questionários do ELSA-Brasil.

Palavras-chave: Aprendizado de máquina. Detecção de anomalias. K-Means. FindCBLOF.

Auditorship of epidemiological study questionnaire using the *K-Means* and *FindCBLOF* algorithms

ABSTRACT

In epidemiological studies, the quality of questionnaires and collected data are determining factors for the legitimacy of conclusions. In this context the ELSA-Brasil (Estudo Longitudinal de Saúde do Adulto – Brazilian Longitudinal Study for Adult Health), the biggest epidemiological study in development in Latin America – adopts a variety of strategies for quality control, depending on the type of questionnaire that's applied. One of these questionnaries is the FRC (Formulário de Revisão Cardiovascular – Cardiovascular Review Form) which is filled by a specialized doctor, according to their analysis of different exams and procedure records done by the study's participant, related to cardiovascular incidents. Since data entry is manual and many of the records are done in paper, this activity is prone to mistakes. FRC passes through an auditorship process where a sample of the questionnaires is performed and its answers are manually checked in the Otus Platform – a technological platform offered by Otus Solutions to build questionnaires and manage interviews – aiming to identify possible filing errors.

Thus, it becomes desirable to have a technique capable of assisting in the auditorship procedure, processing the questionnaire's dataset and indicating which ones can potentially contain filing errors. An existing limitation to the choice of technique is the unavailability of error records found and corrected in previous auditorships. Thus, an unsupervised machine learning approach is suitable. This work aims to apply the clustering algorithm K-means and the outlier detection algorithm FindCBLOF to the FRC questionnaire dataset, to analyze its capability of identifying possible filing errors through introducting artificial errors to the dataset.

When applying the previously mentioned algorithms to the FRC dataset, extracted from the Otus Patform, it is possible to observe that almost half of the introducted errors are detected, for one type of question. For other types, the detection ability is inferior. In is concluded that more studies are needed to identify techniques able to assist in a more effective way the auditorship of ELSA-Brazil questionnaires.

Keywords: Machine learning. Outlier detection. K-Means. FindCBLOF.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
CBLOF	Cluster-Based Local Outlier Factor
ELSA	Estudo Longitudinal de Saúde do Adulto
FRC	Formulário de Revisão Cardiovascular

SUMÁRIO

1 INTRODUÇÃO

Os estudos epidemiológicos buscam entender a distribuição e os fatores determinantes de doenças ou outras condições relacionadas à saúde em uma população. Classificam-se como analíticos os que examinam a existência de associações entre os fatores de exposição da população e a ocorrência das doenças. Entre os vários tipos de estudos analíticos, encontram-se os de coorte. Nestes, os indivíduos são classificados em expostos e não expostos a certos fatores, e são acompanhados por determinado período para verificação da incidência da doença/condição em cada grupo (LIMA-COSTA; BARRETO, 2003).

O Estudo Longitudinal de Saúde do Adulto (ELSA-Brasil), é o maior estudo epidemiológico em desenvolvimento na América Latina. Anunciado oficialmente em 2008 pelo Ministério da Saúde, tem como objetivo determinar a incidência e fatores de risco de doenças crônicas, em particular as cardiovasculares e o diabetes (MINISTÉRIOS, 2009). É um estudo de coorte que acompanha cerca de 15.000 participantes através de exames e entrevistas, realizadas em seis centros de pesquisa de diferentes regiões do Brasil (SCHMIDT et al., 2013).

Dado o porte e a natureza multicêntrica do ELSA-Brasil, sistemas de informação especializados tornam-se necessários. Neste contexto, a empresa Otus Solutions surge para suprir essa necessidade, oferecendo uma plataforma tecnológica para coleta de dados clínicos, monitoramento do andamento da coleta, construção de questionários e gerenciamento de entrevistas.

1.1 Motivação

A qualidade dos questionários e dos dados coletados são fatores determinantes para a validade das conclusões em estudos epidemiológicos (CHOR et al., 2013). Nesse sentido, o ELSA-Brasil adota variadas estratégias de controle de qualidade, realizadas durante a coleta e processamento dos dados, como a observação periódica dos entrevistadores, estudos de teste reteste, monitoramento estatístico dos dados, e visitas cruzadas aos centros de coleta (SCHMIDT et al., 2013). Ainda, estratégias específicas podem ser adotadas conforme o tipo de questionário aplicado.

Na Plataforma Otus, os questionários apresentam-se em duas categorias, de acordo com o método de entrada das respostas:

- Questionários *online*: são preenchidos durante as entrevistas com os participantes do estudo.
- Questionários *offline*: são preenchidos a partir de dados de exames ou outros dados clínicos, a variar pelo objetivo do questionário.

Para os questionários *online*, uma das estratégias adotadas é a gravação em áudio das entrevistas. Posteriormente, as entrevistas são amostradas e analisadas para garantir que o protocolo estabelecido foi seguido sem nenhum desvio por parte do entrevistador, além de serem checadas se as respostas dadas pelo participante foram devidamente registradas na Plataforma Otus.

Os questionários *offline* também passam por processo de amostragem, e têm suas respostas conferidas manualmente na Plataforma Otus, a fim de identificar possíveis erros de preenchimento. Ressalta-se que, no processo de desenvolvimento dos questionários, são incorporadas à Plataforma Otus regras de validação das questões como, por exemplo, a determinação de um intervalo possível de valores. Assim, os erros de entrada são restringidos, mas não eliminados completamente. O processo de amostragem e verificação manual das respostas é um controle adicional, visando aumentar ainda mais a confiabilidade dos dados.

Para este trabalho, foram realizadas entrevistas com especialistas em um dos centros de coleta do ELSA-Brasil, que relataram alguns dos tipos de erros encontrados em auditorias:

- Digitação incorreta de valores em questões numéricas;
- Seleção incorreta de alternativa em questões de múltipla escolha;
- Marcação incorreta de opção em questão de seleção múltipla.

Entre os questionários aplicados no ELSA-Brasil, encontra-se o Formulário de Revisão Cardiovascular (FRC). Trata-se de um questionário *offline*, cujo preenchimento na Plataforma Otus é feito por um médico especialista, a partir da análise de diversos exames e registros de procedimentos realizados pelo participante, relacionados a eventos cardiovasculares. Como a entrada dos dados é manual e muitos dos registros são em papel, esta atividade é propensa aos erros destacados anteriormente. Dessa forma, uma técnica capaz de auxiliar no processo de auditoria, processando o conjunto de dados completo do questionário e indicando quais dos preenchimentos potencialmente possuem erros, torna-se desejável.

Uma limitação existente para a escolha da técnica a ser utilizada é a não disponi-

bilidade, durante o desenvolvimento deste trabalho, dos registros de erros encontrados e corrigidos em auditorias anteriores. Dessa forma, a escolha da técnica dá-se considerando exclusivamente os dados do questionário e a informação relatada sobre os tipos de erros já encontrados.

Considera-se uma instância do FRC como um conjunto de valores preenchidos, para cada uma das questões. Pode-se pensar que instâncias cujas respostas assemelham-se formam agrupamentos. A quantidade de agrupamentos e o quão semelhantes são as instâncias pertencentes a cada um deles, são características inerentes à distribuição dos dados do questionário. Em princípio, tais características são desconhecidas. Porém, gerando-se uma estrutura de agrupamentos, poderia-se observar se a alteração no valor de resposta de uma questão afeta a similaridade da instância em relação ao seu agrupamento original. Havendo a ocorrência de observações desta natureza, poderia ser atribuída uma indicação de que a instância contém um potencial erro de preenchimento.

Dada a ausência de registros de erros antecedentes, uma abordagem de aprendizado de máquina não supervisionado mostra-se adequada. Partindo da noção de agrupamento de instâncias do FRC de acordo com a similaridade das respostas, atenta-se para os algoritmos de agrupamento, ou *clustering*, dos quais destaca-se o *K-means*, popular algoritmo baseado em partições (XU; TIAN, 2015). A identificação de instâncias que diferenciam-se significativamente das demais instâncias do seu agrupamento é uma tarefa que pode ser tratada como detecção de anomalias (outliers). Nesse sentido, apoia-se na área de detecção de anomalias, mais especificamente na definição de anomalia local baseada em agrupamento, ou *cluster-based local outlier factor* (CBLOF), proposta em conjunto com o algoritmo *FindCBLOF* por He, Xu and Deng (2003). Estes conceitos são apresentados com detalhes no Capítulo 2.

1.2 Objetivo

O objetivo deste trabalho é aplicar o algoritmo de agrupamento *K-means* juntamente com o algoritmo de detecção de anomalias *FindCBLOF* ao conjunto de dados do questionário FRC, e analisar sua capacidade de identificar possíveis erros de preenchimento. Para isso, é realizada uma análise exploratória dos dados do FRC extraídos da Plataforma Otus, seguida do pré-processamento do dados. Também são analisados e determinados os parâmetros para os algoritmos escolhidos. Por fim, erros são introduzidos artificialmente nos dados, em virtude da indisponibilidade de registro dos erros reais, e

verificado se os erros são detectados.

1.3 Organização do trabalho

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta os principais conceitos relacionados ao aprendizado de máquina, ao algoritmo *K-means* e sua validação, à detecção de anomalias e ao algoritmo *FindCBLOF*; o Capítulo 3 apresenta trabalhos relacionados ao tema; o Capítulo 4 apresenta a metodologia adotada na análise e no processamento dos dados, bem como na execução dos algoritmos; o Capítulo 5 apresenta os resultados obtidos e o Capítulo 6 apresenta a conclusão.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os conceitos e algoritmos principais utilizados neste trabalho, oferecendo o embasamento teórico necessário à compreensão do mesmo.

2.1 Aprendizado de Máquina

O Aprendizado de Máquina (AM) pode ser definido conceitualmente, de acordo com Mitchell (1997), como a capacidade de um programa de computador qualquer, por meio da experiência, de melhorar seu desempenho em uma determinada tarefa.

Em AM, a experiência é obtida a partir de um conjunto de dados que representam instâncias da tarefa a ser realizada. Distinguem-se três grandes áreas de AM, classificadas de acordo com o conjunto de dados disponível e tipo de tarefa a ser realizada: aprendizado supervisionado, não supervisionado e por reforço. O aprendizado supervisionado se dá quando um conjunto de dados de treinamento, com as respostas corretas para a tarefa, é apresentado, e o algoritmo, a partir desse conjunto, é capaz de fornecer respostas para novas instâncias da tarefa. O aprendizado não supervisionado ocorre com base em um conjunto de dados o qual não contém respostas corretas; o algoritmo busca identificar similaridades entre as instâncias do conjunto de dados, e de alguma forma categorizá-las ou agrupá-las. O aprendizado por reforço usualmente é associado com a ideia de um agente que observa o estado do ambiente ao qual está inserido, e que executa ações que recebem recompensas. O objetivo é a escolha de ações que maximizem as recompensas, considerando também as recompensas futuras (MARSLAND, 2014).

2.2 Aprendizado não supervisionado

Na ausência de informação prévia a respeito das classes às quais as instâncias do conjunto de dados pertencem, geralmente, técnicas de agrupamento, ou *clustering*, são aplicadas. Seu objetivo é organizar o conjunto dados de modo que instâncias similares pertençam ao mesmo agrupamento, enquanto que instâncias menos similares pertençam a agrupamentos diferentes.

Existem diversas abordagens para geração de agrupamentos. Em alto nível, pode-se apontar a hierárquica e a particional (JAIN, 2010). Na hierárquica, a ideia básica é

a construção de uma hierarquia de agrupamentos. Pode-se partir de agrupamentos com apenas uma instância, e sucessivamente agrupá-los de acordo com uma medida de similaridade entre os agrupamentos, até que se tenha apenas um agrupamento que contém todos os demais. De maneira oposta, pode-se partir de um único agrupamento que contém todas as instâncias, e iterativamente dividi-los conforme determinada medida de dissimilaridade entre os agrupamentos. Ambos os processos – de divisão ou de aglomeração – geram uma estrutura hierárquica chamada de dendrograma. Na abordagem particional, a ideia básica é a geração de um particionamento do espaço ao qual as instâncias pertencem, de modo que em cada partição a similaridade entre as instâncias seja maximizada. Assim, cada partição representa um grupo, ou cluster.

2.3 Algoritmo K-means

Entre os algoritmos de agrupamento não supervisionado, o K-means destaca-se como um dos mais populares (XU; TIAN, 2015). Cada agrupamento é representado pela média dos valores dos atributos das instâncias pertencentes ao mesmo. Esta média também é chamada de centroide. Os agrupamentos são formados buscando minimizar a soma do quadrado da distância entre cada instância do agrupamento e seu centroide – a inércia ou o erro quadrático. A partir de um dado número k, o algoritmo K-means busca, iterativamente, formar k agrupamentos cuja inércia total seja minimizada. Um exemplo de iteração do algoritmo com k=3, em um conjunto de dados bidimensional, é apresentado na Figura 2.1.

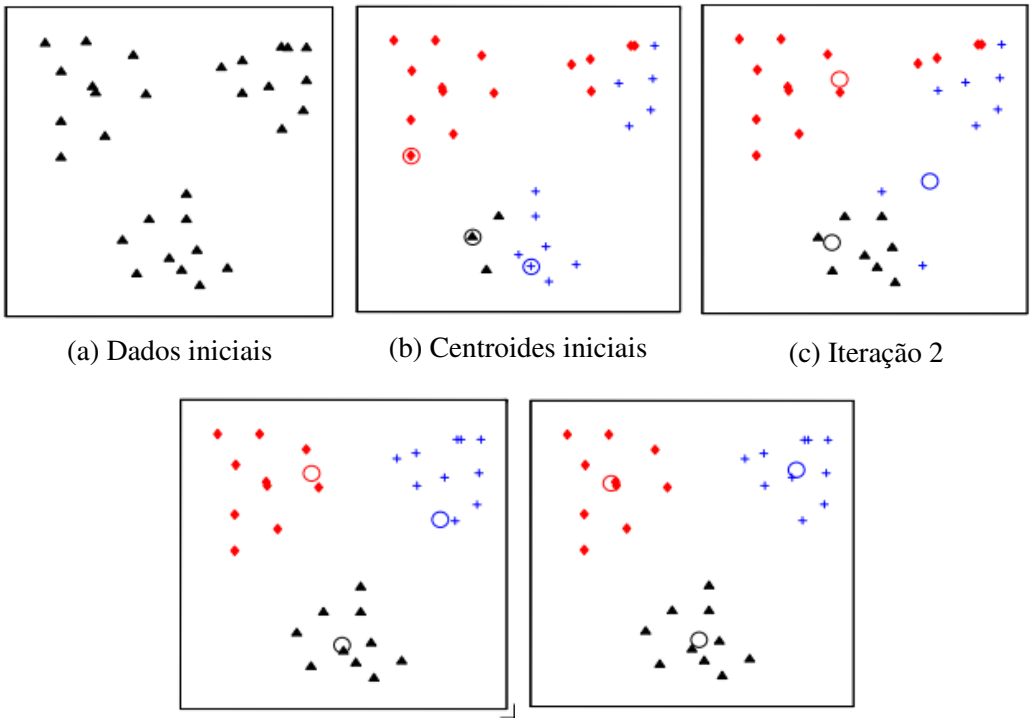
O algoritmo consiste em três etapas principais (JAIN, 2010):

1. Selecionar K centroides iniciais; repetir 2 e 3 enquanto houver alterações nos agrupamentos.
2. Associar cada instância ao agrupamento com centroide mais próximo.
3. Computar os novos centroides.

2.4 Validação dos agrupamentos

Uma vez aplicado um algoritmo de agrupamento, é desejável obter-se uma medida da qualidade dos agrupamentos gerados. Destacam-se duas abordagens para a validação

Figura 2.1: K-means aplicado a conjunto de dados bidimensional com três clusters



(a) Dados iniciais (b) Centroides iniciais (c) Iteração 2

(d) Iteração 3 (e) Iteração final

Fonte: Adaptado de (JAIN, 2010, p. 654)

dos agrupamentos: validação externa e interna. Na validação externa, indicadores de qualidade são calculados com base na informação prévia dos agrupamentos aos quais cada instância do conjunto de dados pertence. Na validação interna, os indicadores baseiam-se exclusivamente nos dados e agrupamentos gerados pelos algoritmos, sem qualquer informação prévia a respeito de quantos ou quais são os agrupamentos. Assim, este indicador pode auxiliar na tarefa de determinar o número de agrupamentos adequado ao conjunto de dados em análise.

Na literatura, encontra-se uma grande variedade de indicadores de validação interna. Segundo Arbelaitz et al. (2013), em experimentos comparando trinta indicadores, o *Silhouette score* é o que apresenta os melhores resultados para conjuntos de dados de diferentes domínios. Portanto, optou-se pela utilização deste na avaliação dos agrupamentos gerados pelo K-means, quando aplicado ao conjunto de dados do FRC.

2.4.1 Silhouette Score

De acordo com Rousseeuw (1987), a definição do Silhouette Score parte do cálculo de um coeficiente para cada instância do conjunto de dados. Este coeficiente possui

duas componentes:

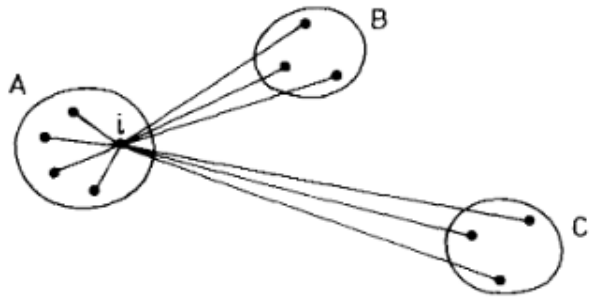
- $a(i)$: distância média entre a instância i e as demais instâncias do mesmo agrupamento;
- $b(i)$: distância média entre a instância i e as demais instâncias do agrupamento mais próximo;

A combinação de $a(i)$ e $b(i)$ é dada pela seguinte fórmula:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

O coeficiente $s(i)$ é computado para cada instância i, e gera um número entre -1 e 1. Quanto mais próximo de 1 é o coeficiente para uma instância, maior a similaridade entre as instâncias do seu agrupamento, como também maior a dissimilaridade com as instâncias de outros agrupamentos. A média dos coeficientes de cada instância determina o Silhouette Score para o conjunto de dados completo, e indica a qualidade dos agrupamentos gerados. A Figura 2.2 ilustra as distâncias entre uma instância i e as demais pertencentes ao seu agrupamento, A, bem como as distâncias entre i e as instâncias dos outros agrupamentos, B e C. Neste caso, $b(i)$ é computado considerando o agrupamento mais próximo, B.

Figura 2.2: Distâncias que compõem o cálculo do coeficiente da silhueta

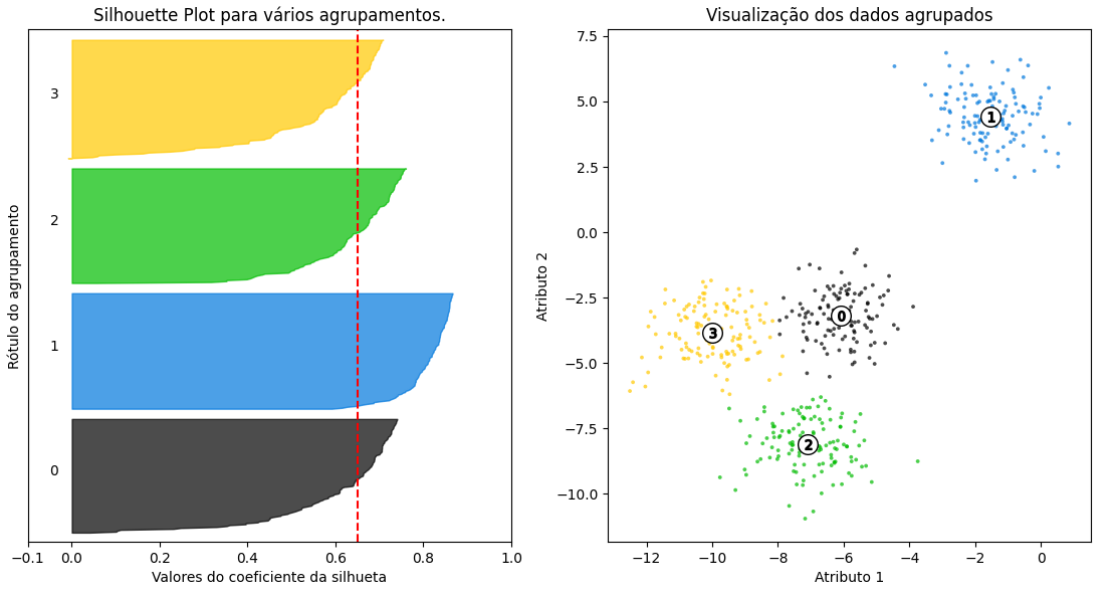


Fonte: (ROUSSEEUW, 1987)

Os coeficientes de cada instância podem ser organizados em função dos agrupamentos e apresentados graficamente, a fim de auxiliar na tarefa avaliação da qualidade dos agrupamentos gerados por um algoritmo. A Figura 2.3 mostra um exemplo de análise do Silhouette Score para dados de exemplo, agrupados pelo K-means. A partir dela, pode-se observar, no gráfico à esquerda: Silhouette Score alto (linha vermelha), acima de 0.6; os agrupamentos possuem tamanhos semelhantes e parte significativa dos coeficien-

tes de cada instância acima do coeficiente médio. Analisando a distribuição espacial dos dados, no gráfico à direita, percebe-se que a análise do Silhouette Score indica uma boa qualidade dos agrupamentos gerados, o que é condizente com o agrupamento existente, de fato, nos dados.

Figura 2.3: Análise da Silhueta para agrupamentos gerados pelo K-means em dados de exemplo



Fonte: Adaptado de (SCIKITLEARN, 2019b)

2.5 Detecção de Anomalias

A detecção de anomalias, de forma geral, trata do problema de encontrar padrões nos dados que não correspondem ao comportamento esperado. Nesse sentido, em um conjunto de dados, uma instância que difere significativamente das demais pode ser considerada uma anomalia. Este estudo encontra aplicação em diferentes domínios, como: detecção de fraudes em transações financeiras; detecção de intrusão em sistemas de informação; detecção de falhas em sistemas críticos, detecção de condições anormais de saúde em pacientes, entre outros. Técnicas de detecção de anomalias apresentam-se, usualmente, em três modos: supervisionadas, semi-supervisionadas e não supervisionadas. Nota-se que a obtenção de dados que representem todos os comportamentos possíveis e sua classificação, total ou parcial, como normais ou anômalos, pode ser inviável em muitos domínios. Sendo assim, usualmente, técnicas não supervisionadas são aplicadas.

(CHANDOLA; BANERJEE; KUMAR, 2009).

Uma perspectiva importante sobre as técnicas de detecção de anomalias, é o modo como as anomalias identificadas são apresentadas. Salientam-se dois tipos de técnicas:

- Baseadas em score: atribuem um score para cada instância, de acordo com o grau de anomalia considerado.
- Baseadas em rótulos: atribuem um rótulo, anômalo ou não anômalo, para cada instância.

Diferentes técnicas de detecção de anomalias apoiam-se em suposições acerca de agrupamentos existentes no conjunto de dados. Conforme Chandola, Banerjee and Kumar (2009), estas técnicas dividem-se de acordo com três suposições:

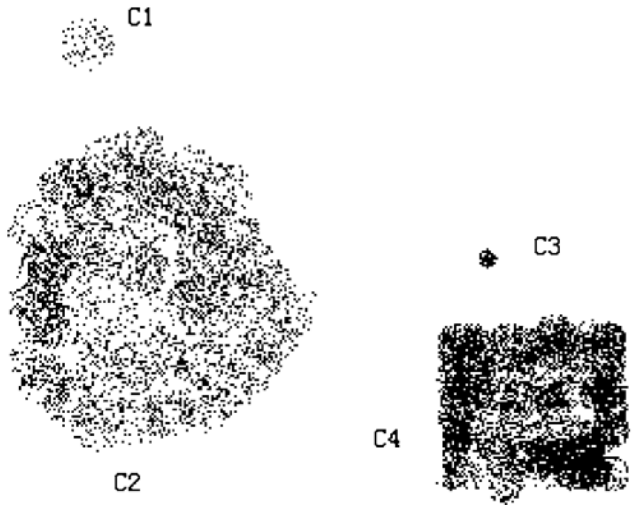
- Instâncias normais naturalmente agrupam-se, enquanto instâncias que não pertencem a nenhum agrupamento são anomalias;
- Instâncias normais localizam-se próximas ao centroide do agrupamento, enquanto as anômalas estão distantes do centroide;
- Instâncias normais pertencem a agrupamentos grandes e densos, enquanto as anômalas pertencem a agrupamentos pequenos ou esparsos;

Neste trabalho, foca-se no algoritmo *Find Cluster-based Local Outlier Factor, FindCBLOF*, proposto por He, Xu and Deng (2003), que enquadra-se como uma variação da terceira categoria de técnicas, pois considera na avaliação das instâncias não só o tamanho do agrupamento, como também a distância entre a instância e o centroide do agrupamento a qual pertence. A partir das ideias de que (i) muitas instâncias do FRC com respostas similares formariam agrupamentos grandes e densos, (ii) instâncias com respostas diferentes formariam agrupamentos menores e esparsos, e (iii) uma instância com um erro de preenchimento distanciaria-se do agrupamento a qual pertence, a aplicação do algoritmo *FindCBLOF* apresenta potencial para detecção de erros de preenchimento, a ser investigado.

2.5.1 Algoritmo *FindCBLOF*

O algoritmo *FindCBLOF* apoia-se em duas definições: anomalias locais baseadas no agrupamento (*cluster-based local outlier – CBLO*), e fator de anomalia (*cluster-based local outlier factor – CBLOF*). A Figura 2.4 ilustra um conjunto de dados bidimensio-

Figura 2.4: *Cluster-based local outlier* em conjunto de dados bidimensional



Fonte: (HE; XU; DENG, 2003)

nal, contendo quatro agrupamentos. De forma intuitiva, pode-se tratar as instâncias dos agrupamentos $C1$ e $C3$ como anomalias, uma vez que não pertencem aos outros dois agrupamentos maiores e densos, que contêm instâncias que seriam normais. Ainda, pode-se observar a ideia de localidade entre os agrupamentos: $C1$ é local em relação a $C2$, pois $C2$ é o agrupamento grande mais próximo; $C3$ é local em relação a $C4$, da mesma forma (HE; XU; DENG, 2003).

Dados um conjunto de instâncias D e um conjunto $C = \{C_1, C_2, ..., C_k\}$ de agrupamentos, em ordem decrescente em função do seu tamanho (número de instâncias), busca-se determinar qual o agrupamento que estabelece a divisão entre agrupamentos grandes e pequenos. Para isso, avalia-se duas condições, dados dois parâmetros, α e β:

$$|C_1| + |C_2| + ... + |C_b| \geq |D| * \alpha \tag{2.1}$$

$$|C_b|/|C_{b+1}| \geq \beta \tag{2.2}$$

Satisfeita uma das condições acima, define-se o conjunto de agrupamentos grandes como $LC = \{C_1, C_2, ..., C_b\}$, e o conjunto de agrupamento pequenos como $SC = \{C_{b+1}, C_{b+2}, ..., C_k\}$. A condição (2.1) considera que a maior parte dos dados não são anomalias. Por exemplo, α igual a 95% significa que os agrupamentos que contêm 95% dos dados serão considerados grandes. A condição (2.2) enfatiza que agrupamentos grandes devem ser notadamente maiores que os pequenos. Por exemplo, com β igual a 3, agrupamentos grandes serão, no mínimo, 3 vezes maiores que um agrupamento pequeno.

Para determinar o fator de anomalia CBLOF de uma instância i pertencente ao agrupamento C_i, calcula-se:

- tamanho de C_i multiplicado pela distância de i ao agrupamento grande mais próximo, se C_i for um agrupamento pequeno; ou
- tamanho de C_i multiplicado pela distância de i ao próprio C_i, se este for grande.

As distâncias, neste caso, podem ser as mesmas utilizadas no algoritmo de agrupamento aplicado, como a Euclidiana, por exemplo. Pode-se considerar o centroide para cáculo das distâncias entre instâncias e agrupamentos. O cálculo do fator de anomalia é realizado para cada instância do conjunto de dados, conforme mostrado no Algoritmo 1. Por fim, pode-se definir as n instâncias com os maiores fatores como anômalas.

Algorithm 1: *FindCBLOF*

Data: Conjunto de dados $D(A_1, ..., A_m)$, parâmetros α e β

Result: Valores de *CBLOF* para todas as instâncias

```
1  Gera um conjunto de clusters C utilizando um algoritmo de agrupamento;
2  Determina os small clusters SC e os large clusters LC a partir de α e β;
3  foreach i ∈ D ;          // Para cada instância do dataset
4  do
5      if C_i ∈ SC then
6          C_j ← Cluster grande mais próximo;
7          CBLOF = |C_i| * distance(i, C_j) ;          //  C_j ∈ LC
8          else
9              CBLOF = |C_i| * distance(i, C_i) ;      //  C_i ∈ LC
10         end
11     end
12     return CBLOF
13 end
```

Fonte: Adaptado de (HE; XU; DENG, 2003).

3 TRABALHOS RELACIONADOS

No trabalho de Ferreira (2018), é proposto um sistema de predição de respostas de questionários para a Plataforma Otus. No contexto do controle de qualidade do ELSA-Brasil, um dos protocolos é a gravação das entrevistas realizadas no estudo, para posterior análise em busca de possíveis desvios no preenchimento dos questionários por parte do entrevistador. Para auxiliar nesta atividade, o trabalho propõe um modelo preditivo que busca tornar possível identificar padrões de respostas para perfis de preenchimento, apontar eventuais desvios, e assim direcionar esforços para o controle de qualidade apenas em itens relevantes. O trabalho é divido em duas partes: a elaboração do modelo de predição e a implementação do mesmo para posterior integração à Plataforma Otus.

Na primeira parte, são abordados dois modelos de classificação: Árvores de Decisão e Naive Bayes. Com o objetivo de classificar a resposta de uma questão como anômala, com base nas respostas às questões anteriores, os dois modelos são treinados com base em um conjunto de dados de um questionário do ELSA-Brasil, com cerca de 10.000 instâncias e 74 atributos. Segundo os resultados dos testes apresentados, o modelo de Árvore de Decisão foi escolhido para implementação. A segunda parte do trabalho foca na definição da arquitetura para a implementação do modelo e integração do mesmo à Plataforma Otus.

Por fim, concluiu-se que o modelo implementado é factível de ser aplicado para auxiliar na tarefa de auditoria dos questionários, porém, destacam-se diversas limitações, como a implementação apenas para questões de seleção única e perda de acurácia do modelo devido à constante alteração da base dos questionários.

No trabalho de Birnbaum (2012), é abordado o problema de identificar a fabricação de dados em questionários por parte dos entrevistadores, o que é uma preocupação primária de qualquer organização que lida com dados de pesquisas. São descritos e avaliados diferentes algoritmos de aprendizado supervisionado e de detecção de anomalias não supervisionada. Os algoritmos supervisionados apresentados foram a Regressão Logística e as Florestas Aleatórias. Os algoritmos de detecção de anomalias apresentados foram o Local Correlation Integral (LOCI), e outros dois propostos pelo autor: Multiminomial Model Algorithm (MMA) e s-Value Algorithm (SVA). Estes dois últimos consideram múltiplas instâncias do conjunto de dados, referentes ao mesmo entrevistador, de forma simultânea.

Adicionalmente, o trabalho propõe modificações em softwares geralmente utiliza-

dos em dispositivos móveis para coleta remota de dados de pesquisas, de forma a capturar dados da interação do entrevistador com o dispositivo e aprimorar a detecção de fabricação de dados.

Os algoritmos foram aplicados em dados reais de pesquisas em saúde, e constatou-se que são capazes de identificar a fabricação de dados em pesquisas de maneira precisa e robusta, mesmo quando os entrevistadores sabem que estas técnicas estão sendo aplicadas e, ainda, incentivados a tentar evitar a detecção. Por fim, os resultados podem ser melhorados a partir da coleta de dados dos dispositivos dos entrevistadores.

O trabalho de Bolton, Hand et al. (2001) aplica duas técnicas de detecção de anomalias não supervisionadas na identificação de fraudes em cartões de crédito. São utilizados os dados de transações dos cartões ao longo do tempo, e o objetivo é identificar transações não usuais ou mudanças de comportamento na utilização dos cartões.

A primeira técnica é chamada de Peer Group Analisys (PGA). Inicialmente, escolhe-se uma medida que sumariza a utilização de um cartão em um determinado período de tempo, por exemplo, o valor médio das transações na última semana. Em seguida, para cada cartão é criado um conjunto – chamado de Peer Group (PG) – que contém os demais cartões cuja utilização é similar. O tamanho do PG é um parâmetro, e controla a sensibilidade da técnica. A medida de similaridade aplicada é a distância Euclidiana das medidas de sumarização dos cartões. Após, calcula-se a medida de sumarização do PG. A ideia é que o PG forneça um modelo local de comportamento de um cartão em um instante de tempo. Assim, se em um instante de tempo subsequente um cartão apresente uma utilização que desvie significativamente do seu PG, marca-se o cartão para investigação de fraude.

A segunda técnica, chamada de Break Point Analysis, opera considerando os cartões individualmente. Define-se uma janela de transações de tamanho fixo em que, para cada transação nova adicionada à janela, a mais antiga é removida. Transações mais recentes são comparadas com as mais antigas, visando identificar mudanças na utilização dos cartões. O tamanho da janela e a diferença na proporção de transações novas e antigas são parâmetros da técnica. Testes estatísticos são aplicados às transações para identificar mudanças abruptas na frequência ou nos valores das compras.

4 METODOLOGIA

Os conceitos e técnicas apresentadas anteriormente fundamentam a metodologia aplicada neste trabalho, cujo objetivo é a aplicação do algoritmo de agrupamento não supervisionado K-Means, em conjunto com o algoritmo de detecção de anomalias FindCBLOF, bem como a investigação da capacidade deste algoritmo em identificar prováveis erros de preenchimento do questionário FRC do ELSA-Brasil.

A metodologia proposta baseia-se nas ideias de que (i) muitas instâncias do FRC com respostas similares formariam agrupamentos grandes e densos, (ii) instâncias com respostas diferentes formariam agrupamentos menores e esparsos, e (iii) uma instância com um erro de preenchimento distanciaria-se do agrupamento a qual pertence. Dessa forma, os algoritmos K-Means e FindCBLOF apresentam potencial para detecção de erros de preenchimento, a ser investigado.

Nas seções seguintes, são apresentados: o conjunto de dados do FRC; como foi realizado o pré-processamento dos dados; a escolha dos parâmetros para o algoritmo K-means; a escolha de parâmetros para o algoritmo FindCBLOF; e como se deu a introdução manual de erros nos dados, para a avaliação desta abordagem.

4.1 Conjunto de Dados

O FRC reúne questões relacionadas a eventos cardiovasculares ocorridos com os participantes do ELSA-Brasil. O preenchimento é realizado por especialistas, a partir de uma série de dados clínicos do participante. O questionário é composto de 52 questões, de 5 tipos diferentes, de acordo com os valores possíveis de resposta: questão de seleção única, questão do tipo data, questão do tipo checkbox, questão de valor inteiro e questão do tipo texto livre. O conjunto de questionários preenchidos, extraídos da Plataforma Otus, contém um total de 805 instâncias. A quantidade de questões por tipo é apresentada na Tabela 4.1.

Na construção do questionário FRC na Plataforma Otus, foram determinados, para as questões de seleção única, rótulos numéricos que representam cada alternativa, a fim de facilitar o posterior processamento dos dados. É possível, para todas as questões, indicar situações predeterminadas (metadados) como, por exemplo, a ausência de dados para responder à questão, ou que a questão não é aplicável. Ainda, há a possibilidade de incluir comentários em texto livre para qualquer questão. A Figura 4.1 mostra um

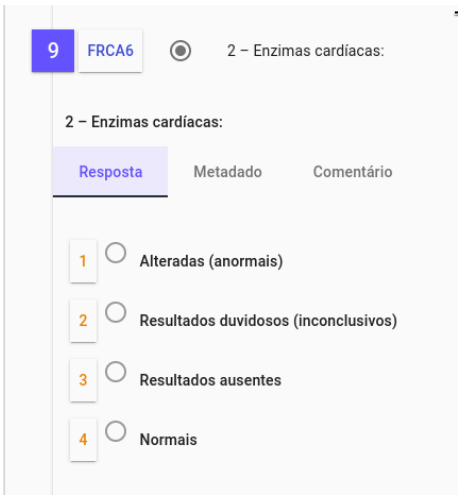
Tabela 4.1: Quantidade de questões por tipo

Tipo de questão	*Quantidade*	*%*
Seleção única	28	52.8
Data	11	20.8
Checkbox	5	9.4
Valor inteiro	4	7.5
Texto livre	4	7.5
Total	52	100

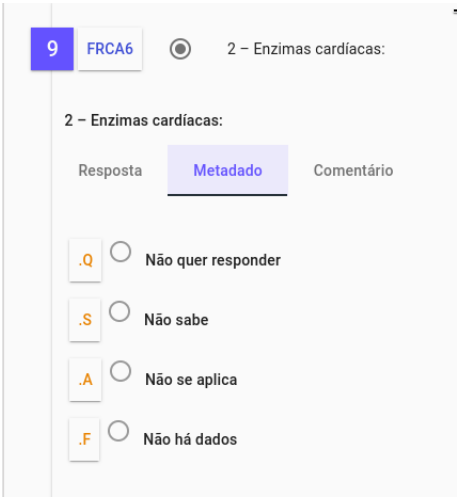
Fonte: O Autor

exemplo de questão de seleção única do FRC na Plataforma Otus, onde apresentam-se as alternativas de resposta, em forma de texto, e seus respectivos rótulos numéricos. Também são apresentados os metadados disponíveis e seus respectivos rótulos.

Figura 4.1: Rótulos de respostas e metadados de questão de seleção única na Plataforma Otus



(a) Alternativas de resposta e valores de extração

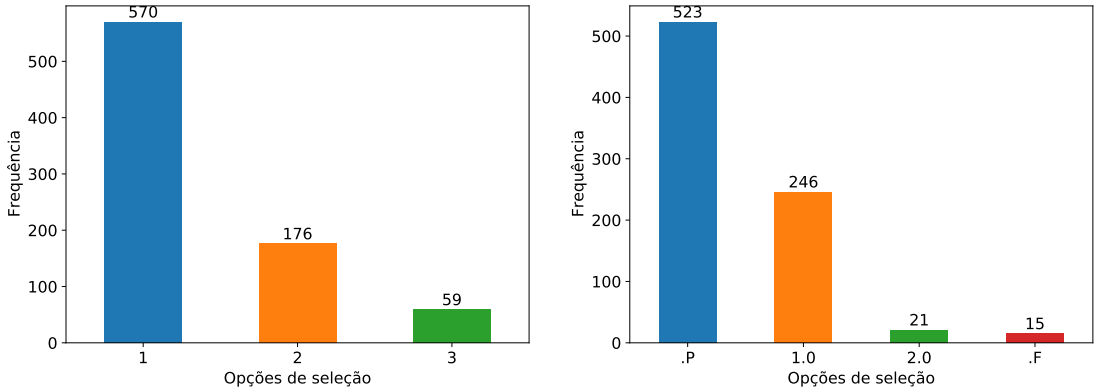


(b) Metadados e seus valores de extração

Fonte: Extraído da Plataforma Otus

Outro ponto a considerar-se na análise do conjunto de dados do FRC é a presença de saltos entre as questões. Na Plataforma Otus, pode-se criar regras em que, de acordo com condições de respostas anteriores, altera-se o fluxo do questionário. Questões que não foram respondidas em virtude de saltos, são preenchidas, por padrão, com um metadado específico para tal. Dessa forma, no conjunto de dados do FRC, existem várias instâncias cujos valores para as respostas é nulo, e que possuem o metadado de salto atribuído. A Figura 4.2a apresenta a quantidade de instâncias por resposta para uma questão de seleção única em que não ocorreu nenhum salto. A Figura 4.2b mostra a quantidade de instâncias por resposta para um questão de seleção única em que ocorreu uma quantidade expressiva de saltos, dada a quantidade de instâncias com o rótulo de extração ".P", que

Figura 4.2: Distribuição das respostas a algumas questões de seleção única do FRC



(a) FRCA1 (b) FRCA5

Fonte: O Autor

corresponde ao metadado de salto.

4.2 Pré-processamento

O pré-processamento dos dados é um aspecto importante em qualquer abordagem de AM. Geralmente, técnicas de pré-processamento são aplicadas a fim de melhorar a qualidade dos dados, bem como adequá-los à necessidade dos algoritmos de AM escolhidos. Para este trabalho, o conjunto completo dos dados do FRC foi extraído da Plataforma Otus, em um arquivo no formato CSV. A partir deste arquivo, aplicaram-se as operações descritas nas subseções seguintes, de modo que fosse possível a aplicação dos algoritmos de agrupamento K-Means e de detecção de anomalias FindCBLOF. As operações de pré-processamento foram implementadas utilizando a linguagem de programação Python, em conjunto com as bibliotecas numpy, pandas e scikit-learn – ferramentas populares para a análise de dados e AM.

4.2.1 Eliminação de atributos e instâncias

O arquivo de extração do FRC contém, originalmente, um total de 1087 instâncias, das quais 282 são questionários iniciados, porém não finalizados até o momento da extração. Dessa forma, procedeu-se com a filtragem das instâncias finalizadas e eliminação das demais, produzindo um novo conjunto composto de 805 instâncias.

O arquivo de extração do FRC possui um total de 194 atributos. Entre eles, consta

uma série de atributos de identificação do entrevistador e do participante. Estes atributos são preenchidos de forma automática pela Plataforma Otus, portanto, foram eliminados. Além destes, foram eliminados os atributos de comentários em texto livre, presentes para cada questão, uma vez que a análise de texto livre não faz parte do objetivo de identificação de erros de preenchimento das questões. O conjunto de dados resultante contém 120 atributos.

4.2.2 Transformação de dados

Após a filtragem das instâncias finalizadas e a eliminação dos atributos não utilizados, faz-se necessário o tratamento dos dados de acordo com o tipo de questão, uma vez que o algoritmo de agrupamento K-means pressupõe dados numéricos. A seguir, são detalhadas tais transformações.

4.2.2.1 Tratamento dos metadados

Na Plataforma Otus, para cada questão pode-se atribuir um valor especial quando não há resposta para a mesma – os metadados. No caso do FRC, os seguintes metadados foram definidos:

- Rótulo ".Q": não quer responder.
- Rótulo ".S": não sabe.
- Rótulo ".A": não se aplica.
- Rótulo ".F": não há dados.
- Rótulo ".P": definido implicitamente quando a questão é pulada.

No arquivo de extração do FRC, cada questão é representada por dois atributos: sua resposta e seu metadado. Os dois atributos não podem ser preenchidos simultaneamente, ou seja, quando um é preenchido, ou outro é nulo. Dessa forma, foi necessário realizar a junção dos dois atributos, para cada questão.

4.2.2.2 Questões do tipo seleção única

Os dados extraídos das questões do tipo seleção única são numéricos, em virtude do rótulo numérico atribuído a cada alternativa. Porém, as alternativas das questões são, originalmente, categóricas, e não guardam nenhuma relação de ordem entre si. Como os

algoritmos que serão aplicados aos dados considerarão os valores numéricos para cálculos de distâncias entre as instâncias, os rótulos numéricos, com diferentes magnitudes, podem induzir medidas incorretas. Assim, optou-se por aplicar uma codificação binária, também chamada de one-hot encoding: para cada alternativa de seleção, cria-se um novo atributo, cujo valor pode ser 0 – a alternativa não foi selecionada, ou 1 – a alternativa foi selecionada. Ressalta-se que, neste caso, também serão gerados atributos para os metadados, quando presentes, em virtude da junção dos metadados com as respostas, realizada no passo de pré-processamento anterior.

4.2.2.3 Questões do tipo checkbok

Nas questões do tipo checkbox, o arquivo de extração já apresenta um atributo para cada opção de seleção, com valores binários. Mas, como foi realizada a junção dos valores de metadados, os atributos não são mais binários, sendo necessário aplicar a mesma codificação novamente, para cada atributo.

4.2.2.4 Questões do tipo valor inteiro

Ainda que as questões do tipo valor inteiro possuam valor de extração numérico, as instâncias cujos valores são nulos (preenchidas com metadados), precisam de tratamento específico. Pode-se adotar diversas abordagens para imputação de valores faltantes, como, por exemplo, estimar o valor a partir da média, ou mediana, do atributo. Todavia, considera-se que o valor do metadado caracteriza uma resposta diferente das presentes nos valores numéricos do atributo. Fazer essa diferenciação, ao invés de estimar o valor faltante, pode ser útil no agrupamento das instâncias que possuem o mesmo metadado como resposta. Assim, adotou-se uma codificação específica para as instâncias com metadados: atribuiu-se um valor numérico para cada rótulo de metadado, de modo que este valor esteja fora do intervalo de valores originais do atributo. Sabe-se que no FRC existem apenas 3 questões do tipo valor inteiro, e todas são questões com respostas em valores percentuais, ou seja, entre 0 e 100. Logo, os metadados foram codificados com valores acima de 100. Por exemplo, o metadado "Não se aplica" recebeu o valor 300. Um aspecto não analisado na metodologia proposta é o impacto da codificação dos metadados nos agrupamentos gerados e na detecção dos erros.

4.2.2.5 Questões do tipo data

As questões do tipo data têm como valor de extração uma String. Aplicou-se, à todas as instâncias com valor não-nulo para o atributo, uma conversão de String para *Unix Timestamp* (segundos desde 1 de Janeiro de 1970). Esta conversão foi escolhida por ser uma representação numérica simples e bastante utilizada para datas. Após, aplica-se o mesmo tratamento das questões do tipo valor inteiro, em relação às instâncias com valor de metadado.

4.2.2.6 Normalização

Após as transformações aplicadas aos dados, todos os atributos possuem valores numéricos. O último passo no pré-processamento, é a normalização dos dados, para que estes apresentem a mesma escala. Sem a normalização, atributos com magnitudes diferentes influenciariam de maneira incorreta as distâncias calculadas pelos algoritmos K-Means e FindCBLOF. Por exemplo, uma instância do FRC possui valor de uma questão do tipo data, no formato timestamp, igual a 1243900800, enquanto que a mesma instância possui o valor de outra questão, do tipo valor inteiro, igual a 42. Portanto, foi aplicada a normalização por reescala, no intervalo de 0 a 1, dada pela fórmula abaixo, onde x é um atributo do conjunto de dados, e x_i é o valor do atributo para uma instância i:

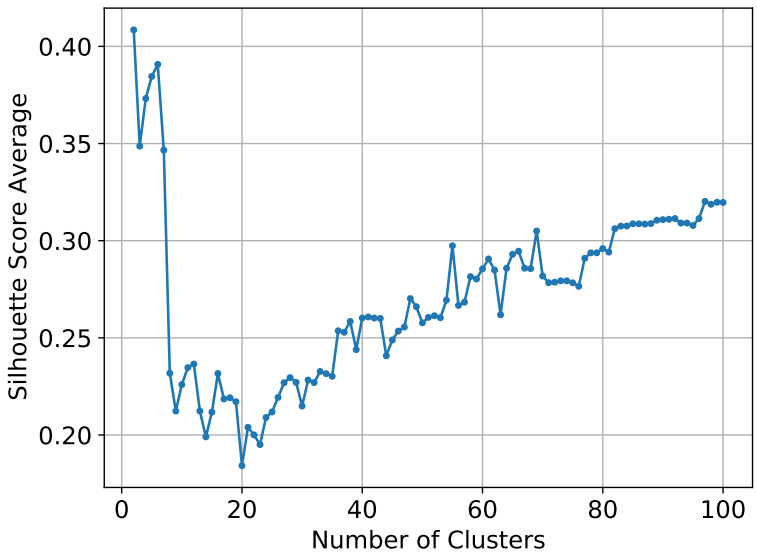
$$x_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{4.1}$$

O conjunto de dados final, após a eliminação de instâncias e atributos, o tratamento dos metadados, a transformação dos dados para cada tipo de questão, e a normalização dos dados, possui 805 instâncias e 304 atributos.

4.3 Parâmetros para o K-Means

O algoritmo K-Means busca separar o conjunto de dados em k agrupamentos, de modo que a soma do quadrado das distâncias entre as instâncias e os centroides seja minimizada. Logo, a escolha do parâmetro k é fundamental para a efetividade da tarefa de agrupamento aplicada. No Capítulo 2, foi apresentada uma medida de validação interna de agrupamentos, o silhouette score, bem como sua visualização gráfica, a fim de auxiliar a escolha do parâmetro k. Nesta seção, analisa-se a média do silhouette score de todos

Figura 4.3: Silhouette Scores para diferentes valores de k

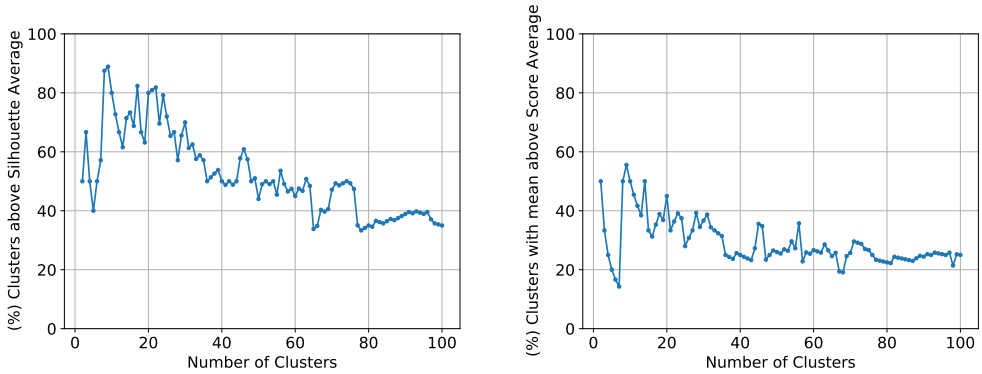


Fonte: O Autor

os agrupamentos, para diferentes valores de k. Após, observa-se a quantidade de agrupamentos com o silhouette score acima da média, ao variar k. Por fim, com um número limitado de valores de k selecionado, realiza-se a análise gráfica do silhouette score. A implementação do K-Means utilizada é a presente na biblioteca scikit-learn, na linguagem Python (SCIKITLEARN, 2019a)

A Figura 4.3 apresenta o silhouette score para valores de k no intervalo de 2 a 100. A partir dela, observa-se scores mais altos para os valores iniciais de k. Para valores de k acima de 20, mostra-se uma tendência de crescimento dos scores. Isto poderia indicar que o aumento de k produziria agrupamentos cada vez mais bem definidos. No entanto, ao observar-se a Figura 4.4a, nota-se que o percentual de agrupamentos com silhouette score acima da média decresce com o aumento de k. Neste percentual, considera-se acima da média qualquer agrupamento em que pelo menos uma instância possui seu score acima da média. Pode-se, ainda, analisar o percentual de agrupamentos cuja média dos scores de suas instâncias esteja acima do score médio de todos os agrupamentos. Esta análise é mostrada na Figura 4.4b. Neste caso, constata-se que o percentual de agrupamentos acima da média torna-se ainda menor. Dessa forma, escolheu-se, para análise gráfica do silhouette score, os valores de k que produzem o maior percentual de agrupamentos com sua média acima da média total, bem como aqueles com as maiores médias totais: 6, 7, 10, e 18.

Figura 4.4: Percentual dos agrupamentos com silhouette score acima da média para diferentes valores de k



(a) Ao menos uma instância acima do score médio

(b) Média das instâncias acima do score médio

Fonte: O Autor

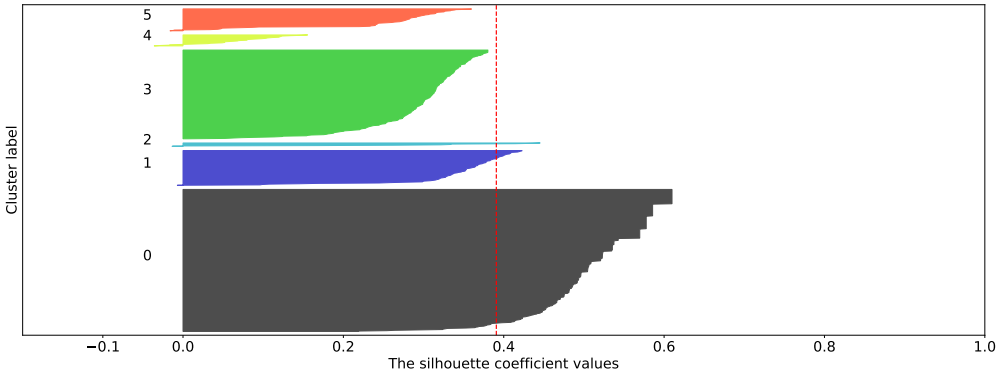
A visualização gráfica dos silhouettes scores, para k igual a 6, 7, 10, e 18, é apresentada na Figura 4.5. Com k igual a 18, observa-se muitos agrupamentos com scores negativos. Isto indica a presença de instâncias que foram atribuídas ao agrupamento incorreto. A mesma observação vale para k igual a 10. Além disso, para ambos os valores de k, o silhouette score é pouco maior que 0.2, o que é baixo, considerando-se que o score máximo é 1. Para k igual a 7, o silhouette score é maior: 0.34. Porém, os agrupamentos de rótulos 3 e 4 possuem scores bem abaixo da média, bem como scores negativos. Já para k igual a 6, observa-se o maior silhouette score: 0.39. Ainda, o agrupamento de rótulo 3 está bem próximo da média, e sem presença de scores negativos. A partir desta análise, escolheu-se o valor de k igual a 6 para o algoritmo K-Means.

Além do número de agrupamentos k, um parâmetro importante na implementação do K-Means pela biblioteca scikit-learn, é o n_init, que indica quantas inicializações dos centroides iniciais serão realizadas. O melhor desempenho, em termos da inércia, define quais serão os centroides iniciais. Utilizou-se o parâmetro n_init com valor igual a 10.

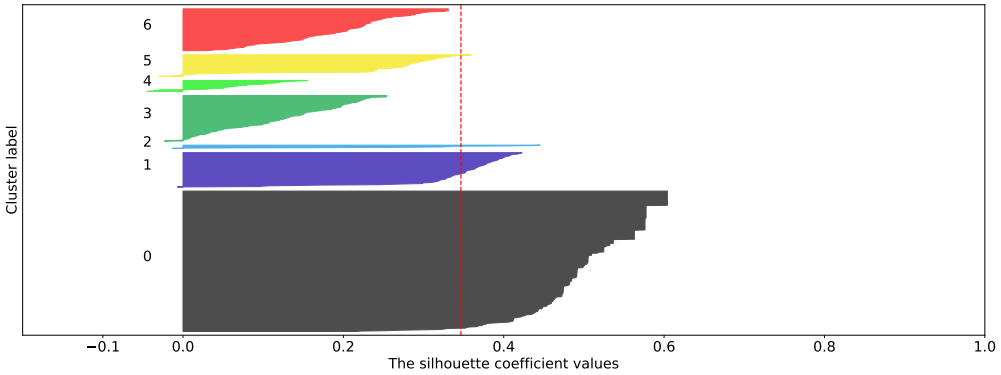
4.4 Parâmetros para o FindCBLOF

O algoritmo FindCBLOF define os conjuntos SC (small clusters) e LC (large clusters), a partir dos parâmetros α e β, respectivamente. O score de anomalia, CBLOF, é calculado para cada instância com base no tamanho do cluster ao qual pertence, e na distância ao cluster mais próximo pertencente a LC – caso a instância pertença à um cluster do conjunto SC, ou na distância ao próprio cluster, caso contrário.

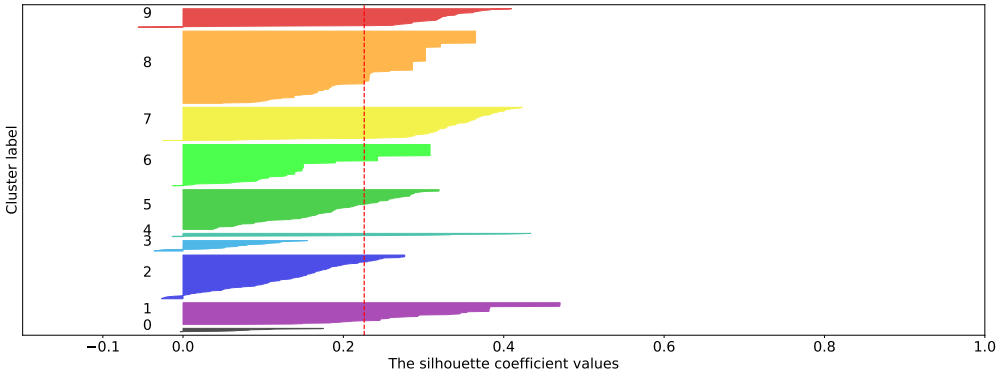
Figura 4.5: Gráficos do Silhouette Score para diferentes valores de k



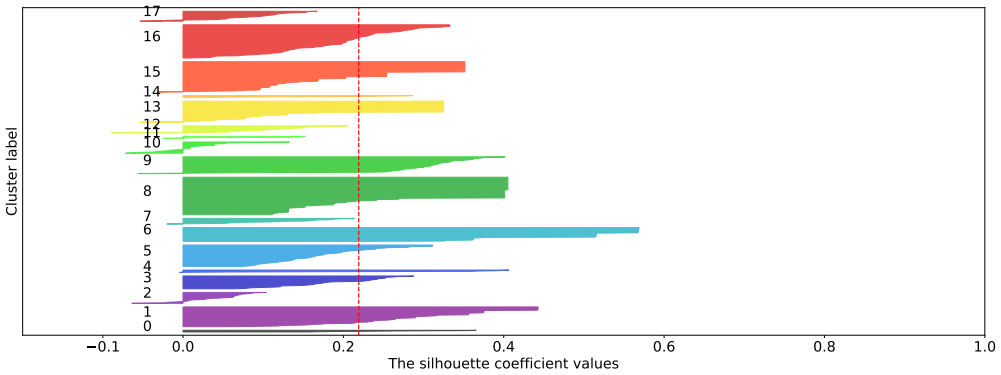
(a) $k = 6$



(b) $k = 7$



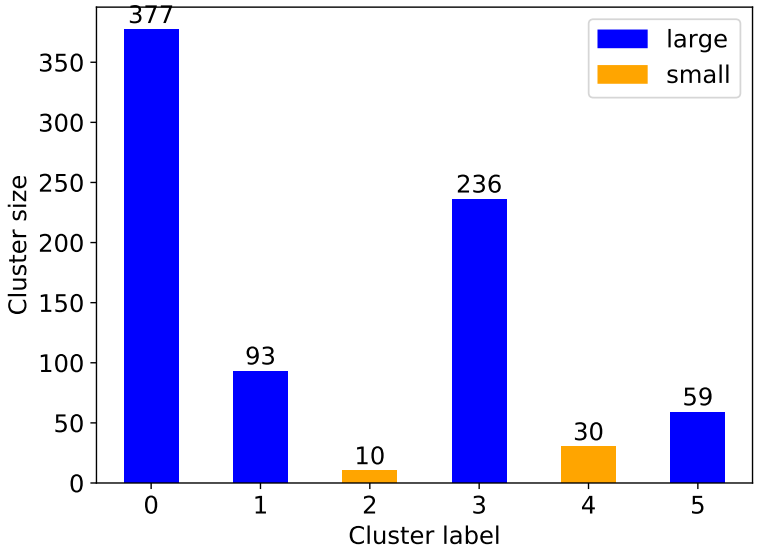
(c) $k = 10$



(d) $k = 18$

Fonte: O Autor

Figura 4.6: Tamanho dos clusters classificados como grandes e pequenos pelo FindCBLOF



Fonte: O Autor

A implementação do FindCBLOF utilizada é a presente na biblioteca PyOD: A Python Toolbox for Scalable Outlier Detection, desenvolvida por Zhao, Nasrullah and Li (2019), que fornece implementações robustas e escaláveis para vários algoritmos de detecção de anomalias.

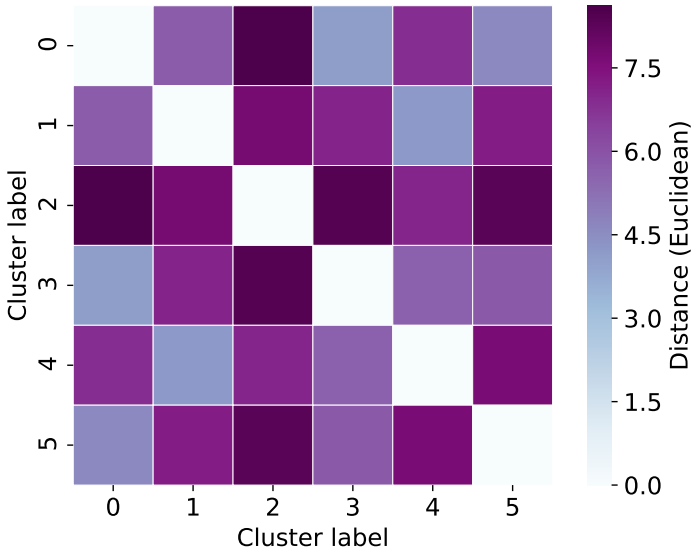
Com os valores de α igual a 0.9, e β igual a 5, conforme sugerido inicialmente por He, Xu and Deng (2003), obtiveram-se os conjuntos SC e LC mostrados na Figura 4.6, utilizando-se 6 clusters, de acordo com a análise do valor k para o K-Means, realizada anteriormente.

Para observar a separação entre os clusters gerados, utilizou-se um *heatmap*, apresentado na Figura 4.7, onde em cada eixo contém os rótulos dos clusters, e a distância Euclidiana entre os centroides dos clusters é codificada em uma escala de cores – quanto mais escura a cor, maior a distância entre os centroides dos clusters. Destaca-se que o cluster de rótulo 2 – um cluster classificado como pequeno, de tamanho 10 – é o que encontra-se mais distante dos demais. Assim, pode-se esperar que as instâncias pertencentes a este cluster sejam identificadas como anomalias. Porém, deve-se considerar que, no cálculo do fator de anomalia, o tamanho do cluster é multiplicado pela distância até o centroide. Logo, um cluster muito pequeno pode não gerar fatores de anomalia altos o suficiente para que a instância seja rotulada como anômala.

Um parâmetro importante da implementação do FindCBLOF é o fator de conta-

minação, que define o limite a partir do qual instâncias serão consideradas anômalas. Por exemplo, um fator de contaminação igual a 0.1 significa que 10% das instâncias do conjunto de dados serão consideradas anomalias, ou seja, os 10% maiores fatores CBLOF. Este parâmetro é explorado na avaliação da introdução de erros artificiais, conforme detalhado na seção seguinte.

Figura 4.7: Distância Euclidiana entre os centroides dos clusters



Fonte: O Autor

4.5 Introdução Artificial de Erros

A não disponibilidade dos registros de quais instâncias do conjunto de dados apresentaram erros de preenchimento, em auditorias anteriores, impõe, não só a aplicação de técnicas de aprendizado não supervisionado, como também a definição de uma abordagem para avaliar a capacidade de tais técnicas em identificar erros de preenchimento.

A introdução artificial de erros consiste em alterar o valor de preenchimento de uma questão manualmente, considerando-se o valor anterior como correto e o valor alterado como um erro de preenchimento. Para isso, realizou-se o processo de introdução de erros em 4 etapas, uma para cada tipo de questão, separadamente. Em cada etapa, inicia-se selecionando uma amostra aleatória de instâncias do conjunto de dados. Em seguida, para cada instância, escolhe-se, também de maneira aleatória, um atributo que pertença ao tipo de questão da etapa em consideração. Então, substitui-se o valor atual

do atributo da instância por um novo valor, pertencente ao intervalo de valores possíveis para a questão. Por fim, aplica-se os algoritmos K-Means e FindCBLOF, e verifica-se se o erro introduzido foi detectado.

O conjunto de dados do FRC possui 805 instâncias. Por limitações de tempo de execução, selecionou-se, aleatoriamente, 80 instâncias (aproximadamente 10%) do conjunto de dados para a introdução dos erros.

5 RESULTADOS

Os resultados apresentados a seguir, foram divididos de acordo com a estratégia de introdução artificial de erros no conjunto de dados do FRC, detalhada no capítulo anterior, onde erros são introduzidos individualmente em cada instância de uma amostra, por tipo de questão. O tamanho da amostra é de 80 instâncias. Ao selecioná-las, verifica-se a presença de instâncias sem nenhuma resposta para determinado tipo questão, antes de adicioná-las ao conjunto de teste. Isto é necessário para as questão do tipo valor inteiro, por exemplo. Existem 4 questões do tipo valor inteiro no questionário do FRC. Porém, uma é desconsiderada, pois trata-se de um campo de identificação do participante do ELSA-Brasil, validado pela Plataforma Otus. Assim, para as 3 questões restantes, existe uma regra de pulo no questionário, permitindo que elas não sejam respondidas . Dessa forma, várias instâncias não possuem resposta para estas questões, sendo necessário desconsiderá-las na composição do conjunto para testes.

Nas tabelas que seguem, a *Contaminação* refere-se ao percentual das instâncias com os maiores fatores de anomalia, atribuídos pelo algoritmo FindCBLOF, que serão consideradas anômalas. Por exemplo, executar o FindCBLOF com o parâmetro de contaminação igual à 1%, significa que as 8 instâncias (1% do conjunto de dados) com maior fator de anomalia serão rotuladas como anômalas. No contexto deste trabalho, significaria que estas 8 instâncias são potenciais erros de preenchimento.

A coluna *Erros detectados* representa a quantidade dos erros introduzidos que foram detectados, isto é, a instância alterada foi rotulada como anômala. Por exemplo, considerando-se que o processo de introdução de erro e aplicação do FindCBLOF foi executado 80 vezes. Se em 9 das vezes o FindCBLOF rotular a instância alterada como anômala, o número de erros detectados será 9, bem como a *Cobertura* será de 11%. Além disso, também apresentam-se os rótulos dos clusters aos quais as instâncias alteradas pertencem.

A execução de um algoritmo de detecção de anomalias e verificação do percentual de instâncias que pertencem à classe rara ou anômala, é um dos modos de avaliar o quão bom é o algoritmo para o conjunto de dados (AGGARWAL; YU, 2001). Se a detecção funcionar corretamente, espera-se valores altos de cobertura.

5.1 Questões do tipo seleção única

Para introdução de erros em questões do tipo seleção única, selecionou-se 80 das 805 instâncias do conjunto de dados. Cada uma das 80 instâncias tiveram seu valor original alterado para uma outra opção de seleção, e então o algoritmo FindCBLOF foi executado, com diferentes valores para o fator de contaminação, conforme apresentado na Tabela 5.1.

Percebe-se que com o aumento do fator de contaminação, a cobertura aumenta significativamente, porém não o suficiente para detectar todos os erros introduzidos. O melhor resultado foi de 44% de cobertura, com 30% de contaminação. Observa-se, também, que as instâncias cujos erros foram detectados pertencem aos clusters de rótulos 0 e 3, que são os dois maiores clusters gerados pelo K-Means. Assim, nota-se o peso do fator local do tamanho do cluster na determinação das anomalias.

Tabela 5.1: Detecção de erros após alterações em questões do tipo seleção única

Contaminação (número de instâncias)	**Erros detectados**	**Clusters com erros detectados**	**Clusters com erros não detectados**	**Cobertura**
1% (8)	0 (80)		{0,1,2,3,5}	0%
5% (40)	9 (80)	{3}	{0,1,2,3,5}	11%
10% (80)	16 (80)	{0,3}	{0,1,2,3,5}	20%
15% (120)	20 (80)	{0,3}	{0,1,2,3,5}	25%
20% (161)	28 (80)	{0,3}	{0,1,2,3,5}	35%
25% (201)	29 (80)	{0,3}	{0,1,2,3,5}	36%
30% (241)	35 (80)	{0,3}	{0,1,2,3,5}	44%

Fonte: O Autor

5.2 Questões do tipo checkbox

A introdução de erros em questões do tipo checkbox deu-se também a partir de uma amostra de 80 instâncias das 805 disponíveis. Para cada instância, inverteu-se o valor original de uma opção de marcação: se possuía valor 0 (desmarcada), alterou-se para 1 (marcada), e vice-versa. A tabela 5.2 apresenta a quantidade de erros detectados pelo algotitmo FindCBLOF conforme aumenta-se o fator de contaminação.

Para este tipo de questão, a cobertura foi mais baixa, alcançando apenas 16%, com contaminação igual a 30%. Percebe-se que as instâncias alteradas e não detectadas foram atribuídas aos clusters de rótulos 0, 1, 4 e 5. O cluster 4 (tamanho 10) é um

cluster classificado como pequeno pelo FindCBLOF, e os clusters 1 (tamanho 93) e 5 (tamanho 59), embora grandes, são bem menores que o cluster 0 (tamanho 377). Isto mostra que, as alterações nos valores das questões, promovem uma alteração na distância entre a instância e o centroide cuja magnitude não é grande o suficiente para sobrepor o fator de anomalia das instâncias que pertencem aos clusters maiores.

Tabela 5.2: Detecção de erros após alterações em questões do tipo checkbox

Contaminação (número de instâncias)	**Erros detectados**	**Clusters com erros detectados**	**Clusters com erros não detectados**	**Cobertura**
1% (8)	0 (80)		{0,1,4,5}	0%
5% (40)	0 (80)	{0}	{0,1,4,5}	0%
10% (80)	2 (80)	{0}	{0,1,4,5}	2%
15% (120)	2 (80)	{0}	{0,1,4,5}	2%
20% (161)	4 (80)	{0}	{0,1,4,5}	5%
25% (201)	9 (80)	{0}	{0,1,4,5}	11%
30% (241)	13 (80)	{0}	{0,1,4,5}	16%

Fonte: O Autor

5.3 Questões do tipo data

Para as questões do tipo data, da mesma forma que nos tipos anteriores, selecionou-se 80 das 805 instâncias do conjunto de dados. A alteração do valor original da questão é realizada, primeiramente, obtendo-se os valores mínimos e máximos de data para a questão. Após, escolhe-se um valor de data diferente, mas dentro do intervalo entre valores mínimo e máximo, visando atribuir um valor que seja factível em uma situação real de erro de preenchimento.

Os valores de cobertura alcançados foram melhores que os das questões de checkbox, mas não superaram os das questões de seleção única. A cobertura máxima ficou em 32%, dado o fator de contaminação igual a 30%, conforme a Tabela 5.3.

Observa-se, novamente, que as instâncias cujo erro foi detectado encontram-se nos 3 maiores clusters gerados pelo K-Means, reforçando o fator local do tamanho do cluster para detecção do erro.

Tabela 5.3: Detecção de erros após alterações em questões do tipo data

Contaminação (número de instâncias)	Erros detectados	Clusters com erros detectados	Clusters com erros não detectados	Cobertura
1% (8)	2 (80)	{3}	{0,1,2,3,4,5}	2%
5% (40)	4 (80)	{0,3}	{0,1,2,3,4,5}	5%
10% (80)	12 (80)	{0,3}	{0,1,2,3,4,5}	15%
15% (120)	16 (80)	{0,3}	{0,1,3,4,5}	20%
20% (161)	22 (80)	{0,3}	{0,1,2,3,4,5}	27%
25% (201)	23 (80)	{0,3}	{0,1,3,4,5}	29%
30% (241)	26 (80)	{0,1,3}	{0,1,3,4,5}	32%

Fonte: O Autor

5.4 Questões do tipo valor inteiro

A introdução de erros nas questões do tipo valor inteiro difere-se das demais. O motivo principal é a pequena quantidade de questões desse tipo no FRC. As 3 questões presentes no conjunto de dados, foram criadas com regras de salto entre elas, de modo que nunca serão respondidas simultaneamente, e ainda, muitas instâncias não possuem respostas para elas, também em função dos saltos. Dessa forma, o conjunto de instâncias para seleção e introdução de erros neste tipo de questão, é bem restrito: apenas 73 de 805. Assim, procedeu-se com a introdução de erros individuais, em cada uma das instâncias, de modo semelhante às questões do tipo data: identificando os valores mínimo e máximo, e escolhendo um novo dentro do intervalo.

O resultado foi bastante ruim, com nenhum erro sendo detectado. Pela Tabela 5.4, pode-se observar que as instâncias alteradas, e não detectadas, pertencem aos clusters 4 e 5, segundo e terceiro menores clusters.

Tabela 5.4: Detecção de erros após alterações em questões do tipo valor inteiro

Contaminação (número de instâncias)	Erros detectados	Clusters com erros detectados	Clusters com erros não detectados	Cobertura
1% (8)	0 (73)		{4,5}	0%
5% (40)	0 (73)		{4,5}	0%
10% (80)	0 (73)		{4,5}	0%
15% (120)	0 (73)		{4,5}	0%
20% (161)	0 (73)		{4,5}	0%
25% (201)	0 (73)		{4,5}	0%
30% (241)	0 (73)		{4,5}	0%

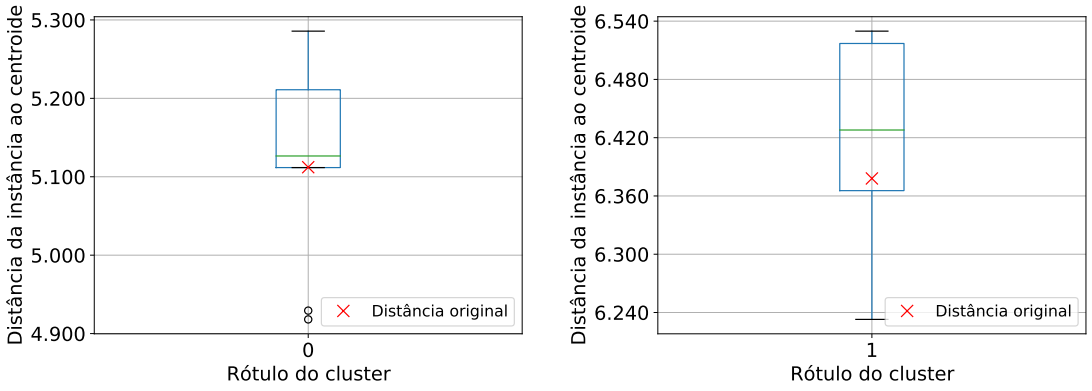
Fonte: O Autor

Para investigar o motivo da baixa detecção dos erros introduzidos, analisou-se como a alteração no valor de uma questão afeta o fator de anomalia gerado pelo algoritmo

FindCBLOF. O fator de anomalia de uma instância é calculado através da multiplicação do tamanho do cluster ao qual a instância pertence – se este for um cluster grande, caso contrário considera-se o cluster grande mais próximo – e a distância entre a instância e o centroide do cluster. Nesse sentido, selecionou-se a instância com maior fator de anomalia em cada cluster, e alterou-se cada uma de suas questões, separadamente. Então, após cada alteração, mediu-se a distância da instância ao centroide do cluster. A Figura 5.1 apresenta um boxplot com as distâncias medidas, para cada uma das instâncias. O marcador 'x' no boxplot representa a distância original da instância ao centroide, antes das alterações nas questões. Pode-se observar que a magnitude das alterações nas distâncias é pequena, se comparada ao tamanho dos clusters. Por exemplo, na Figura 5.1c a maior distância obtida com as alterações, em relação a distância original, é inferior a 0.1.

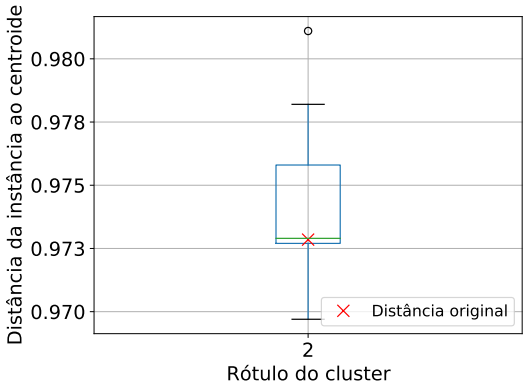
Este comportamento pode ser explicado pelo número de dimensões do conjunto de dados. Com 304 atributos, a distância Euclidiana é pouco afetada ao alterar-se o valor de uma única dimensão. Além disso, a magnitude das distâncias é menor que a do tamanho dos clusters. Dessa forma, a componente de tamanho do cluster tem maior influência. Esta influência é ainda maior na presença de clusters muito grandes no conjunto de dados. As instâncias pertencentes aos clusters grandes tendem a ter os maiores fatores de anomalia, tornando as modificações de valores, aplicadas na introdução artificial de erros, insuficientes para alterar significativamente o fator de anomalia. Com isto, a capacidade de detecção da metodologia adotada é baixa.

Figura 5.1: Distância da instância ao centroide conforme alteração dos valores das questões

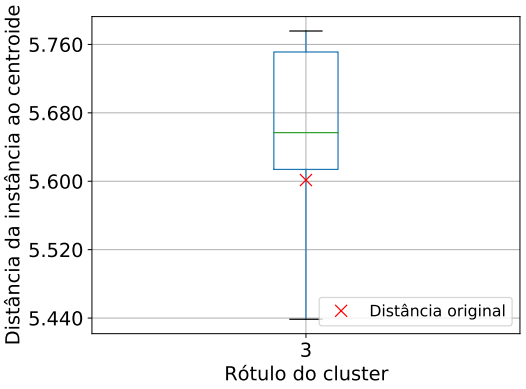


(a) Instância pertencente ao cluster 0

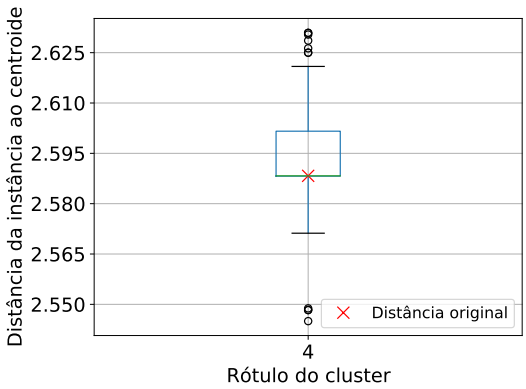
(b) Instância pertencente ao cluster 1



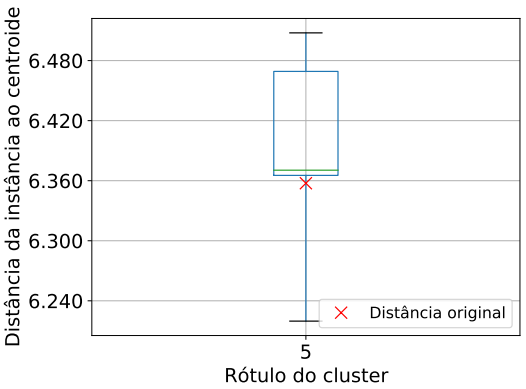
(c) Instância pertencente ao cluster 2



(d) Instância pertencente ao cluster 3



(e) Instância pertencente ao cluster 4



(f) Instância pertencente ao cluster 5

Fonte: O Autor

6 CONCLUSÃO

Em estudos epidemiológicos, a qualidade dos dados coletados é essencial para a validade dos resultados. Nesse sentido, o ELSA-Brasil adota variados mecanismos de controle de qualidade. Entre eles, encontra-se a auditoria dos questionários aplicados. O preenchimento do questionário FRC caracteriza-se pela entrada de dados, por um médico especialista, a partir de diversos registros, geralmente em papel, de exames e dados clínicos de um participante. Esta atividade é propensa a erros de preenchimento. A auditoria consiste na seleção de uma amostra dos questionários preenchidos, seguida de uma verificação manual do correto preenchimento dos dados na Plataforma Otus.

A partir da não disponibilidade dos registros de auditorias anteriores – durante o desenvolvimento deste trabalho – aplicaram-se duas técnicas de aprendizado não supervisionado: o algoritmo de agrupamento K-Means e o algoritmo de detecção de anomalias FindCBLOF. O conjunto de dados do FRC passou por um pré-processamento, que consiste na imputação de valores faltantes, codificação de atributos categóricos, transformação de atributos numéricos e a normalização dos valores. Também aplicou-se a análise do Silhouette Score – medida de validação de agrupamentos – para a escolha do número de clusters do K-Means.

Para avaliar a capacidade dos algoritmos escolhidos em detectar erros de preenchimento, realizou-se o processo de introdução de erros de preenchimento. Este processo consiste na seleção de uma amostra de questionários do FRC, seguida da alteração do valor original de preenchimento de uma questão, para cada questionário da amostra. Após, é contabilizado quantas das instâncias com o valor original alterado foram apontadas como anômalas pelo algoritmo FindCBLOF.

Com base nos resultados apresentados no capítulo anterior, conclui-se que a utilização do algoritmo de agrupamento K-Means, em conjunto com o algoritmo de detecção de anomalias FindCBLOF, não apresenta capacidade de detecção de erros maior que 44%, considerando erros de preenchimento individuais em questões de seleção única do FRC. Para os demais tipos de questão, os resultados são inferiores, com o pior caso sendo a não detecção de nenhum erro para as questões do tipo valor inteiro. Observou-se que o tamanho do cluster exerce grande influência no cálculo do fator de anomalia, enquanto que as alterações nas distâncias de uma instância ao centroide do cluster, produzidas pela introdução dos erros, não são suficientes para elevar o fator de anomalia e, assim, identificar a instância como anômala.

Ressalta-se que a não disponibilidade dos registros de erros em auditorias anteriores, durante o desenvolvimento deste trabalho, foi um fator limitante para a escolha das técnicas de aprendizado de máquina. Contudo, sugere-se a exploração de outras técnicas de detecção não supervisionada de anomalias, baseadas em densidade, em modelos probabilísticos, entre outras disponíveis nas bibliotecas utilizadas neste trabalho. Uma vez que após a introdução artificial de erros sabe-se quais instâncias possuem o valor de um atributo alterado, poderia-se explorar técnicas de aprendizado semi-supervisionado. Sugere-se também a investigação de conjuntos de dados de outros questionários *offline* do ELSA-Brasil.

Dessa forma, estudos mais abrangentes são necessários para identificar uma melhor abordagem ao problema de auxiliar no processo de auditoria dos questionários do ELSA-Brasil, através da detecção automatizada de erros de preenchimento.

REFERÊNCIAS

AGGARWAL, C. C.; YU, P. S. Outlier detection for high dimensional data. In: ACM. **ACM Sigmod Record**. [S.l.], 2001. v. 30, n. 2, p. 37–46.

ARBELAITZ, O. et al. An extensive comparative study of cluster validity indices. **Pattern Recognition**, Elsevier, v. 46, n. 1, p. 243–256, 2013.

BIRNBAUM, B. **Algorithmic approaches to detecting interviewer fabrication in surveys**. [S.l.]: University of Washington, 2012.

BOLTON, R. J.; HAND, D. J. et al. Unsupervised profiling methods for fraud detection. **Credit scoring and credit control VII**, Citeseer, p. 235–255, 2001.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM computing surveys (CSUR)**, ACM, v. 41, n. 3, p. 15, 2009.

CHOR, D. et al. Questionário do ELSA-Brasil: desafios na elaboração de instrumento multidimensional. **Revista de saúde pública**, SciELO Public Health, v. 47, p. 27–36, 2013.

FERREIRA, D. R. **Sistema para predição de respostas de questionários aplicados a Plataforma Otus**. [S.l.]: Universidade Federal do Rio Grande do Sul. Não publicado, 2018.

HE, Z.; XU, X.; DENG, S. Discovering cluster-based local outliers. **Pattern Recognition Letters**, Elsevier, v. 24, n. 9-10, p. 1641–1650, 2003.

JAIN, A. K. Data clustering: 50 years beyond k-means. **Pattern recognition letters**, Elsevier, v. 31, n. 8, p. 651–666, 2010.

LIMA-COSTA, M. F.; BARRETO, S. M. Tipos de estudos epidemiológicos: conceitos básicos e aplicações na área do envelhecimento. **Epidemiologia e serviços de saúde**, Coordenação-Geral de Desenvolvimento da Epidemiologia em Serviços/Secretaria . . . , v. 12, n. 4, p. 189–201, 2003.

MARSLAND, S. **Machine learning: an algorithmic perspective**. [S.l.]: Chapman and Hall/CRC, 2014.

MINISTÉRIOS, E. dos. ELSA Brasil: maior estudo epidemiológico da América Latina. **Rev Saúde Pública**, SciELO Public Health, v. 43, n. 1, 2009.

MITCHELL, T. M. **Machine learning**. [S.l.]: McGraw hill, 1997.

ROUSSEEUW, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal of computational and applied mathematics**, Elsevier, v. 20, p. 53–65, 1987.

SCHMIDT, M. I. et al. Estratégias e desenvolvimento de garantia e controle de qualidade no elsa-brasil. **Revista de Saúde Pública**, SciELO Public Health, v. 47, p. 105–112, 2013.

SCIKITLEARN. **K-Means clustering**. 2019. Acessado: 20/11/2019. Available from Internet: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans>.

SCIKITLEARN. **Selecting the number of clusters with silhouette analysis on KMeans clustering**. 2019. Acessado: 16/11/2019. Available from Internet: <https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html>.

XU, D.; TIAN, Y. A comprehensive survey of clustering algorithms. **Annals of Data Science**, Springer, v. 2, n. 2, p. 165–193, 2015.

ZHAO, Y.; NASRULLAH, Z.; LI, Z. Pyod: A python toolbox for scalable outlier detection. **Journal of Machine Learning Research**, v. 20, n. 96, p. 1–7, 2019.