

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Victor Castro Silva Finzi

**Gestão do Conhecimento usando Mineração de
Dados em Avaliação de Pós-ocupação**

Uberlândia, Brasil

2017

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Victor Castro Silva Finzi

**Gestão do Conhecimento usando Mineração de Dados
em Avaliação de Pós-ocupação**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, como parte dos requi-
sitos exigidos para a obtenção título de Ba-
charel em Ciências da Computação.

Orientador: Maria Adriana Vidigal de Lima

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2017

Victor Castro Silva Finzi

Gestão do Conhecimento usando Mineração de Dados em Avaliação de Pós-ocupação

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, como parte dos requi-
sitos exigidos para a obtenção título de Ba-
charel em Ciências da Computação.

Trabalho aprovado. Uberlândia, Brasil, 21 de Dezembro de 2017:

Maria Adriana Vidigal de Lima
Orientador

Cláudio Camargo Rodrigues

Anilton Joaquim da Silva

Uberlândia, Brasil
2017

Resumo

As tecnologias da informação vêm sendo largamente utilizadas para facilitar o processo da gestão do conhecimento, que abrange a geração, o armazenamento, a distribuição e a análise dos dados, devido ao grande aumento no volume de dados nas diversas áreas de negócios e pesquisas científicas. A descoberta de conhecimento em bases de dados é uma metodologia que possibilita a gestão do conhecimento e a mineração de dados faz parte desta metodologia. Neste contexto, este trabalho propõe o estudo da base de dados sobre o morar e a identificação de algoritmos apropriados de mineração de dados que possam auxiliar na gestão do conhecimento armazenado sobre a pós-ocupação em habitações. A base de dados sobre o morar armazena avaliações de pós-ocupação propostas e aplicadas pela equipe do laboratório MORA da Faculdade de Arquitetura e Urbanismo da Universidade Federal de Uberlândia.

Palavras-chave: Mineração de dados, Descoberta de conhecimento em bases de dados, Avaliação de Pós-ocupação.

Lista de ilustrações

Figura 1 – Visão geral do processo de Descoberta de Conhecimento em base de dados	13
Figura 2 – Exemplo de Base de transações	20
Figura 3 – Árvore inicial da estrutura <i>FP-tree</i>	21
Figura 4 – Árvore final da estrutura <i>FP-tree</i>	22
Figura 5 – Pseudo código da função FP-Tree (MARIANO, 2011)	23
Figura 6 – Pseudo código da função FP-Growth (MARIANO, 2011)	23
Figura 7 – Lista de drivers para conexão do RapidMiner com bancos de dados . .	24
Figura 8 – Tela de exibição da ferramenta RapidMiner	25
Figura 9 – Trecho de um questionário de Avaliação de Pós-ocupação	28
Figura 10 – Exemplo de questões para Avaliação de Pós-ocupação	28
Figura 11 – Dados selecionados na ferramenta MySQL	29
Figura 12 – Dados da tabela de resposta na ferramenta RapidMiner	30
Figura 13 – Operadores de extração dos dados e seleção de atributos	30
Figura 14 – Tela de configuração dos parâmetros do operador Select Attributes . .	31
Figura 15 – Tela com os dados selecionados na primeira etapa do processo de etl . .	31
Figura 16 – Aplicação do operador Generate Concatenation no fluxo do processo . .	32
Figura 17 – Tela de parâmetros do operador de concatenação	32
Figura 18 – Resultado da aplicação do operador de concatenação	33
Figura 19 – Segunda aplicação do operador de concatenação	33
Figura 20 – Resultado da aplicação do segundo operador de concatenação	33
Figura 21 – Fluxo do processo com o operador Generate Attributes	34
Figura 22 – Configuração da coluna resp com a sua função de geração de atributo .	34
Figura 23 – Tela de resposta após o operador de geração de atributo ser executado .	35
Figura 24 – Fluxo do processo com a segunda etapa da aplicação da seleção de atributos	35
Figura 25 – Tela de resposta após aplicação do segundo operador de seleção de atributos	36
Figura 26 – Fluxo do processo com o operador Pivot	36
Figura 27 – Resultado da aplicação do operador Pivot	37
Figura 28 – Fluxo do processo com o operador Replace Missing Values	37
Figura 29 – Tela com os resultados após a aplicação do fluxo do processo da Figura 28 .	38
Figura 30 – Fluxo do processo de ETL com o operador Numerical to Binominal . . .	38
Figura 31 – Tela da configuração dos parâmetros do operador Numerical to Binominal .	39
Figura 32 – Resultados da aplicação do operador Numerical to Binominal	39
Figura 33 – Etapa final do processo de ETL com o operador Set Role	39

Figura 34 – Tela de parâmetros do operador Set Role	40
Figura 35 – Tela com os resultados após a aplicação do operador Set Role	40
Figura 36 – Etapa com o operador FP-Growth no processo de mineração de dados .	41
Figura 37 – Tela da configuração de parâmetros do operador FP-Growth	42
Figura 38 – Tela dos itens frequentes encontrados pelo operador FP-Growth	42
Figura 39 – Etapa final do processo de mineração de dados com o operador Create Association Rules	43
Figura 40 – Tela de configuração dos parâmetros do operador Create Association Rules	44
Figura 41 – Tela das regras de associações encontradas na etapa final do processo de mineração	44
Figura 42 – Apresentação do grafo gerado pelo operador Create Association Rules .	45
Figura 43 – Visualização das regras de associação criadas no formato $A \rightarrow B$	45
Figura 44 – Regras de associação encontradas no final do processo de mineração de dados	47

Lista de abreviaturas e siglas

APO	Avaliação de pós-ocupação
KDD	Knowledge Discovery in Databases
DM	Data Mining
FAUeD	Faculdade de Arquitetura e Urbanismo e Design
ETL	Extraction Transformation and Load
DW	Data Warehouse

Sumário

1	INTRODUÇÃO	8
1.1	Objetivos	9
1.2	Justificativas	9
1.3	Organização do Trabalho	10
2	METODOLOGIA	11
3	FUNDAMENTAÇÃO TEÓRICA	13
3.1	KDD - Knowledge Discovery in Databases	13
3.2	Extração, Transformação e Carga	14
3.3	Data Mining	16
3.4	Algoritmos de Associação	17
3.4.1	Regras de associação	18
3.4.2	FP-Growth	20
3.5	Ferramenta RapidMiner	24
3.6	Trabalhos Correlatos	25
4	DESENVOLVIMENTO	27
4.1	Avaliação de Pós-Ocupação em Moradias	27
4.2	ETL	29
4.3	Data Mining	40
5	ANÁLISE DOS RESULTADOS	46
6	CONCLUSÃO	49
	REFERÊNCIAS	51

1 Introdução

Antes da ascensão dos computadores, um dos maiores problemas para as pessoas e organizações era o armazenamento de dados e informações, como e onde estes dados seriam armazenados. Nas últimas décadas com a evolução das estruturas computacionais de armazenamento como os bancos de dados, e o baixo custo dos hardwares, o problema de armazenamento de dados vem sendo estudado e melhorado, possibilitando que as organizações pudessem, cada vez mais, armazenar grandes quantidades de dados.

Com a era digital, a quantidade de dados produzidos pelas organizações, tornou-se gigantesca, este acontecimento gerou o seguinte problema: uma grande quantidade de dados que gera pouca informação. A partir deste problema, as organizações e pesquisadores buscaram desenvolver métodos para encontrar padrões, relações e associações destes dados, ou seja, informações úteis para as empresas. Então foi criada uma técnica na área de computação, denominada de mineração de dados (do inglês *Data Mining*) que gera a informação que é tão valiosa para as empresas.

Com o conhecimento proveniente da mineração de dados, as empresas começaram a basear suas tomadas de decisão fundamentando-se nos resultados das análises dos dados minerados. O conhecimento obtido a partir desta técnica ajuda várias áreas a tomarem decisões mais acertadas. Por exemplo, na área de medicina, pode-se melhorar a precisão dos diagnósticos. Outras áreas como sistema eleitoral, bancos, telemarketing e segurança podem se beneficiar da aplicação de técnicas de mineração de dados.

A mineração de dados está inserida dentro de um processo muito maior que é o de descoberta de conhecimento em uma base de dados (ou KDD - *Knowledge Discovery in Databases*). Dentro deste processo existem etapas que vão desde a seleção dos dados em seu estado bruto até o final que é a obtenção do conhecimento das relações entre os dados. A mineração de dados é apenas uma das etapas de todo o processo, e que está inserida no meio dele. O KDD surgiu do interesse de três áreas diferentes: estatística, banco de dados e inteligência artificial.

Este trabalho propõe o estudo da base de dados sobre o morar e a aplicação de um algoritmo de mineração de dados apropriado para a gestão do conhecimento neste banco de dados. A base de dados sobre o morar mantém os resultados de diversas pesquisas de avaliação de pós-ocupação (APO) em habitações criadas e aplicadas pela equipe do laboratório MORA da Faculdade de Arquitetura e Urbanismo da Universidade Federal de Uberlândia. A aplicação do algoritmo de mineração de dados sobre os dados coletados de qualidade em moradias será utilizada dentro do processo do KDD, gerando informações que serão úteis para a equipe do laboratório MORA. Para a aplicação do algoritmo de

mineração de dados será utilizada uma ferramenta computacional, denominada RapidMiner (HOFFMANN; KLINKENBERG, 2013).

A avaliação pós-ocupação é um processo sistematizado para a avaliação de habitações, depois de algum tempo de sua construção e ocupação, realizada para aferir a qualidade dos ambientes ocupados, considerando os contextos funcional, comportamental e ambiental. A APO tem como objetivo evidenciar a importância que a qualidade da moradia exerce no bem estar de seus usuários. Um dos instrumentos que possibilita a verificação da satisfação do morador é a aplicação de um questionário, representando um formulário amplo capaz de aumentar a compreensão da realidade da moradia por parte dos usuários. O questionário segue um roteiro de perguntas relativas às características socioeconômicas do morador, aspectos gerais da habitação, conforto ambiental dos cômodos, acessibilidade, segurança, equipamentos existentes, atividades realizadas nos espaços, aspectos ambientais e hábitos sustentáveis. A base de dados sobre o morar mantém organizadas as perguntas, seus conceitos, atributos e qualificadores, e armazena as respostas fornecidas pelos moradores.

1.1 Objetivos

Este trabalho tem como objetivo estudar todo o processo de KDD (*Knowledge Discovery in Database*) e o algoritmo FP-Growth para *Data Mining*, para que seja aplicado na gestão do conhecimento da base de dados sobre o morar. Ao final do processo deseja-se analisar e comparar as informações obtidas, gerando conhecimento e informação sobre a APO. Para isto, será necessário:

- Analisar e entender os dados brutos contidos na base de dados, e quais informações se deseja obter a partir deles.
- Aplicar os procedimentos contidos nas etapas de pré-processamento, de mineração de dados e de pós-processamento. Estes procedimentos serão aplicados após a seleção de quais dados serão minerados, utilizando o algoritmo de mineração de dados denominado FP-Growth (HAN; PEI; YIN, 2000).
- Analisar e comparar os resultados obtidos após a fase de pós-processamento ter sido finalizada.

1.2 Justificativas

A mineração de dados é um ramo da ciência da computação, que vem crescendo e se destacando no âmbito da geração de conhecimento a partir de bases de dados. A necessidade de gerar informações úteis para as organizações, empresas e pessoas, a partir

de grandes quantidades de dados que, no seu estado bruto não representam e não fornecem informação alguma, motivou vários pesquisadores a desenvolverem técnicas baseadas em estatística, banco de dados e inteligência artificial, para resolverem este problema.

Este trabalho visa o conhecimento e o estudo destas técnicas, para que sejam empregadas em uma base de dados obtida através da APO (Avaliação de Pós-ocupação) em moradias, de forma a proporcionar a geração de novas informações, resultantes das análises sobre os dados. O conhecimento decorrente da mineração de dados pode ser utilizado para a melhoria da qualidade dos projetos de moradia, visto que os dados coletados pelas APOs são relativos à forma de morar, considerando: características dos moradores, informações da moradia, do entorno e hábitos sustentáveis.

1.3 Organização do Trabalho

Este trabalho está dividido em 6 capítulos. O primeiro capítulo aborda de uma maneira geral a contextualização do problema das grandes massas de dados, e fornece uma breve explicação sobre mineração de dados, descoberta de conhecimento e avaliação de pós ocupação, os objetivos e justificativas do trabalho.

No segundo capítulo apresenta a metodologia utilizada, definindo especificamente como o trabalho foi desenvolvido, de acordo com os requisitos e regras de negócios da APO. O terceiro capítulo apresenta toda a fundamentação teórica e explicita os conceitos iniciais de KDD, a extração dos dados, a mineração, exemplificando o uso da técnica de regras de associação considerando o algoritmo FP-Growth (HAN; PEI; YIN, 2000) e a ferramenta utilizada para desenvolver o processo de mineração.

O quarto capítulo apresenta o desenvolvimento do trabalho, explicando como e de onde os dados foram extraídos, e a geração das informações que este trabalho propõe a encontrar. No quinto capítulo é feita a análise dos resultados que o processo de mineração de dados gerou, discutindo a aplicação das regras de associação. O sexto e último capítulo as conclusões que foram obtidas após inúmeras análises e estudos feitos sobre os dados.

2 Metodologia

Os métodos utilizados para a realização deste trabalho foram definidos de acordo com os fundamentos e as etapas do KDD. Para a aplicação do KDD no contexto da avaliação de pós-ocupação em moradias, definiu-se os seguintes passos:

1. Realizar pesquisa documental sobre o tema KDD, e entender como funciona cada uma das etapas:
 - Seleção: Como a mineração de dados é aplicada a uma grande base de dados, seleciona-se um conjunto menor de dados e mais relevantes para serem tratados.
 - Pré-processamento: Os dados são tratados e padronizados de acordo com as atividades responsáveis por esta parte do processo.
 - Transformação: As ferramentas de mineração de dados exigem certos tipos de formato para os dados, nesta etapa aplica-se a transformação destes dados para o padrão exigido pela ferramenta.
 - Mineração de dados: Aplica-se as técnicas e algum algoritmo de mineração de dados, gerando informação.
 - Pós-processamento: O usuário final analisa e extrai conhecimento das informações geradas no passo anterior, a fim de verificar se o objetivo final foi atingido.
2. Pesquisar sobre os algoritmos de mineração de dados, entender como as áreas de estatística, banco de dados e inteligência artificial foram utilizadas para a criação da mineração de dados, estudar os algoritmos de regras de associação, notadamente o algoritmo FP-Growth, para que os dados das APOs possam ser analisados.
3. Estudar a ferramenta RapidMiner para que a mesma possa ser utilizada como instrumento de apoio à este trabalho. Estudar os mecanismos que ela fornece para a preparação e a análise dos dados, considerando o algoritmo FP-Growth e os parâmetros que podem ser ajustados para a aplicação da mineração de dados.
4. Efetuar a análise dos dados APO (Avaliação de Pós Ocupação). Os dados a serem analisados são respostas à diversos questionários relativos à forma de morar dos habitantes de uma determinada região, considerando informações dos moradores, da construção física da moradia, do entorno, da área comum (no caso de condomínios) e de hábitos sustentáveis.
5. Analisar e interpretar as regras de associação obtidas após a aplicação do algoritmo de associação, gerando a conclusão sobre os padrões interessantes. As respostas

às questões da APO são tabulares e a análise dos dados, a partir de regras de associação, permite identificar padrões de respostas comuns, que podem auxiliar na compreensão do resultado geral das APOs.

3 Fundamentação Teórica

3.1 KDD - Knowledge Discovery in Databases

O processo de descoberta de conhecimento em uma base de dados (KDD do inglês *knowledge discovery in databases*) teve origem em 1989, baseado nos problemas gerados pelas grandes quantidades de dados armazenados pelas organizações, houve a necessidade de se extrair informações úteis. Com as técnicas e análise de dados utilizadas na época, não era possível encontrar padrões e associações dentro do grande volume de dados. O KDD foi desenvolvido com a participação de pesquisadores de diversas áreas como estatística, banco de dados e inteligência artificial, com o intuito de desenvolverem ferramentas mais eficazes para lidar com estes dados. Podemos definir KDD como: "um processo não trivial na identificação de padrões nos dados válidos, novos, úteis e entendíveis" (FAYYAD et al., 1996). A Figura 1 apresenta uma visão geral das etapas do processo de KDD.

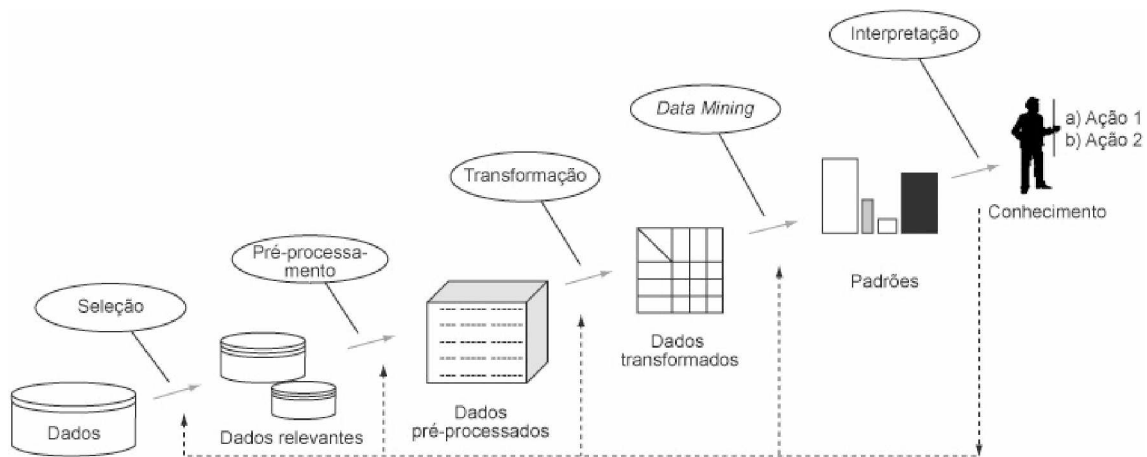


Figura 1 – Visão geral do processo de Descoberta de Conhecimento em base de dados

O processo de descoberta de conhecimento inicia-se baseado no modelo de negócios e nos dados, que serão posteriormente analisados. É composto de três passos principais, pré-processamento, mineração de dados e pós-processamento, sendo eles interativos e iterativos.

Iniciando-se pela etapa de seleção dos dados, anterior à de pré-processamento, os dados que serão selecionados farão parte de um subconjunto de informações válidas a serem processadas em seguida. O entendimento desse subconjunto é importante, pois sua escolha errada podera gerar conhecimento inválido ao final do processo (GOMES, 2016).

Na etapa de pré-processamento, após a seleção, os dados serão tratados e preparados para a fase de mineração, de acordo com a padronização gerada. Nesta fase

são realizadas atividades como gerenciamento do banco de dados, limpeza, verificação semântica, remoção de dados duplicados, tratamento e enriquecimento de dados. O pré-processamento pode ser a etapa mais demorada e trabalhosa dentro de todos os processos de descoberta de conhecimento, devido as diversas formas que os dados podem ser armazenados e selecionados.

Os dados pré-processados seguem para a etapa de transformação, onde serão formatados de acordo com a ferramenta utilizada, servindo de entrada para o processamento. Esta formatação para o padrão correto leva à uma maior eficiência no processamento. Já na etapa de mineração de dados deve-se inicialmente decidir o tipo de análise a ser utilizada. Entre os possíveis tipos de análise, podem-se destacar a sumarização, regressão, agrupamento e regras de associação. Na fase final, de pós-processamento, as informações geradas serão analisadas e interpretadas, possibilitando a geração de conhecimento. Desta forma, o usuário final pode se beneficiar deste conhecimento para as tomadas de decisão organizacionais.

Todas as etapas são fundamentais para o processo de descoberta de conhecimento, porém deve-se ressaltar a importância da etapa de mineração de dados, que concentra a metodologia de construção do conhecimento pretendido.

3.2 Extração, Transformação e Carga

Com a grande quantidade de informações que são geradas pelas empresas nos dias de hoje, ficou impraticável continuar utilizando papéis e arquivos manuais para a análise dessas informações. Em consequência, os bancos de dados começaram a ser utilizados para solucionar a falta de organização e demora das análises manuais. Além de armazenar e gerenciar os dados, surge a necessidade de transformar estes dados em conhecimento, para que as organizações possam, a partir de seus próprios dados, conhecer melhor seus processos, clientes. Neste contexto, as bases de dados têm grande importância e a utilização dos dados ali contidos pode ser ainda maximizada.

Os dados que são utilizados no processo de KDD na maioria das vezes não estão no formato certo para que possam ser processados na etapa de DM, para isto é necessário que a etapa de pré-processamento seja aplicada. O ETL (*Extraction, Transformation and Load*) é o processo responsável pela extração dos dados de suas fontes primárias, pela limpeza e transformação dos mesmos, colocando-os no formato correto e carregando para a ferramenta de mineração um conjunto de dados adequado e harmonizado.

Em projetos de DW (*Data Warehouse*) o ETL é utilizado para extrair e integrar dados de diversas fontes e carregar as informações no DW (OLIVEIRA, 2009 apud COSTA, 2009). No meio deste processo pode ser aplicada a transformação/limpeza dos dados, não sendo obrigatória, mas a aplicação do tratamento dos dados viabiliza a me-

lhoria da qualidade dos dados quando já estão no DW. Para isto podem ser utilizadas diversas ferramentas como Oracle Data Integrator, Microsoft Integration Server, SAP Business Objects Data Services, e outras diversas ferramentas.

Antes de partimos para a seleção dos dados, é analisado quais são os objetivos que se quer alcançar com o processo, ao conhecer as metas que devem ser atingidas, é realizada a etapa de seleção dos dados que serão utilizados nos processos posteriores. Em diversos casos temos muitos bancos de dados que possuem diversas tabelas e integrações com outros bancos com diferentes informações, isto pode ser um problema na hora de selecionar quais tabelas e atributos serão utilizados, tornando a seleção uma tarefa difícil.

Após a seleção dos dados entramos nas etapas de limpeza e transformação, aplicando diferentes técnicas para a limpeza dos dados, podemos tratar os dados ausentes, retirar os registros duplicados e incompletos, correção dos dados inconsistentes, isto pode ser feito simplesmente substituindo estes valores ou excluindo-os. A etapa de transformação consiste em modificar os dados para um padrão que seja compreendido pelo algoritmo de mineração, por exemplo transformar dados numéricos para binomiais. (OLIVEIRA, 2009 apud COSTA, 2009).

De acordo com (HAN; KAMBER; PEI, 2012) existem vários tipos de estratégias a serem aplicadas na etapa de transformação dos dados, estas estratégias são:

- Suavização: está estratégia retira os ruídos dos dados, aplicando técnicas de binning, regressão e agrupamento.
- Novos atributos: são criados atributos para ajudar na etapa de mineração de dados.
- Agregação: utilizado para gerar um conjunto de dados agregados para serem analisados.
- Normalização: redimensiona os valores dos dados dentro de um intervalo numérico, como por exemplo de 0 a 1.
- Discretização: altera-se os valores brutos de um atributo numérico por valores dentro de um intervalo, ou são substituídos por rótulos conceituais.
- Geração de hierarquia para dados nominais: Generaliza a hierarquia de atributos nominais através de conceitos de nível superior.

Após a aplicação das estratégias de transformação dos dados, é finalizada a etapa de ETL e então os dados estarão preparados para serem processados na etapa de mineração.

3.3 Data Mining

Pode-se definir *Data Mining* como o conjunto de técnicas organizadas, que agem sobre grandes massas de dados, tornando útil e recente o que antes era ignorado. Os pesquisadores que criaram a área de mineração de dados, tiveram sua motivação baseada nos problemas de escalabilidade pois os algoritmos devem escalar de acordo com o tamanho do conjunto de dados crescente, a alta dimensionalidade devido a grande quantidade de atributos encontradas nos conjuntos de dados, dados complexos, distribuição de dados (quando os dados estão armazenados em diferentes repositórios e locais) e análise não tradicional (TAN; STEINBACH; KUMAR, 2009).

A mineração de dados não é um processo simples, é composto por diferentes áreas de estudo, como estatística, inteligência artificial, aprendizagem de máquina, banco de dados, computação paralela e distribuída. Considerada um dos principais processos na descoberta de conhecimento, sua definição ainda é confusa e considerada um sinônimo do KDD. A mineração de dados fornece diversas alternativas para analisar dados, sendo elas, classificação, estimativa ou regressão, associação, clusterização ou agrupamento (KIRA, 2016).

1. Classificação

A classificação visa identificar a qual classe um determinado item pertence. Para este fim, são utilizados os diversos atributos que compõem o banco de dados. Dentre eles, decide-se quais podem ser utilizados pelo algoritmo de classificação, de acordo com as propriedades qualitativas do conhecimento a ser descoberto: compreensibilidade, validade e grau de interesse. Um dos atributos deve ser utilizado como atributo meta, e outros são escolhidos como atributos previsores. Assim, a partir destes atributos, os itens são classificados. Esta técnica pode ser utilizada para entender como os dados existentes na base se comportam e prever o comportamento de novos itens inseridos na base (PHYU, 2009).

2. Estimativa ou Regressão

Utilizando métodos numéricos e estatísticos, a estimativa tem como objetivo determinar o valor de um índice a partir de dados passados ou semelhantes. Para realizá-la são utilizados algoritmos de redes neurais e regressão, de forma que possa ser determinado o valor do índice. Pode ser utilizada para estimar, por exemplo: número de filhos ou renda total de uma família e demanda de um consumidor para um novo produto. A estimativa pode ser usada para auxiliar na execução de uma tarefa de classificação, sendo que diferentes intervalos de valores contínuos, referentes à atributos meta ou atributos previsores, devem corresponder a diferentes classes do problema em questão (CUNHA; ARAUJO; SANTOS, 2003).

3. Associação

A tarefa de associação tem como objetivo correlacionar os objetos de uma grande base de dados, identificando padrões interessantes a serem analisados posteriormente pelo usuário final. Para que a tarefa identifique uma relação entre os dados, os mesmos devem estar em um padrão formalizado como $A \rightarrow B$, onde A e B são conjuntos de itens. Um exemplo que motivou o desenvolvimento desta técnica é a análise da cesta de mercado, que tem como objetivo identificar se existe uma relação nos produtos que os clientes de um supermercado comprem, identificando estes produtos a equipe de marketing pode tomar decisões que levem ao supermercado aumentar suas vendas (AMO, 2006).

4. Clusterização ou Agrupamento

Parecida com a classificação, a clusterização tem como objetivo dividir os dados em grupos de características parecidas, que não sejam semelhantes aos objetos dos outros grupos, porém não tem prévio conhecimento das categorias que serão criadas. Aqui as classes serão determinadas posteriormente, pois serão o resultado final da mineração. A diferença fundamental entre a formação de agrupamento e a classificação é que no agrupamento não existem classes predefinidas para classificar os registros em estudo (CORTES; PORCARO; LIFSCHITZ, 2002).

Para a execução das tarefas do *Data Mining*, utilizam-se técnicas de mineração de dados que são compostas por algoritmos específicos e softwares desenvolvidos especificamente para esta área. Os algoritmos estudados neste trabalho serão os baseados em regras de associações. A técnica de regras de associação busca encontrar relações entre os dados. Composta por duas etapas, os dados serão analisados para encontrar os itens mais frequentes, e em seguida, tais itens são utilizados para a geração das regras de associação.

3.4 Algoritmos de Associação

A tarefa de associação tem como objetivo correlacionar os objetos de uma grande base de dados, identificando padrões interessantes a serem analisados posteriormente pelo usuário final (AGRAWAL; IMIELINSKI; SWAMI, 1993). Esta tarefa talvez seja uma das mais importantes e mais simples, e é utilizada para correlacionar os objetos de uma grande base de dados, identificando padrões que sejam interessantes a serem analisados posteriormente pelo usuário final.

Um exemplo clássico da técnica de associação e aplicação prática é a análise da cesta de compras (*Market Basket Analysis*), definida num cenário de um supermercado com inúmeras compras realizadas pelos seus clientes (GIUDICI; FIGINI, 2009). O conjunto de todas essas compras é armazenado numa base de dados gerando várias transa-

ções, sendo que cada transação é composta pelos produtos que os clientes compraram. Porém, o armazenamento destes dados por si só, não indica informações que possam ser efetivamente utilizadas e aplicadas em alguma estratégia de marketing ou vendas do supermercado, seja pelo gerente ou direção.

Neste contexto, pode-se desejar a maximização das vendas do supermercado, e para isso, surge a questão de como esta tarefa pode ser realizada a partir do conhecimento do comportamento dos clientes, registrado de forma bruta na base de dados transacional. Para responder à esta pergunta, podem-se aplicar regras de associação sobre estes dados. Em meio à milhões de compras no banco de dados do supermercado, esta técnica permite descobrir combinações de itens que mais aparecem dentro do banco de transações. A a frequência acima do normal dos padrões encontrados gera uma informação valiosa que pode ser considerada na tomada de decisão do gerente.

As regras de associação utilizam análises estatísticas para que sejam confiáveis. Um exemplo prático de uma regra encontrada na análise da cesta de compras, pode ser: (leite, pão) $\rightarrow$ (manteiga). Esta associação indica que uma determinada porcentagem de clientes que comprem leite e pão, utilizando um determinado grau de certeza também comprem manteiga. Uma regra de associação é escrita desta forma para que qualquer pessoa possa compreender e tirar conclusões com um grau de certeza confiável.

Com o objetivo final de encontrar relações e padrões em transações de um banco de dados, esta tarefa vem sendo uma das mais utilizadas ao longo dos anos em diferentes tipos de problemas como em análise de logs de servidores web e sistemas de recomendação (GONCALVES, 2013). As regras de associação se tornaram grandes aliadas na tomada de decisão de empresas, dando respostas confiáveis baseadas em métodos estatísticos garantido uma assertividade nas decisões escolhidas.

A subseção 3.4.1 detalha e formaliza os conceitos gerais utilizados na tarefa de construção de regras de associação como *itemset*, *confiança*, *suporte*, entre outros conceitos que são necessários para garantir a confiança da regra gerada, e apresenta o algoritmo que foi utilizado e estudado na aplicação para criar as regras de associação na base de dados APO, o algoritmo FP-Growth.

3.4.1 Regras de associação

Uma regra de associação nada mais é que a existência de um conjunto de itens dentro de uma transação implicando na existência de algum outro conjunto de itens. Para que possam ser identificados e avaliados os padrões interessantes, devem-se definir alguns conceitos básicos para a compreensão do funcionamento destas regras e de como as mesmas são analisadas.

As regras são obtidas a partir de um banco de dados transacional, ou seja, a análise

é feita sobre as transações presentes no banco de dados. Utilizando-se o exemplo da cesta de compras no banco de dados do supermercado, cada compra é uma transação única que pode conter diversos produtos diferentes em uma única compra. Assim, todos os itens comprados pelo cliente compõem uma transação que é chamada de *itemset*.

A avaliação dos padrões interessantes é realizada a partir dos seguintes conceitos de regra de associação:

- Seja I um conjunto de itens tais que $I = \{i_1, i_2, \dots, i_n\}$ e T um banco de dados com m transações, $T = \{t_1, t_2, \dots, t_n\}$ onde cada transação possui um TID (identificador da transação). Um conjunto de itens S é chamado de *itemset* e é um subconjunto não vazio de I , onde S está contido em I , ou seja todos os itens de uma transação, e um *itemset* composto por k itens é chamado de *k-itemset*.
- Uma regra de associação é uma expressão da forma $A \rightarrow B$, onde A e B são *itemsets*, e significa que as transações que contém o *itemset* A podem conter também o *itemset* B , sendo A o antecedente e B o consequente.
- A cada *itemset* é atribuído um valor denominado suporte, o suporte do *itemset* é a porcentagem de transações à qual este *itemset* pertence. Por exemplo, para uma regra de associação $A \rightarrow B$ o valor do suporte é:

$$\text{sup}(A \rightarrow B) = \text{quantidade de transações que contém } A \cup B / \text{quantidade total de transações } T.$$

- Determina-se que um *itemset* é frequente a partir de um valor mínimo atribuído ao suporte min_sup , então todos os *itemsets* que tiverem um valor maior ou igual ao min_sup são considerados *itemsets* frequentes.
- Para cada regra de associação é aplicado um determinado grau de confiança. Este valor indica a força da regra de associação: para uma regra de associação $A \rightarrow B$, o seu grau de confiança é definido como:

$$\text{confiança}(A \rightarrow B) = \text{sup}(A \cup B) / \text{sup}(A),$$

ou seja, a porcentagem das transações de $A \cup B$ sobre todas as transações T que possuem A . Também é definido um valor mínimo de confiança para as regras de associação, denominado min_conf , que é utilizado para saber se uma regra é forte.

A tarefa de encontrar as regras de associação consiste em duas etapas conforme [RapidMiner \(2014\)](#) explica:

1. A primeira etapa consiste em encontrar todos os *itemsets* frequentes da base de dados. Considerando-se um banco de transações que possui n itens, tem-se uma

análise de $2^n - 1$ conjuntos de itens, ou seja todas as combinações possíveis dos itens excluindo-se o conjunto vazio. Esta etapa utiliza o valor do suporte mínimo para eliminar os *itemsets* que não são frequentes nas transações, e a exclusão desses *itemsets* não frequentes ajuda na análise apenas dos *itemsets* interessantes.

2. A segunda etapa estabelece a criação das regras de associação a partir dos *itemsets* frequentes encontrados na primeira etapa. Nesta segunda etapa a confiança é utilizada para definir a força da regra, e caso a regra tenha um valor maior ou igual ao valor mínimo de confiança estipulado, então esta regra de associação será considerada. Para um banco de transações com n itens podem ser geradas $[3^n] - [2^n] + 1$ regras de associação.

Por exemplo, para um conjunto de itens $I = \{A, B, C\}$ podem ser encontrados 7 *itemsets* frequentes e 10 regras de associação conforme descrito nas etapas 1 e 2. Com essas duas etapas definidas, de uma forma geral, a tarefa de regra de associação servirá para encontrar os *itemsets* frequentes utilizando a propriedade de que se um *itemset* é frequente então todos os subconjuntos deste *itemset* também são frequentes, excluindo-se os *itemsets* não frequentes de acordo com o valor do suporte mínimo.

3.4.2 FP-Growth

O algoritmo FP-growth é uma alternativa para encontrar os *itemsets* frequentes, baseando-se numa estrutura de dados mais eficiente e compacta no formato de uma árvore chamada *FP-tree* (*Frequent Pattern tree*), construída com o objetivo de que não seja necessário varrer a base de dados inúmeras vezes. O algoritmo FP-Growth pode ser dividido em duas fases principais: a primeira consiste em construir a *FP-tree* e a segunda fase utiliza as informações que foram criadas na *FP-tree* para computar os *itemsets* frequentes de acordo com o valor do suporte.

Para detalhar como a *FP-tree* é construída, a Figura 2 apresenta um exemplo simples de uma base de dados contendo 3 itens e 3 transações, sendo $I = \{A, B, C\}$. Cada linha da tabela representa uma transação que agrupa um conjunto de itens.

TID	ITENS	ORDENAÇÃO
1	A B C	B C A
2	B C	B C
3	A	A

Figura 2 – Exemplo de Base de transações

Conforme explicado em [RapidMiner \(2014\)](#), inicialmente o algoritmo da *FP-tree* define a contagem da quantidade de cada item em relação a quantidade de transações,

ou seja, faz o cálculo do suporte de cada um dos itens. Em seguida, são ordenados os itens de cada transação de forma decrescente de acordo com a sua contagem ou valor de suporte. Esta ordenação é utilizada para facilitar a formação dos caminhos da *FP-tree*. Esta ordenação pode ser vista na terceira coluna da tabela da Figura 2. Em seguida, a segunda fase do algoritmo define o início da construção da árvore a partir da primeira transação, criando os nós a partir do nó nulo, conforme Figura 3:

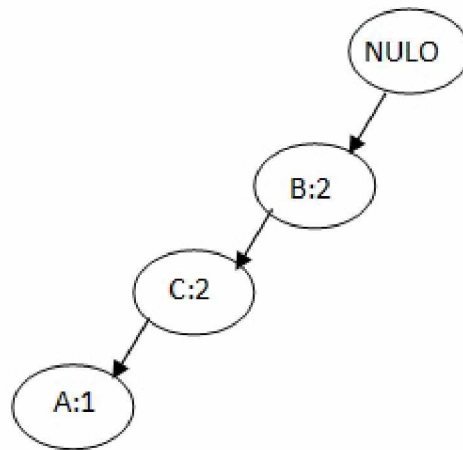
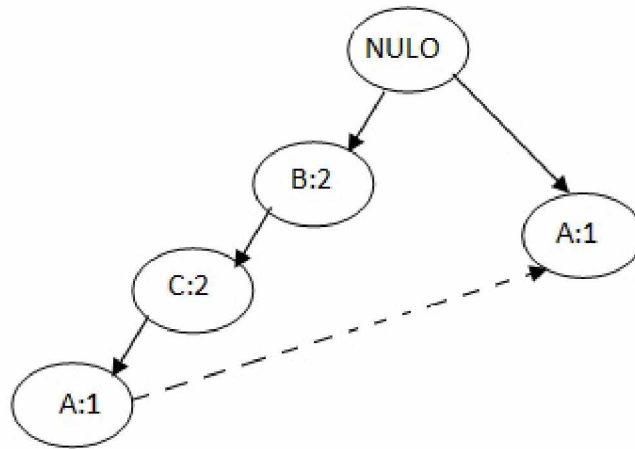


Figura 3 – Árvore inicial da estrutura *FP-tree*

Cada nó da árvore possui o seu item com um número que indica a quantidade de transações daquele item naquele caminho. A primeira transação define o caminho $B \rightarrow C \rightarrow A$, com a contagem de 1 para cada item. Na segunda transação, utiliza-se o caminho já existente $B \rightarrow C$ e apenas é incrementada a contagem dos itens B e C. Os caminhos gerados a partir das transações 1 e 2 estão ilustrados na Figura 3.

Para terceira transação contida na base de dados, o algoritmo cria outro caminho para o item A, pois esta transação não pertence ao caminho $B \rightarrow C \rightarrow A$. Para nós que estão relacionados ao mesmo item, porém estão em caminhos diferentes, estes devem ser ligados por uma linha tracejada. A Figura 4 apresenta a árvore final para as 3 transações.

Figura 4 – Árvore final da estrutura *FP-tree*

A construção da árvore é construída enquanto existem transações a serem consumidas. Em consequência, a árvore estará mais comprimida de acordo com a quantidade de transações que passam por um mesmo caminho. A próxima etapa do algoritmo FP-Growth analisa os caminhos que foram gerados, extraíndo os *itemsets* frequentes.

Finalizada a primeira fase de construção da *FP-tree*, o algoritmo começa a realizar a extração dos itens frequentes, utilizando uma abordagem de baixo para cima. Os itens menos frequentes estão nos nós folha, pois a construção da árvore baseia-se na ordenação de acordo com os valores de suporte. No exemplo da Figura 4 o algoritmo começa pelo nó menos frequente que seria o (A:1), de acordo com o seu suporte. Caso o item não passe no critério do suporte mínimo, ele deve ser ignorado e retirado da árvore. Em seguida, o percurso na árvore continua, na mesma direção (folhas-raiz) e o próximo nó a ser visitado é o próximo menos frequente. Portanto, no final do processamento do algoritmo para este exemplo, restará na árvore apenas o conjunto de itens frequentes {B,C}, representado pelo caminho $B \rightarrow C$.

As Figuras 5 e 6 apresentam as fases 1 e 2, respectivamente, do pseudo-código para o algoritmo do FP-Growth. A fase 1 é responsável pela construção da árvore *FP-tree* a ser utilizada na mineração dos dados, e a fase 2 gerencia o percurso na *FP-tree*, em que cada nó será avaliado, e possivelmente podado. A árvore resultante revela os caminhos para os padrões mais relevantes da base de dados utilizada no processo de descoberta de conhecimento.

```

1.  Entrada: Uma base de dados  $D$  e o valor de suporte mínimo  $min\_sup$ .
2.  Saida: O conjunto  $L$  com todos os itemsets frequentes.
// Fase 1 - Construção da FP-Tree
3.  Função FP-Tree( $D$ ,  $min\_sup$ )
4.  percorra a base de dados  $D$  uma vez;
5.  determine o conjunto de itens frequentes  $F$  e seus suportes;
6.  ordene  $F$  em ordem decrescente em função do suporte e chamá-la de  $L$ ;
7.  crie a raiz da FP-Tree  $T$  e coloque como “null”;
8.  para cada transação  $t$  em  $D$  faça
9.      selecione e ordene os itens frequentes em  $t$  de acordo com a ordem
        de  $L$ , tornando a lista de itens frequentes em  $t$  igual a  $[p|P]$ , onde  $p$  é o
        primeiro elemento e  $P$  é o resto da lista;
10.     execute insere_tree( $[p|P]$ ,  $T$ ).
11.     se  $T$  tiver um filho  $N$  em que  $N.nome\_item = p.nome\_item$  então
12.         incremente o contador de  $N$  por em 1;
13.     senão
14.         crie um novo nó  $N$ , e inicie seu contador com 1;
15.         ligue o seu parent-link a  $T$ , e seu node-link aos nós de mesmo
            (nome_item) através da estrutura dos node-links;
16.     fim se
17.     se  $P$  não for vazio então
18.         chame insere_tree( $P$ ,  $N$ ) recursivamente;
19.     fim se
20. fim para

```

Figura 5 – Pseudo código da função *FP-Tree* (MARIANO, 2011)

```

// Fase 2 - Mineração da FP-Tree
21. Função FP-Growth( $Tree$ ,  $\alpha$ )
22. se  $Tree$  contém apenas um caminho  $P$  então
23.     para cada combinação  $\beta$  de nós no caminho  $P$  faça
24.         gere o padrão  $\beta \cup \alpha$  com suporte =  $min\_sup$  dos nós em  $\beta$ ;
25.     fim para
26. senão
27.     para cada  $a_i$  na tabela de node-links de  $Tree$  faça
28.         gere o padrão  $\beta = a_i \cup \alpha$  com suporte =  $a_i.suporte$ ;
29.         construa a base de padrões condicionada de  $\beta$  e crie
            a FP-Tree condicionada de  $\beta$  chamada de  $Tree_\beta$ ;
30.         se ( $Tree_\beta \neq \emptyset$ ) então
31.             FP-growth( $Tree_\beta$ ,  $\beta$ );
32.         fim se
33.     fim para
34. fim se

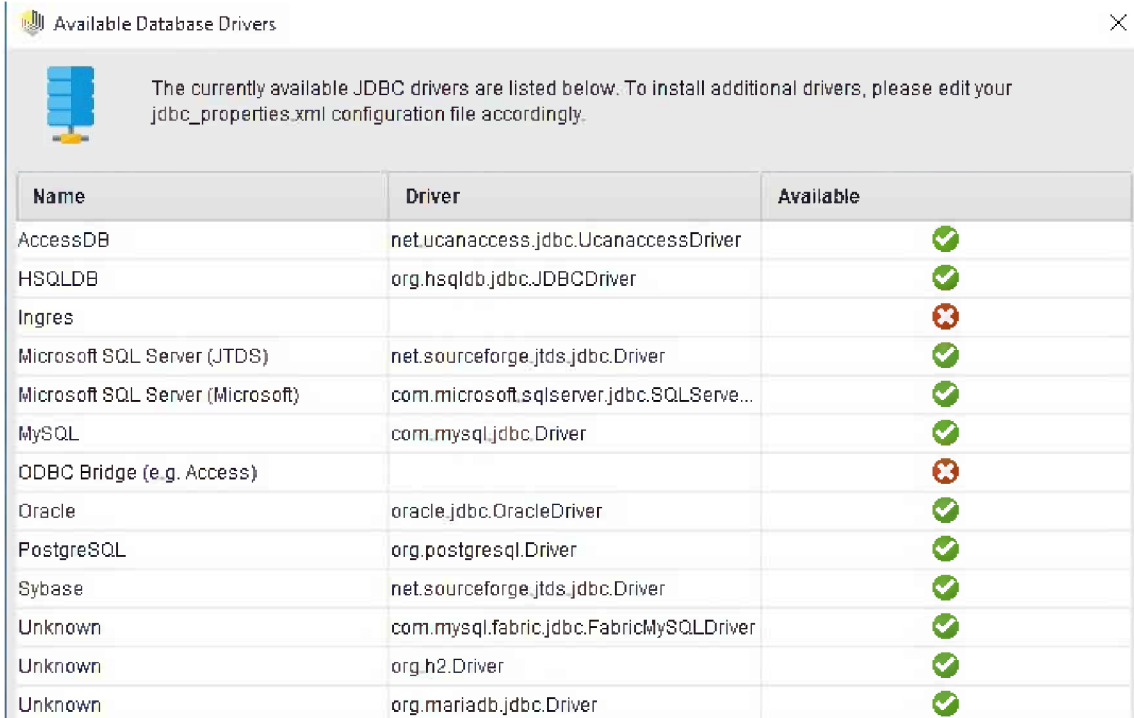
```

Figura 6 – Pseudo código da função *FP-Growth* (MARIANO, 2011)

3.5 Ferramenta RapidMiner

Nos dias de hoje já existem várias ferramentas desenvolvidas com o intuito de tornar melhor a experiência do usuário com a mineração de dados, algumas delas são o Inteligente Miner, MineSet, Clementine, DBMiner, e todas essas ferramentas tem algum destaque em suas funcionalidades (AMO, 2006). A ferramenta RapidMiner possui uma interface simples e de fácil compreensão e além de ser gratuita para fins acadêmicos, oferece uma grande quantidade de algoritmos que podem ser aplicados em diversas técnicas. Estes motivos levaram à escolha da utilização desta ferramenta neste trabalho.

Começando pela extração dos dados, o Rapidminer oferece várias opções no carregamento de informações, são várias extensões aplicadas a diferentes tipos de bancos de dados como MySQL, Oracle, PostgreSQL entre outros. Planilhas em Excel também podem ser utilizadas para a extração das informações no Rapidminer, e serviços em nuvem também são oferecidos. Por exemplo para a extração de dados do banco MySQL é feita uma conexão com o driver do MySQL e são especificadas e configuradas as portas do servidor MySQL. A Figura 7 apresenta os drivers de conexão com banco de dados que são oferecidos pelo RapidMiner.



Name	Driver	Available
AccessDB	net.ucanaccess.jdbc.UcanaccessDriver	✓
HSQLDB	org.hsqldb.jdbc.JDBCDriver	✓
Ingres		✗
Microsoft SQL Server (JTDS)	net.sourceforge.jtds.jdbc.Driver	✓
Microsoft SQL Server (Microsoft)	com.microsoft.sqlserver.jdbc.SQLServe...	✓
MySQL	com.mysql.jdbc.Driver	✓
ODBC Bridge (e.g. Access)		✗
Oracle	oracle.jdbc.OracleDriver	✓
PostgreSQL	org.postgresql.Driver	✓
Sybase	net.sourceforge.jtds.jdbc.Driver	✓
Unknown	com.mysql.fabric.jdbc.FabricMySQLDriver	✓
Unknown	org.h2.Driver	✓
Unknown	org.mariadb.jdbc.Driver	✓

Figura 7 – Lista de drivers para conexão do RapidMiner com bancos de dados

O RapidMiner dispõe de vários documentos técnicos, vídeos, treinamentos e tutoriais para a familiarização com a ferramenta, explicando desde o começo do processo de mineração ou análise de dados, até o fim que seria a análise do pós-processamento e a visualização dos resultados, que podem ser aplicados em gráficos por exemplo.

Na parte de desenvolvimento do processo de mineração, são oferecidos diversos operadores para trabalhar no pré-processamento, operadores que realizam a seleção, limpeza e transformação dos dados. No processo de mineração dos dados o Rapidminer também oferece inúmeros algoritmos para todas as técnicas como regras de associação, classificação, regressão, clusterização, etc. Para todos os operadores da ferramenta, são oferecidos parâmetros de configuração e explicação dos detalhes sobre eles e como devem ser utilizados.

Com um design agradável e interativo, durante a finalização de um processo de mineração, o RapidMiner apresenta a análise dos dados de uma forma simples e objetiva para o usuário. A Figura 8 apresenta a interface gráfica do RapidMiner com um processo contido no tutorial do algoritmo FP-Growth como exemplo:

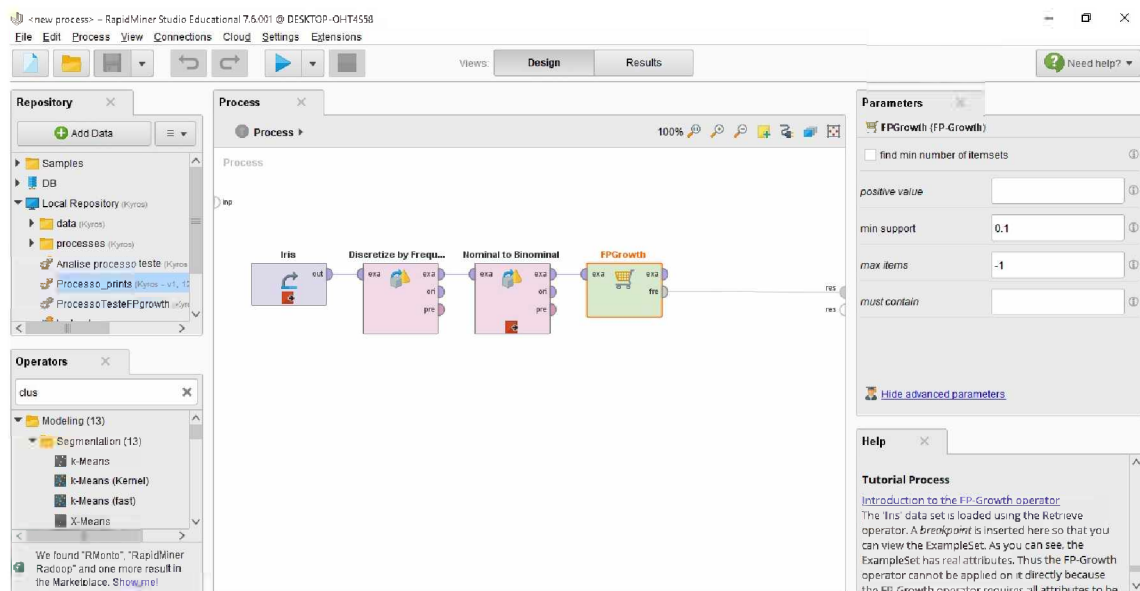


Figura 8 – Tela de exibição da ferramenta RapidMiner

3.6 Trabalhos Correlatos

O trabalho de Nandi et al. (2015) apresenta a utilização do algoritmo FP-Growth em uma ferramenta de mineração de dados chamada Shell Orion Data Mining Engine com uma base de dados que contém informações de pacientes com hepatite. A escolha do algoritmo FP-Growth foi feita como uma alternativa ao algoritmo Apriori. O algoritmo FP-Growth trabalha sem a geração de candidatos, comparado ao algoritmo Apriori, e faz um aproveitamento mais eficiente dos recursos computacionais como a memória.

O processamento das regras de associação é realizado na ferramenta Shell Orion e na ferramenta Weka. Em Nandi et al. (2015) os resultados são comparados utilizando-se o algoritmo FP-Growth nas duas ferramentas e as mesmas regras de associações são encontradas em ambos os casos, para as duas ferramentas. Faz-se também uma comparação

entre os tempos de processamento nas duas ferramentas, e não foi observada variação significativa entre os respectivos tempos de processamento.

Um trabalho realizado sobre uma base de dados do TSE, analisa os resultados obtidos através da aplicação de duas tarefas de mineração de dados: classificação e associação, utilizando apenas o desenvolvimento e análise da parte de regras de associação para ser comparado aos objetivos deste trabalho (KIRA, 2016). O algoritmo utilizado no trabalho de Kira (2016) foi o apriori, com uma abordagem diferente da utilizada na base da APO, o algoritmo FP-Growth, e a ferramenta Weka.

O trabalho de Kira (2016) mostra que após a identificação das regras de associação não revelou muitas novas associações relevantes. Por exemplo uma regra que diz que "advogados possuem ensino superior", e este tipo de regra é intuitiva e não é muito relevante na análise de regras de associação. Porém também foi possível identificar regras que mostram que a participação das mulheres no cenário eleitoral não é significativa. A comparação das duas tarefas de classificação e associação no trabalho de Kira (2016) demonstrou que a tarefa de classificação revelou resultados mais interessantes, e isto se deve também à forma como os dados foram tratados antes de serem minerados.

A partir dos resultados expostos em Kira (2016), observa-se que dependendo das regras de associação geradas, as mesmas não possuem um sentido sintático lógico, e que as regras e os algoritmos de associação dependem muito de como os dados estão dispostos e formatados, para que sejam encontradas regras interessantes que não seriam possíveis de serem inferidas de uma forma intuitiva.

4 Desenvolvimento

4.1 Avaliação de Pós-Ocupação em Moradias

A avaliação de pós-ocupação é uma análise realizada pela Faculdade de Arquitetura e Urbanismo e Design (FAUeD) com o intuito de coletar informações dos moradores sobre como eles avaliam as residências. A avaliação de pós-ocupação baseia-se no princípio de que ambientes edificados e espaços livres devem estar em permanente análise, considerando-se, do ponto de vista dos usuários, os aspectos construtivos, espaciais, comportamentais e ambientais (PEREIRA; ABREU; REIS, 2017).

A base de dados é constituída pelos questionários de pós-ocupação, respondida por vários moradores de diferentes tipos de habitação. Com o tempo estes questionários formam uma base de dados com várias informações que podem ser analisadas, para que possam ser encontradas correlações entre as respostas dos moradores, oferecendo à FAUeD informações que podem ser utilizadas para desenvolver e melhorar os projetos de habitação.

O intuito deste trabalho é analisar as informações e descobrir relações entre as respostas de diferentes moradores através dos questionários de pós-ocupação, utilizando a técnica de regras de associação sobre a base de dados, que será minerada para encontrar correlações entre respostas dos moradores. Um questionário de avaliação de pós-ocupação é composto por várias perguntas de diferentes tipos, como perguntas sobre o sexo do morador, escolaridade, quantidade de pessoas que moram juntas, faixa salarial, e sobre o que a pessoa acha da residência, como aspectos negativos e positivos do local, tamanho dos quartos, ventilação entre outras perguntas.

As figuras 9 e 10 mostram um exemplo do questionário que é respondido pelos moradores:

Universidade Federal de Uberlândia - Faculdade de Arquitetura e Urbanismo e Design
[MORA] PESQUISA EM HABITAÇÃO - Pesquisa de Satisfação

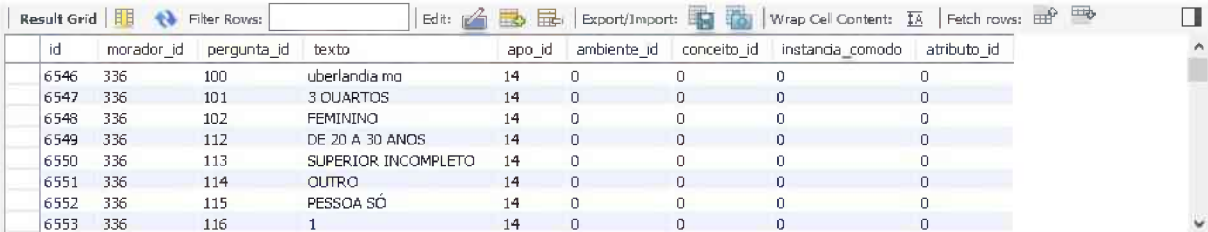
QUESTIONÁRIO - APARTAMENTO					
Nº apto:	Bloco:	Data:	Horário:		
CARACTERÍSTICAS DO ENTREVISTADO					
1. Qual seu sexo e idade? F () M ()					
() 15-20	() 20-30	() 31-40	() 41-50	() 51-60	() 61-70
2. Qual seu grau de escolaridade?					
() Nenhuma escolaridade		() Ensino fundamental incompleto		() Ensino fundamental completo	
() Ensino médio completo		() Superior incompleto		() Superior completo	
				() Pós-Graduado	
3. Qual posição você ocupa no seu grupo familiar?					
() marido	() esposa	() filho(a)	() amigo(a)	() parente	() outro
4. A qual tipo de família você pertence?					
() família nuclear (pai + mãe + filhos)		() pessoa só		() família monoparental (pai ou mãe + filhos)	
() família nuclear expandida (pai + mãe + filhos + parente)		() DINC (casal sem filhos)		() coabitação (sem vínculo familiar)	
				() outro?	
5. Qual o número total de pessoas que moram neste apartamento? _____					
6. Qual sua renda familiar em reais?					
() 1.000 - 2.000	() 2.001 - 4.000	() 4.001 - 6.000	() 6.001 - 8.000	() 8.001-10.000	() Mais que 10.000
7. Este apartamento é:					
() alugado	() próprio	() emprestado	() financiado		
8. Em que ano você e sua família se mudaram para este edifício? _____					
9. Neste apartamento trabalham:					
() funcionários mensalistas		() funcionários diaristas		() não conta com funcionários	

Figura 9 – Trecho de um questionário de Avaliação de Pós-ocupação

EQUIPAMENTOS DE USO COMUM DO EDIFÍCIO									
18. Sobre os equipamentos de uso comum, indique sua impressão em relação aos aspectos abaixo:									
	TAMANHO	APARÊNCIA	FREQUÊNCIA - USO	MOBILIÁRIO		TAMANHO	APARÊNCIA	FREQUÊNCIA - USO	MOBILIÁRIO
SALA DE FESTAS	() Ótimo	() Ótima	() 1x por mês	() Ótimo	PLAYGROUND	() Ótimo	() Ótima	() Diariamente	() Ótimo
	() Bom	() Boa	() 1x por semestre	() Bom		() Bom	() Boa	() 4-5x por semana	() Bom
	() Regular	() Regular	() 1x por ano	() Regular		() Regular	() Regular	() 2-3x por semana	() Regular
	() Ruim	() Ruim	() A cada dois anos	() Ruim		() Ruim	() Ruim	() 1x por semana	() Ruim
	() Péssimo	() Péssima	() Nunca uso	() Péssimo		() Péssimo	() Péssima	() Nunca uso	() Péssimo
PISCINA	() Ótimo	() Ótima	() 2-3x por semana	() Ótimo	ACADEMIA	() Ótimo	() Ótima	() Diariamente	() Ótimo
	() Bom	() Boa	() 1x por semana	() Bom		() Bom	() Boa	() 4-5x por semana	() Bom
	() Regular	() Regular	() 1x a cada 15 dias	() Regular		() Regular	() Regular	() 2-3x por semana	() Regular
	() Ruim	() Ruim	() 1x por mês	() Ruim		() Ruim	() Ruim	() 1x por semana	() Ruim
	() Péssimo	() Péssima	() Nunca uso	() Péssimo		() Péssimo	() Péssima	() Nunca uso	() Péssimo
QUADRA	() Ótimo	() Ótima	() 2-3x por semana	() Ótimo	SALA DE JOGOS	() Ótimo	() Ótima	() 2-3x por semana	() Ótimo
	() Bom	() Boa	() 1x por semana	() Bom		() Bom	() Boa	() 1x por semana	() Bom
	() Regular	() Regular	() 1x a cada 15 dias	() Regular		() Regular	() Regular	() 1x a cada 15 dias	() Regular
	() Ruim	() Ruim	() 1x por mês	() Ruim		() Ruim	() Ruim	() 1x por mês	() Ruim
	() Péssimo	() Péssima	() Nunca uso	() Péssimo		() Péssimo	() Péssima	() Nunca uso	() Péssimo
CHURRASQUEIRA	() Ótimo	() Ótima	() 1x por mês	() Ótimo	ESPAÇOS GOURMET	() Ótimo	() Ótima	() 1x por mês	() Ótimo
	() Bom	() Boa	() 1x por semestre	() Bom		() Bom	() Boa	() 1x por semestre	() Bom
	() Regular	() Regular	() 1x por ano	() Regular		() Regular	() Regular	() 1x por ano	() Regular
	() Ruim	() Ruim	() A cada dois anos	() Ruim		() Ruim	() Ruim	() A cada dois anos	() Ruim
	() Péssimo	() Péssima	() Nunca uso	() Péssimo		() Péssimo	() Péssima	() Nunca uso	() Péssimo
PISCINA INFANTIL	() Ótimo	() Ótima	() 1x por mês	() Ótimo	SAUNA	() Ótimo	() Ótima	() 1x por mês	() Ótimo
	() Bom	() Boa	() 1x por semestre	() Bom		() Bom	() Boa	() 1x por semestre	() Bom
	() Regular	() Regular	() 1x por ano	() Regular		() Regular	() Regular	() 1x por ano	() Regular
	() Ruim	() Ruim	() A cada dois anos	() Ruim		() Ruim	() Ruim	() A cada dois anos	() Ruim
	() Péssimo	() Péssima	() Nunca uso	() Péssimo		() Péssimo	() Péssima	() Nunca uso	() Péssimo
QUIOSQUES	() Ótimo	() Ótima	() 1x por mês	() Ótimo					
	() Bom	() Boa	() 1x por semestre	() Bom					
	() Regular	() Regular	() 1x por ano	() Regular					
	() Ruim	() Ruim	() A cada dois anos	() Ruim					
	() Péssimo	() Péssima	() Nunca uso	() Péssimo					

Figura 10 – Exemplo de questões para Avaliação de Pós-ocupação

Para realizar a mineração de dados destas informações, estes questionários foram armazenados num banco de dados MySQL. O banco é composto por várias tabelas que foram desenvolvidas na linguagem SQL, como tabela de perguntas, questões, tipos de questões, moradores, categorias, entre outras várias tabelas. A tabela principal neste trabalho é a tabela de respostas que contém as avaliações realizadas pelos moradores. Esta tabela será utilizada para realizar a mineração das informações. A Figura 11 apresenta as informações na interface do MySQL após a realização do comando select na tabela de respostas.



id	morador_id	pergunta_id	texto	apo_id	ambiente_id	conceito_id	instancia_comodo	atributo_id
6546	336	100	uberlandia mo	14	0	0	0	0
6547	336	101	3 QUARTOS	14	0	0	0	0
6548	336	102	FEMININO	14	0	0	0	0
6549	336	112	DE 20 A 30 ANOS	14	0	0	0	0
6550	336	113	SUPERIOR INCOMPLETO	14	0	0	0	0
6551	336	114	OUTRO	14	0	0	0	0
6552	336	115	PESSOA SÓ	14	0	0	0	0
6553	336	116	1	14	0	0	0	0

Figura 11 – Dados selecionados na ferramenta MySQL

4.2 ETL

O desenvolvimento do processo de mineração da base de dados da APO foi realizado na ferramenta RapidMiner com integração com o banco de dados MySQL. No MySQL temos todas as tabelas com as informações e respostas que foram gerados através do questionário da APO respondido pelos moradores.

Foi utilizada a tabela de respostas para realizar todo o processo de mineração, e para isto, utilizou-se a conexão com o servidor do banco de dados MySQL. A ferramenta Rapidminer foi utilizada para a realização do conjunto de operações da ETL, sendo a primeira delas a obtenção das informações da tabela de respostas. Uma lista de tuplas presentes da tabela de respostas pode ser vista na Figura 12.

id	morador_id	pergunta_id	texto	apo_id	ambiente_id	conceito_id	instancia_c...	atributo_id
6546	336	100	uberlandia mg	14	0	0	0	0
6547	336	101	3 QUARTOS	14	0	0	0	0
6548	336	102	FEMININO	14	0	0	0	0
6549	336	112	DE 20 A 30 A...	14	0	0	0	0
6550	336	113	SUPERIOR I...	14	0	0	0	0
6551	336	114	OUTRO	14	0	0	0	0
6552	336	115	PESSOA SÓ	14	0	0	0	0
6553	336	116	1	14	0	0	0	0
6554	336	117	ENTRE R\$10...	14	0	0	0	0
6555	336	118	PRÓPRIO	14	0	0	0	0
6556	336	119	2013	14	0	0	0	0
6557	336	120	NINGUÉM	14	0	0	0	0
6558	336	172	CASA	14	0	0	0	0
6559	336	173	PRÓPRIA	14	0	0	0	0
6560	336	174	MAIOR	14	0	0	0	0
6561	336	103	BOA	14	8	29	0	0

Figura 12 – Dados da tabela de resposta na ferramenta RapidMiner

Através do bloco **Retrieve resposta**, mostrado na Figura 13, as tuplas são obtidas da tabela respostas e carregadas para que o processo de ETL possa se iniciar. A realização de toda a etapa de pré-processamento se inicia através da seleção e da transformação dos dados para o formato que é exigido pelo algoritmo de regra de associação FP-Growth. Com o operador **Select Attributes** foram selecionados os atributos a serem minerados. Desta forma, foi possível iniciar o fluxo do processamento, ao associar operação **Retrieve resposta** com a operação **Select Attributes**, conforme ilustra a Figura 13:

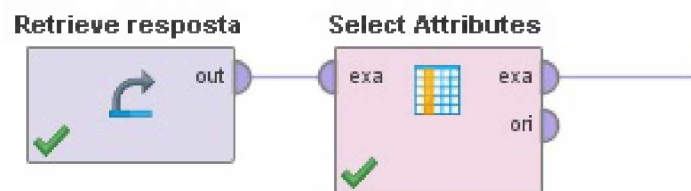


Figura 13 – Operadores de extração dos dados e seleção de atributos

O operador que seleciona os atributos (**Select Attributes**) oferece uma configuração de *subset*. Esta opção permite que um subconjunto de atributos (colunas) seja escolhido. No caso da APO, foram selecionadas as colunas: **morador_id**, **pergunta_id**, **texto** e **conceito_id** para serem minerados. A Figura 14 apresenta os parâmetros ofe-

recidos pelo RapidMiner, e a escolha da opção *subset* para a APO. É ainda possível inverter a seleção de atributos para e incluir atributos especiais.

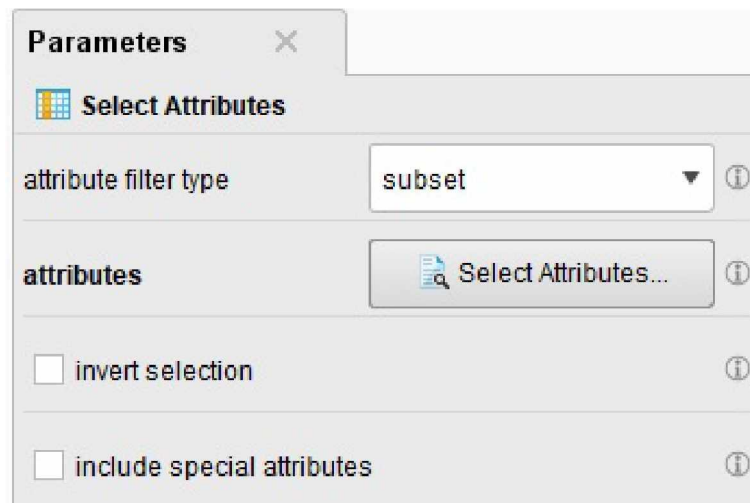


Figura 14 – Tela de configuração dos parâmetros do operador Select Attributes

O resultado destas duas operações de ETL promoveu a nova tabela, contendo a partir deste ponto apenas as colunas necessárias à análise, como mostra a Figura 15.

Row No.	morador_id	pergunta_id	texto	conceito_id
1	336	100	uberlandia mg	0
2	336	101	3 QUARTOS	0
3	336	102	FEMININO	0
4	336	112	DE 20 A 30 A...	0
5	336	113	SUPERIOR I...	0
6	336	114	OUTRO	0
7	336	115	PESSOA SÓ	0
8	336	116	1	0
9	336	117	ENTRE R\$10...	0
10	336	118	PRÓPRIO	0

Figura 15 – Tela com os dados selecionados na primeira etapa do processo de etl

Depois de selecionados os atributos, foi utilizado o operador **Generate Concatenation** para concatenar as colunas `pergunta_id` e `texto`, pois para construir os dados de uma forma que seja entendida pelo operador de mineração, a concatenação irá gerar um produto em que será dado para cada pergunta a sua resposta. Como serão geradas várias respostas para a mesma pergunta, cada pergunta com a sua resposta deve virar uma coluna, dessa forma para cada morador será dada a sua resposta para a respectiva pergunta, conforme a Figura 16.

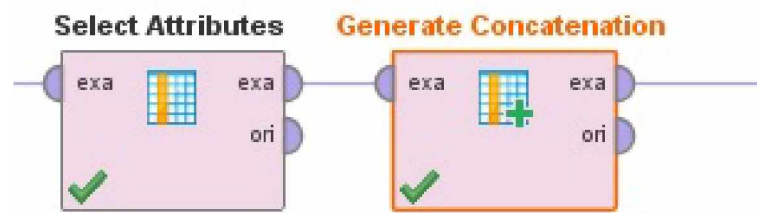


Figura 16 – Aplicação do operador Generate Concatenation no fluxo do processo

O operador **Generate Concatenation** fornece a possibilidade de escolher e configurar os parâmetros que serão concatenados. No caso da APO, serão concatenados os atributos `pergunta_id` e `texto`, conforme a Figura 17.

A imagem mostra a interface de configuração do operador **Generate Concatenation**. O título da janela é **Parameters**. Abaixo, há um ícone de grade e o texto **Generate Concatenation**. As configurações são as seguintes:

- first attribute**: dropdown menu com o valor `pergunta_id`.
- second attribute**: dropdown menu com o valor `texto`.
- separator**: campo de texto com o valor `-`.
- trim values**: checkbox desativado.

Cada campo de entrada possui um ícone de ajuda (i) à direita.

Figura 17 – Tela de parâmetros do operador de concatenação

Após aplicar as configurações o resultado gerado é uma nova tabela, contendo a coluna resultante da concatenação dos atributos escolhidos, conforme pode ser visto na Figura 18 para o novo atributo `pergunta_id_texto`. Na concatenação, configurou-se a utilização do símbolo de pontuação (.) para a separação das strings. Assim, antes do símbolo de pontuação tem-se o número identificador da pergunta e após, o valor da resposta fornecido pelo morador para aquela pergunta.

Row No.	morador_id	pergunta_id	texto	conceito_id	pergunta_id_texto
1	336	100	uberlandia mg	0	100.0_uberlandia mg
2	336	101	3 QUARTOS	0	101.0_3 QUARTOS
3	336	102	FEMININO	0	102.0_FEMININO
4	336	112	DE 20 A 30 A...	0	112.0_DE 20 A 30 ANOS
5	336	113	SUPERIOR I...	0	113.0_SUPERIOR INCOMPLETO
6	336	114	OUTRO	0	114.0_OUTRO
7	336	115	PESSOA SÓ	0	115.0_PESSOA SÓ
8	336	116	1	0	116.0_1
9	336	117	ENTRE R\$10...	0	117.0_ENTRE R\$1000 E R\$2000
10	336	118	PRÓPRIO	0	118.0_PRÓPRIO

Figura 18 – Resultado da aplicação do operador de concatenação

Uma nova concatenação foi necessária, para agregar mais um atributo ao conjunto *pergunta+resposta*. Este novo atributo, o *conceito_id* acrescenta mais detalhes à resposta do morador, quando este avalia sua satisfação para uma pergunta relativa à um cômodo, considerando conceitos como: iluminação, ventilação, espaço, etc. Esta nova operação de concatenação integra o fluxo da ETL, como mostra a Figura 19, e é responsável pela inclusão da nova coluna, denominada *pergunta_id_texto_conceito_id* (Figura 20).



Figura 19 – Segunda aplicação do operador de concatenação

Após a aplicação da segunda operação de concatenação de valores, a tabela gerada até este ponto encontra-se na Figura 20.

Row No.	morador...	pergunta_id	texto	conceito_id	pergunta_id_texto	pergunta_id_texto_conceito_id
1	336	100	uberlandia mg	0	100.0_uberlandia mg	100.0_uberlandia mg_0.0
2	336	101	3 QUARTOS	0	101.0_3 QUARTOS	101.0_3 QUARTOS_0.0
3	336	102	FEMININO	0	102.0_FEMININO	102.0_FEMININO_0.0
4	336	112	DE 20 A 30 A...	0	112.0_DE 20 A 30 ANOS	112.0_DE 20 A 30 ANOS_0.0
5	336	113	SUPERIOR I...	0	113.0_SUPERIOR INCOMPL...	113.0_SUPERIOR INCOMPLETO_0.0
6	336	114	OUTRO	0	114.0_OUTRO	114.0_OUTRO_0.0
7	336	115	PESSOA SÓ	0	115.0_PESSOA SÓ	115.0_PESSOA SÓ_0.0
8	336	116	1	0	116.0_1	116.0_1_0.0
9	336	117	ENTRE R\$10...	0	117.0_ENTRE R\$1000 E R\$...	117.0_ENTRE R\$1000 E R\$2000_0.0
10	336	118	PRÓPRIO	0	118.0_PRÓPRIO	118.0_PRÓPRIO_0.0

Figura 20 – Resultado da aplicação do segundo operador de concatenação

Até este ponto foi realizada unicamente a seleção e o tratamento dos atributos que registram pergunta e respostas, mas não se sabe ainda se a resposta do morador é não-vazia para uma determinada pergunta. Para isso, foi criado um novo atributo, de nome **resp**, que com a ajuda do operador **Generate Attributes**, é utilizado para assinalar se a pergunta foi efetivamente respondida pelo do morador, conforme a Figura 21.

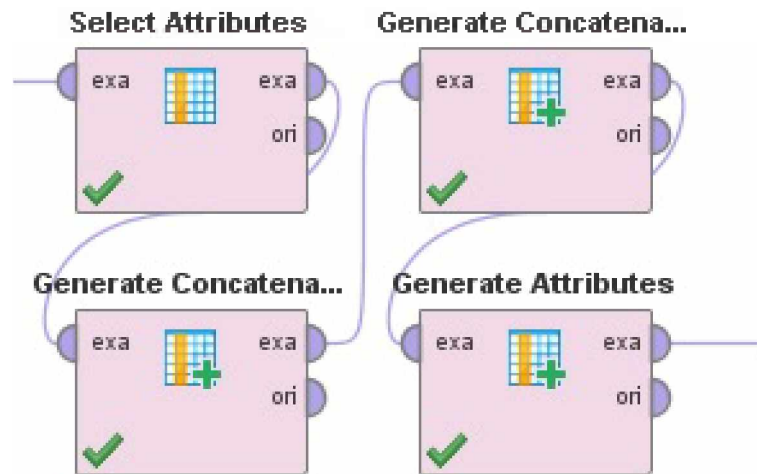


Figura 21 – Fluxo do processo com o operador **Generate Attributes**

O operador **Generate Attributes** funciona a partir de uma expressão matemática, por exemplo, com a definição do nome da nova coluna como **resp** (*attribute name*) associada à uma expressão lógica ou matemática (*function expressions*). No caso ca APO, a coluna **texto** guarda a resposta dada à uma pergunta, e caso este valor seja vazio, deduz-se que a pergunta não foi respondida, e o atributo **resp** tem valor zero. A Figura 22 mostra a função que gera o novo atributo **resp**.

attribute name	function expressions
resp	if(texto=="",0,1)

Figura 22 – Configuração da coluna **resp** com a sua função de geração de atributo

Após a aplicação do processo, obteve-se a seguinte tabela, com a coluna **resp** com os valores já calculados para o novo atributo **resp**. A Figura 23 mostra o resultado após a aplicação do operador **Generate Attributes**.

Row No.	morador_id	pergunta_id	texto	conc...	pergunta_id_texto	pergunta_id_texto_conceito_id	resp
1	336	100	uberlandia mg	0	100.0_uberlandia mg	100.0_uberlandia mg_0.0	1
2	336	101	3 QUARTOS	0	101.0_3 QUARTOS	101.0_3 QUARTOS_0.0	1
3	336	102	FEMININO	0	102.0_FEMININO	102.0_FEMININO_0.0	1
4	336	112	DE 20 A 30 A...	0	112.0_DE 20 A 30 AN...	112.0_DE 20 A 30 ANOS_0.0	1
5	336	113	SUPERIOR I...	0	113.0_SUPERIOR IN...	113.0_SUPERIOR INCOMPLET...	1
6	336	114	OUTRO	0	114.0_OUTRO	114.0_OUTRO_0.0	1
7	336	115	PESSOA SÓ	0	115.0_PESSOA SÓ	115.0_PESSOA SÓ_0.0	1
8	336	116	1	0	116.0_1	116.0_1_0.0	1
9	336	117	ENTRE R\$10...	0	117.0_ENTRE R\$100...	117.0_ENTRE R\$1000 E R\$20...	1
10	336	118	PRÓPRIO	0	118.0_PRÓPRIO	118.0_PRÓPRIO_0.0	1

Figura 23 – Tela de resposta após o operador de geração de atributo ser executado

Após geradas as concatenações e após inclusos os novos atributos, foram preparadas as colunas que deverão ser mineradas. Para este fim, aplicada novamente a seleção de atributos. A Figura 24 mostra a continuação do fluxo do processo .

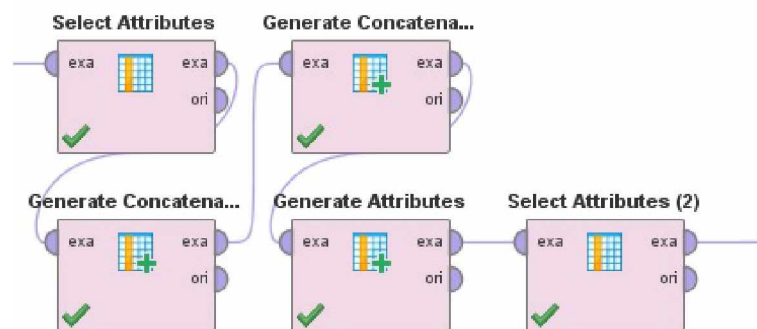


Figura 24 – Fluxo do processo com a segunda etapa da aplicação da seleção de atributos

Na nova aplicação da operação **Select Attributes**, a configuração para a nova seleção é feita com o parâmetro **subset**, indicando que um novo subconjunto será escolhido. Para a continuação da mineração dos dados da APO selecionou-se as colunas: **morador_id**, **pergunta_id_texto_conceito_id**, **resp**, conforme Figura 25.

Row No.	morador_id	pergunta_id_texto_conceito_id	resp
54	336	132.0_BOA,_43.0	1
55	336	133.0_BOM,_44.0	1
56	336	180.0_NORMAL,_44.0	1
57	336	135.0_NÃO SE APLICA,_46.0	1
58	336	135.0_NÃO SE APLICA,NÃO SE APLICA,_4...	1
59	336	181.0_NORMAL,_58.0	1
60	336	136.0_MÉDIO,_29.0	1
61	336	179.0_NORMAL,_42.0	1
62	336	181.0__58.0	0

Figura 25 – Tela de resposta após aplicação do segundo operador de seleção de atributos

Neste ponto, as colunas que serão utilizadas foram configuradas, e pode-se então aplicar o operador **Pivot** para realizar a rotação das linhas em colunas. Desta forma, agrupam-se os dados por **morador_id** e transformam-se cada um dos valores (nas linhas) do atributo **pergunta_id_texto_conceito_id** em coluna. O Rapidminer oferece a configuração dos parâmetros para a realização da rotação. No campo **group attribute** define-se a coluna que irá especificar o atributo de agrupamento para a escolha do **morador_id**. No campo **index attribute** especifica-se qual o atributo contém os valores que serão usados para identificar os exemplos dentro dos grupos. Os valores desse atributo são usados para nomear as colunas durante a rotação, conforme o fluxo mostrado na Figura 26.

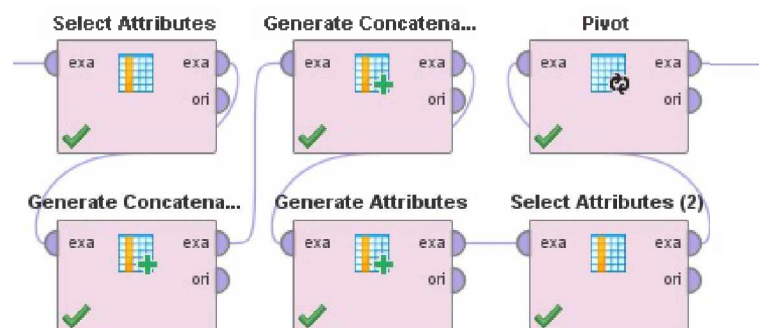


Figura 26 – Fluxo do processo com o operador Pivot

O resultado gerado após a aplicação do operador **Pivot**, é o agrupamento feito de acordo com a coluna **morador_id**. Então, para cada morador têm-se todas as opções

de resposta e pergunta. Pode-se notar que a coluna **resp** foi integrada junto às outras colunas, conforme mostra a Figura 27.

Row No.	morador_id	resp_142.0_BOM_0.0	resp_142.0_MUITO BOM_0.0	resp_142.0_MUITO RUIM_0.0	resp_142.0_...	resp_142.0_...
1	0	1	1	1	1	1
2	336	?	?	?	?	?
3	337	?	?	?	?	?
4	339	?	?	?	?	?
5	340	?	?	?	?	?
6	341	?	?	?	?	?
7	342	?	?	?	?	?
8	343	?	?	?	?	?
9	344	?	?	?	?	?
10	345	?	?	?	?	?

Figura 27 – Resultado da aplicação do operador Pivot

Pode-se notar que após a aplicação do Pivot conforme a Figura 27, foram colocados pontos de interrogação como valor onde o morador não respondeu a pergunta. Para este fim, utilizou-se o operador **Replace Missing Values**, para substituir os pontos de interrogação pelo valor zero. Tem-se então um padrão para todos os valores, em 0 ou 1. A Figura 28 apresenta o fluxo contendo operador **Replace Missing Values**.

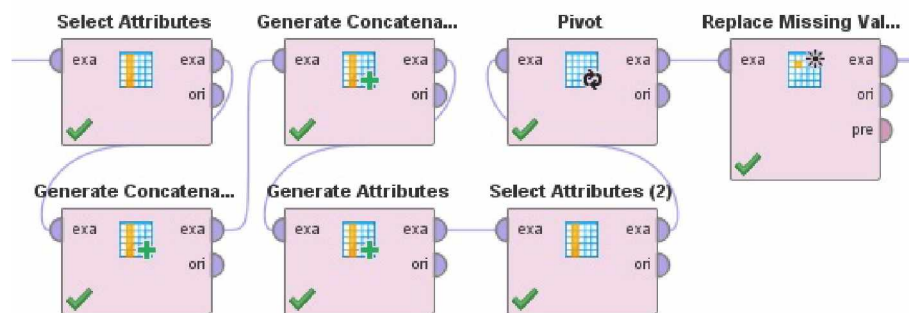


Figura 28 – Fluxo do processo com o operador Replace Missing Values

O resultado após a aplicação do operador **Replace Missing Values** é mostrado na Figura 29.

Row No.	morador_id	resp_142.0_BOM_0.0	resp_142.0_MUITO BOM_0.0	resp_142.0_...	resp_142.0_...	resp_142.0_...	resp_143.0_...
1	0	1	1	1	1	1	1
2	336	0	0	0	0	0	0
3	337	0	0	0	0	0	0
4	339	0	0	0	0	0	0
5	340	0	0	0	0	0	0
6	341	0	0	0	0	0	0
7	342	0	0	0	0	0	0
8	343	0	0	0	0	0	0
9	344	0	0	0	0	0	0
10	345	0	0	0	0	0	0

Figura 29 – Tela com os resultados após a aplicação do fluxo do processo da Figura 28

Na etapa seguinte os dados devem ser transformados em valores binomiais. O operador **Numerical to Binominal** altera o tipo dos atributos numéricos para falso ou verdadeiro. No Rapidminer estes valores são **true** e **false**, e a inclusão do operador é mostrada na Figura 30.

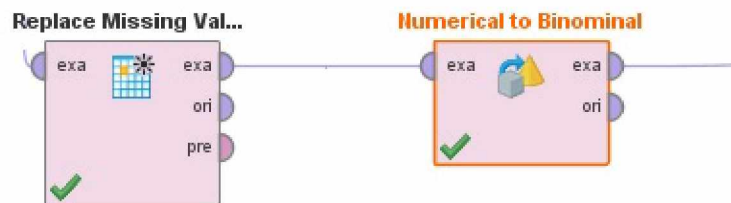


Figura 30 – Fluxo do processo de ETL com o operador Numerical to Binominal

Ainda no operador **Numerical to Binomial** é possível criar uma expressão regular para aplicar a função de troca de valor numérico para booleano. Esta expressão permite exprimir que todas as colunas começadas com o termo **resp** terão seus valores transformados. Assim, o uso da expressão regular na configuração dos parâmetros permite a aplicação do operador apenas para as colunas que satisfazem a expressão regular, e os outros atributos não se alteram. A Figura 31 mostra a escrita da expressão regular para que as colunas contendo respostas pudessem ter seus valores transformados em verdadeiro e falso.

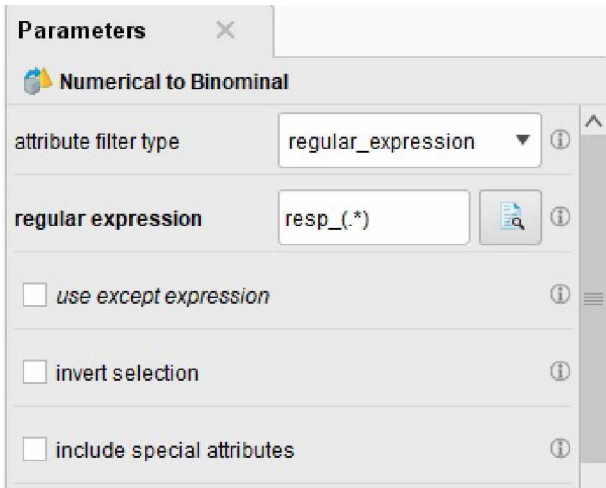


Figura 31 – Tela da configuração dos parâmetros do operador Numerical to Binominal

A Figura 32 mostra o resultado após a aplicação do operador Numerical to Binominal.

Row No.	resp_142.0_BOM_0.0	resp_142.0_MUITO BOM_0.0	resp_142.0_MUITO RUIM_0.0	resp_142.0_...	resp_142.0_...
1	true	true	true	true	true
2	false	false	false	false	false
3	false	false	false	false	false
4	false	false	false	false	false
5	false	false	false	false	false
6	false	false	false	false	false
7	false	false	false	false	false
8	false	false	false	false	false
9	false	false	false	false	false
10	false	false	false	false	false

Figura 32 – Resultados da aplicação do operador Numerical to Binominal

Como último operador a ser aplicado no pré processamento, o **Set Role** designa qual será o papel do atributo dentro do processo, e conforme a Figura 33, este operador é incluído ao final do processo de ETL.

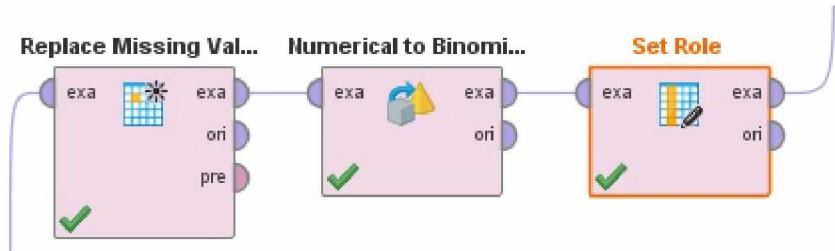


Figura 33 – Etapa final do processo de ETL com o operador Set Role

O operador oferece a configuração de parâmetros para que um dos atributos tenha papel preponderante sobre os outros. Para a aplicação da APO, o atributo `morador_id` foi definido com o papel de ser o `id` do processo, assim ele atua como um identificador para as respostas que serão analisadas. A Figura 34 mostra a configuração dos parâmetros para o operador `Set Role`.

Figura 34 – Tela de parâmetros do operador `Set Role`

No final do processo de ETL, obtiveram-se todos os dados selecionados e transformados para o padrão exigido pelo algoritmo `FP-growth`. Após a aplicação do operador `Set Role`, os dados são colocados a disposição da etapa de mineração de acordo com a Figura 35.

Row No.	morador_id	resp_142.0_...	resp_142.0_...	resp_142.0_...	resp_142.0_...	resp_142.0_...	resp_143.0_...	resp_143.0_...
1	0	true	true	true	true	true	true	true
2	336	false	false	false	false	false	false	false
3	337	false	false	false	false	false	false	false
4	339	false	false	false	false	false	false	false
5	340	false	false	false	false	false	false	false
6	341	false	false	false	false	false	false	false
7	342	false	false	false	false	false	false	false
8	343	false	false	false	false	false	false	false
9	344	false	false	false	false	false	false	false
10	345	false	false	false	false	false	false	false

Figura 35 – Tela com os resultados após a aplicação do operador `Set Role`

4.3 Data Mining

Entrando na etapa em que acontece a mineração, o operador `FP-Growth` realiza o trabalho de encontrar os `itemsets` frequentes conforme detalhado na seção 3.4.2. No

Rapidminer o operador **FP-Growth** calcula todos os itemsets frequentes utilizando a estrutura de dados da árvore FP, e para isso, ele recebe como entrada todos os atributos do conjunto de dados no formato booleano (operador Numerical to Binominal transforma os dados de numéricos para booleanos).

O conceito básico da análise da cesta de supermercado é aplicado neste algoritmo, utilizando os conceitos de transações e itens. O conjunto de itens é representado pelo conjunto de atributos e as transações são os exemplos, o algoritmo então assume que se o valor de um atributo for verdadeiro, neste caso o item está presente na transação, caso contrário não está na transação.

A árvore FP é construída pelo operador que calcula todos os conjuntos de itens frequentes a partir da base de dados de transação, neste caso a tabela de resposta. É construída uma cópia dos dados, e na maioria dos casos esta cópia é armazenada na memória principal. Mesmo para grandes base de dados, todos os itens frequentes são derivados da árvore FP. O fluxo do processo continua conforme a Figura 36.

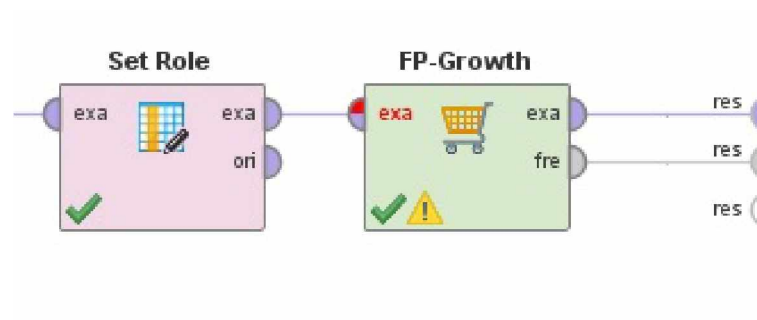


Figura 36 – Etapa com o operador FP-Growth no processo de mineração de dados

Para calcular os itens frequentes o algoritmo utiliza o critério de *suporte mínimo*. Este critério é utilizado para tentar solucionar o problema de encontrar conjuntos de itens que aparecem juntos em pelo menos uma certa relação de transações. Para configurar o suporte, o Rapidminer oferece alguns parâmetros para serem estipulados, que são:

- **Min number of itemsets:** este parâmetro especifica o número mínimo de *itemsets* que devem ser minerados.
- **Max number of retries:** Este operador determina quantas vezes deve-se diminuir o valor do suporte mínimo para encontrar a quantidade mínima de *itemsets*, por convenção cada vez que é aplicado é diminuído em 20%.
- **Min support:** Define o valor do critério do suporte mínimo.
- **Max items:** Determina o número máximo de itens em um conjunto. Se definido como -1, este parâmetro não impõe nenhum limite superior.

A Figura 37 apresenta a configuração dos parâmetros:

Figura 37 – Tela da configuração de parâmetros do operador FP-Growth

Após a aplicação do algoritmo, foram gerados os itens frequentes, conforme a Figura 38:

FrequentItemSets (FP-Growth)		ExampleSet (Set Role)			
No. of Sets: 29					
Total Max. Size: 3					
Min. Size: 1					
Max. Size: 3					
Contains Item:					
Update View					
Size	Support	Item 1	Item 2	Item 3	
2	0.237	resp_138.0_SIM_0.0	resp_140.0_DIARIAMENTE_0.0		
2	0.224	resp_1700.0_Estou de acordo c...	resp_1729.0_Todo dia_127.0		
2	0.224	resp_1700.0_Estou de acordo c...	resp_1729.0_Cozinha_129.0		
2	0.221	resp_1700.0_Estou de acordo c...	resp_1723.0_Sim_0.0		
2	0.221	resp_1700.0_Estou de acordo c...	resp_1709.0_própria em aqui...		
2	0.218	resp_1700.0_Estou de acordo c...	resp_1728.0_Agradável_115.0		
2	0.218	resp_1700.0_Estou de acordo c...	resp_1729.0_Todo dia_129.0		
2	0.221	resp_1729.0_Todo dia_127.0	resp_1729.0_Cozinha_129.0		
2	0.218	resp_1729.0_Todo dia_127.0	resp_1723.0_Sim_0.0		
2	0.221	resp_1729.0_Todo dia_127.0	resp_1709.0_própria em aqui...		
2	0.221	resp_1729.0_Cozinha_129.0	resp_1723.0_Sim_0.0		
2	0.218	resp_1729.0_Cozinha_129.0	resp_1709.0_própria em aqui...		
3	0.237	resp_177.0_CONVENCIONAL_...	resp_138.0_SIM_0.0	resp_140.0_DIARIAMEN...	
3	0.218	resp_1700.0_Estou de acordo c...	resp_1729.0_Todo dia_127.0	resp_1729.0_Cozinha_1...	
3	0.218	resp_1700.0_Estou de acordo c...	resp_1729.0_Todo dia_127.0	resp_1709.0_própria em...	
3	0.218	resp_1700.0_Estou de acordo c...	resp_1729.0_Cozinha_129.0	resp_1723.0_Sim_0.0	

Figura 38 – Tela dos itens frequentes encontrados pelo operador FP-Growth

As vezes o conceito de encontrar os itens frequentes é confundido com a criação das regras de associação. Depois de processado o operador FP-Growth, o operador Create Association Rules é aplicado para criar as regras de associação.

Como último operador a ser aplicado na descoberta das regras de associação, o **Create Association Rules** utiliza os itens frequentes encontrados na etapa anterior para gerar as regras. Estas regras descobrem a relação entre dados que parecem não estar relacionados, para isso utiliza o formato de regras se/então.

Uma regra de associação é construída em duas partes, um antecedente e um conseqüente, o antecedente é formado por um item ou um conjunto de itens encontrados nos dados e o conseqüente é um item ou um conjunto de itens que foram encontrados em combinação com o antecedente.

O operador utiliza o critério de confiança para identificar os relacionamentos mais importantes, a confiança indica o número de vezes que as regras de associação foram verdadeiras, então o conjunto de itens frequentes é escolhido através do critério de suporte e de confiança. A Figura 39 apresenta a aplicação do último operador no processo.

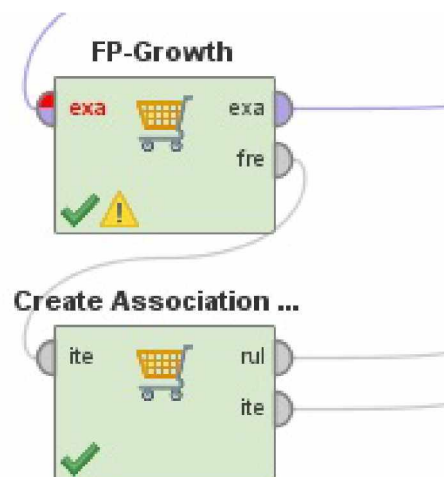


Figura 39 – Etapa final do processo de mineração de dados com o operador Create Association Rules

O Rapidminer oferece alguns critérios que são utilizados para a criação das regras, neste caso foi utilizado o critério da confiança. Os outros critérios são **lift**, **conviction**, **ps**, **gain**, **laplace** que podem ser selecionados no parâmetro **criterion**. O parâmetro **min confidence** determina qual o grau mínimo de confiança das regras, e os parâmetros **gain theta** e **laplace k** são opcionais utilizados quando os critérios **laplace** e **gain** são selecionados. A Figura 40 apresenta a tela de parâmetros.

Parameters X

Create Association Rules

criterion: confidence

min confidence: 0.8

gain theta: 2.0

laplace k: 1.0

Figura 40 – Tela de configuração dos parâmetros do operador Create Association Rules

A Figura 41, apresenta os resultados de todas as regras que foram encontradas de acordo com os critérios estabelecidos nas duas etapas do processo de mineração.

Show rules matching

all of these conclusions:

resp_177.0_CONVENCIONAL_0.0
resp_138.0_SIM_0.0
resp_140.0_DIARIAMENTE_0.0
resp_1700.0_Estou de acordo co...
resp_1729.0_Todo dia_127.0
resp_1729.0_Cozinha_129.0
resp_1723.0_Sim_0.0
resp_1709.0_própria em aquisiçã...
resp_1728.0_Agradável_115.0
resp_1729.0_Todo dia_129.0

Min. Criterion: confidence

Min. Criterion Value: 0.0

No.	Premises	Conclusion	Support	Confidence	LaPla
5	resp_177.0_CONVENCIONAL_0.0, resp_138.0_SIM_0.0	resp_140.0_DIARIAMENTE_0.0	0.237	0.872	0.972
6	resp_1700.0_Estou de acordo com o termo de consentiment...	resp_1728.0_Agradável_115.0	0.218	0.945	0.990
7	resp_1700.0_Estou de acordo com o termo de consentiment...	resp_1729.0_Todo dia_129.0	0.218	0.945	0.990
8	resp_1700.0_Estou de acordo com o termo de consentiment...	resp_1729.0_Todo dia_127.0, resp_1729.0_Cozi...	0.218	0.945	0.990
9	resp_1700.0_Estou de acordo com o termo de consentiment...	resp_1729.0_Todo dia_127.0, resp_1709.0_próp...	0.218	0.945	0.990
10	resp_1700.0_Estou de acordo com o termo de consentiment...	resp_1729.0_Cozinha_129.0, resp_1723.0_Sim_...	0.218	0.945	0.990
11	resp_1729.0_Todo dia_127.0	resp_1723.0_Sim_0.0	0.218	0.958	0.992
12	resp_1729.0_Cozinha_129.0	resp_1709.0_própria em aquisição_0.0	0.218	0.958	0.992
13	resp_1729.0_Todo dia_127.0	resp_1700.0_Estou de acordo com o termo de co...	0.218	0.958	0.992
14	resp_1729.0_Cozinha_129.0	resp_1700.0_Estou de acordo com o termo de co...	0.218	0.958	0.992
15	resp_1729.0_Todo dia_127.0	resp_1700.0_Estou de acordo com o termo de co...	0.218	0.958	0.992
16	resp_1729.0_Cozinha_129.0	resp_1700.0_Estou de acordo com o termo de co...	0.218	0.958	0.992
17	resp_1700.0_Estou de acordo com o termo de consentiment...	resp_1723.0_Sim_0.0	0.221	0.959	0.992
18	resp_1700.0_Estou de acordo com o termo de consentiment...	resp_1709.0_própria em aquisição_0.0	0.221	0.959	0.992
19	resp_1723.0_Sim_0.0	resp_1729.0_Todo dia_127.0	0.218	0.972	0.995
20	resp_1709.0_própria em aquisição_0.0	resp_1729.0_Cozinha_129.0	0.218	0.972	0.995

Figura 41 – Tela das regras de associações encontradas na etapa final do processo de mineração

Na apresentação dos resultados o Rapidminer oferece para o operador **Create Association Rules** mais outros dois tipos de visualização, o **Graph** e o **Description**. Na aba **Graph** apresentam-se as regras no esquema de grafo, porém a visualização do grafo com a quantidade de regras geradas não o torna compreensível para ser analisado, a Figura 42 mostra um grafo gerado.

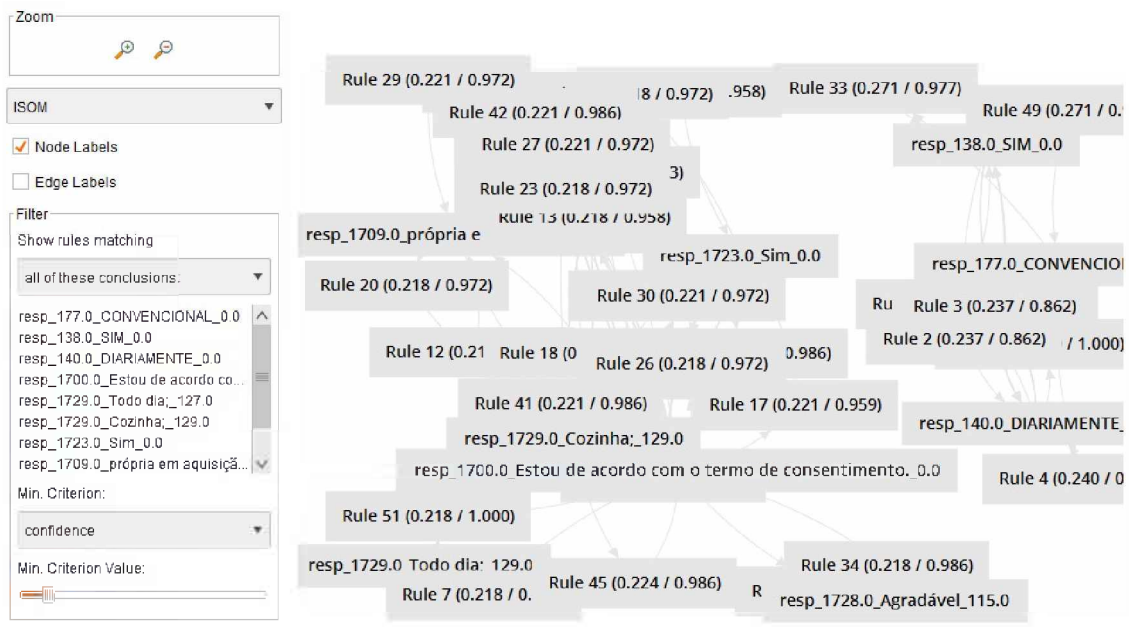


Figura 42 – Apresentação do grafo gerado pelo operador Create Association Rules

O esquema de visualização da aba **Description** lista todas as regras de associação geradas no formato antecedente/consequente. Este tipo de visualização torna a análise das regras de associações mais fácil e objetiva. A Figura 43 apresenta os resultados das regras encontradas para a APO.

AssociationRules
Association Rules <pre> [resp_177.0_CONVENCIONAL_0.0] --> [resp_138.0_SIM_0.0, resp_140.0_DIARIAMENTE_0.0] (confidence: 0.852) [resp_138.0_SIM_0.0] --> [resp_140.0_DIARIAMENTE_0.0] (confidence: 0.862) [resp_138.0_SIM_0.0] --> [resp_177.0_CONVENCIONAL_0.0, resp_140.0_DIARIAMENTE_0.0] (confidence: 0.862) [resp_177.0_CONVENCIONAL_0.0] --> [resp_140.0_DIARIAMENTE_0.0] (confidence: 0.864) [resp_177.0_CONVENCIONAL_0.0, resp_138.0_SIM_0.0] --> [resp_140.0_DIARIAMENTE_0.0] (confidence: 0.872) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1728.0_Agradável_115.0] (confidence: 0.945) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1729.0_Todo dia; 129.0] (confidence: 0.945) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1729.0_Todo dia; 127.0, resp_1729.0_Cozinha; 129.0] (confidence: 0.945) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1729.0_Todo dia; 127.0, resp_1709.0_própria em aquisição_0.0] (confidence: 0.945) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1729.0_Cozinha; 129.0, resp_1723.0_Sim_0.0] (confidence: 0.958) [resp_1729.0_Cozinha; 129.0] --> [resp_1709.0_própria em aquisição_0.0] (confidence: 0.958) [resp_1729.0_Todo dia; 127.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Cozinha; 129.0] (confidence: 0.972) [resp_1729.0_Cozinha; 129.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Todo dia; 127.0] (confidence: 0.972) [resp_1729.0_Todo dia; 127.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1709.0_própria em aquisição_0.0] (confidence: 0.972) [resp_1729.0_Cozinha; 129.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1723.0_Sim_0.0] (confidence: 0.972) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1723.0_Sim_0.0] (confidence: 0.959) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1709.0_própria em aquisição_0.0] (confidence: 0.972) [resp_1709.0_própria em aquisição_0.0] --> [resp_1729.0_Cozinha; 129.0] (confidence: 0.972) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Todo dia; 127.0] --> [resp_1729.0_Cozinha; 129.0] (confidence: 0.972) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Cozinha; 129.0] --> [resp_1729.0_Todo dia; 127.0] (confidence: 0.972) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Todo dia; 127.0] --> [resp_1709.0_própria em aquisição_0.0] (confidence: 0.972) [resp_1709.0_própria em aquisição_0.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Cozinha; 129.0] (confidence: 0.972) [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Cozinha; 129.0] --> [resp_1723.0_Sim_0.0] (confidence: 0.972) [resp_1723.0_Sim_0.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Cozinha; 129.0] (confidence: 0.972) </pre>

Figura 43 – Visualização das regras de associação criadas no formato A → B

5 Análise dos Resultados

A técnica de regras de associação depende de dois parâmetros: o suporte e a confiança. Os resultados variam de acordo com os valores estipulados para estes dois parâmetros, e para que sejam encontradas regras interessantes, é necessário que os valores mínimos de suporte e confiança sejam alcançados e que sejam valores relativamente altos para que a regra seja forte, dependendo da frequência das transações.

Os resultados apresentados pelo RapidMiner são de fácil compreensão para os profissionais que entendem do processo de mineração e sabem como funciona a tarefa de regras de associação, porém para os usuários finais da FAUeD as informações que foram geradas não seriam muito bem compreendidas pois não estão em um formato claro ou intuitivo, como por exemplo, se X então Y. Este trabalho não tem o intuito de transformar as regras de associação encontradas em uma linguagem formal, podendo assim esta transformação para o usuário final ser uma continuação deste trabalho.

Analisando os resultados obtidos, na configuração do FP-Growth, o valor do suporte mínimo foi de 0.95 (considerando que o valor varia de 0.0 a 1.0) com a expectativa de encontrar no mínimo 100 itens frequentes e a quantidade de vezes que o operador poderia reduzir o valor do suporte em 20% foi de 15 vezes, ou seja, a cada iteração do algoritmo que não fosse atingida a meta de encontrar 100 itens frequentes, o operador poderia reduzir o valor do suporte para tentar alcançar a meta. No operador **Create Association Rules** foi utilizado o critério de confiança com o valor mínimo de 0.8 (o valor da confiança pode variar entre 0.0 - 1.0).

Com os parâmetros configurados, os resultados obtidos no operador FP-Growth foram de 29 itens frequentes encontrados com o suporte destes itens variando de 0.278 a 0.218, e o conjunto de *itemsets* variando entre 1 e 3 itens. Os valores do suporte não foram valores altos e o algoritmo precisou reduzir o valor do suporte para encontrar os itens frequentes, considerando que a mineração inicial para validação do processo de ETL e mineração de dados foi realizada sobre uma base de dados com respostas de 317 moradores. Com essas informações pode-se concluir que não há uma relação forte entre as perguntas com as respostas dos moradores em relação a quantidade total de transações que foram mineradas.

Na etapa de encontrar as regras de associação, o RapidMiner gerou 47 regras de associação, com o valor da confiança dessas regras variando entre 0.872 e 1, considerando que o valor do suporte destas regras está na faixa de 0.218 - 0.278, não se pode afirmar que as regras encontradas são fortes e interessantes, pois mesmo com os valores de confiança perto do valor máximo que é 1, estes *itemsets* podem aparecer poucas vezes nas transações,

o que não torna a regra forte, pois o algoritmo teve de reduzir o valor do suporte e a confiança depende dos valores de suporte dos itemsets.

Nos resultados do operador Association Rules, ele fornece as regras de associação no formato *precedente* $\rightarrow$ *consequente*, conforme a Figura 44:

AssociationRules

```
Association Rules
[resp_177.0_CONVENCIONAL_0.0] --> [resp_138.0_SIM_0.0, resp_140.0_DIARIAMENTE_0.0] (confidence: 0.852)
[resp_138.0_SIM_0.0] --> [resp_140.0_DIARIAMENTE_0.0] (confidence: 0.862)
[resp_138.0_SIM_0.0] --> [resp_177.0_CONVENCIONAL_0.0, resp_140.0_DIARIAMENTE_0.0] (confidence: 0.862)
[resp_177.0_CONVENCIONAL_0.0] --> [resp_140.0_DIARIAMENTE_0.0] (confidence: 0.864)
[resp_177.0_CONVENCIONAL_0.0, resp_138.0_SIM_0.0] --> [resp_140.0_DIARIAMENTE_0.0] (confidence: 0.872)
[resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1728.0_Agradável_115.0] (confidence: 0.945)
[resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1729.0_Todo dia;_129.0] (confidence: 0.945)
[resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1729.0_Todo dia;_127.0, resp_1729.0_Cozinha;_129.0] (confidence: 0.945)
[resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1729.0_Todo dia;_127.0, resp_1709.0_própria_129.0] (confidence: 0.945)
[resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1729.0_Cozinha;_129.0, resp_1723.0_Sim_0.0] (confidence: 0.958)
[resp_1729.0_Todo dia;_127.0] --> [resp_1723.0_Sim_0.0] (confidence: 0.958)
[resp_1729.0_Cozinha;_129.0] --> [resp_1709.0_própria em aquisição_0.0] (confidence: 0.958)
[resp_1729.0_Todo dia;_127.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Cozinha;_129.0] (confidence: 0.958)
[resp_1729.0_Cozinha;_129.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1729.0_Todo dia;_127.0] (confidence: 0.958)
[resp_1729.0_Todo dia;_127.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1709.0_própria_129.0] (confidence: 0.958)
[resp_1729.0_Cozinha;_129.0] --> [resp_1700.0_Estou de acordo com o termo de consentimento_0.0, resp_1723.0_Sim_0.0] (confidence: 0.959)
[resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1723.0_Sim_0.0] (confidence: 0.959)
[resp_1700.0_Estou de acordo com o termo de consentimento_0.0] --> [resp_1709.0_própria em aquisição_0.0] (confidence: 0.972)
[resp_1723.0_Sim_0.0] --> [resp_1729.0_Todo dia;_127.0] (confidence: 0.972)
```

Figura 44 – Regras de associação encontradas no final do processo de mineração de dados

As regras de associação neste formato podem ser interpretadas para uma linguagem mais formal, por exemplo a regra $[resp_177.0_CONVENCIONAL_0.0] \rightarrow [resp_138.0_SIM_0.0, resp_140.0_DIARIAMENTE_0.0]$, tem a seguinte interpretação: "Alguns moradores que responderam a pergunta 177 (Qual é a tipologia do seu apartamento?) com a resposta Convencional, também responderam a pergunta 138 (Neste apartamento alguém usa internet?) com a resposta Sim e a pergunta 140 (Qual sua frequência em relação as refeições neste apartamento?) com a resposta Diariamente".

Foram realizados alguns testes com a variação dos valores dos critérios de suporte e confiança, se os valores são diminuídos mais itens frequentes são gerados e mais regras de associação são encontradas. Por exemplo alterando apenas o valor do suporte para 0.5, foram encontrados 291 itemsets frequentes e 2283 regras de associação, porém quanto menor os valores dos critérios de avaliação, menos "interessante" será essa regra.

Em relação ao tempo de execução e processamento de todo o processo de mineração, conforme foram configurados cada operador e levando em consideração o tamanho da base de dados, que não foi uma grande massa de dados a ser analisada, o algoritmo FP-Growth mostrou ser eficiente na geração dos itens frequentes, isso se deve a estrutura de dados e a forma com que o algoritmo utiliza a memória para alocar a árvore FP-tree, diminuindo a quantidade de vezes e o tempo que é gasto varrendo toda a base de dados.

Nota-se que as relações entre as perguntas que foram respondidas podem não ter algum sentido lógico entre elas, por exemplo regras com a premissa ou consequente

'estou de acordo com o termo de consentimento' utilizam uma resposta que é sempre afirmativa pelo usuário e intuitiva na análise, mas a interpretação destas perguntas com os seus significados podem ser facilmente construídas para a análise dos usuários finais que aplicarão essas informações em futuros projetos. Contudo foram criadas as regras de associação respeitando a teoria e os critérios da tarefa de associação.

6 Conclusão

Ao longo do tempo, um grande volume de informação vem sendo criado e gerando uma imensa demanda de armazenamento dessas informações, com a ascensão da tecnologia e dos bancos de dados ao longo dos anos, este problema foi solucionado, possibilitando extrair toda a informação que antes era guardada em papéis e planilhas, em bancos de dados que fornecem um acesso rápido e confiável a estas informações. A mineração de dados surgiu como uma solução para a extração desses dados brutos sem um conteúdo que possa realmente ser utilizado pelas empresas, transformando e oferecendo uma análise dessas informações que posteriormente possam gerar valor de alguma forma para diferentes áreas, como a comercial e o marketing.

A moradia é uma necessidade básica e primordial para as pessoas do mundo inteiro, desde o início da história da humanidade, nossos ancestrais buscavam lugares que fornecessem segurança, a melhor forma possível de acesso a outras necessidades como se alimentar e encontrar água, e com o tempo foram desenvolvendo novas estruturas para as casas até chegar nos dias de hoje, onde as casas e apartamentos são completamente adaptáveis ao terreno e ao clima do local. Realizando o estudo sobre as avaliações de pós-ocupação, juntamente com o estudo da mineração de dados, este trabalho tenta encontrar relações em diversas opiniões de diferentes pessoas que possam ajudar a entender melhor as necessidades que elas possuem com as suas moradias.

O algoritmo FP-Growth utilizado na etapa de mineração, é um algoritmo um pouco mais complexo do que outros algoritmos que foram candidatos a serem utilizados como o algoritmo Apriori, essa complexidade se deve a forma com que é criada a estrutura de dados FP-Growth, que é necessário empregar algoritmos de busca em profundidade e largura e topologias bottom-up, porém a sua eficiência é maior do que a dos outros algoritmos e é recomendado para grandes volumes de dados que vão ser minerados. Com a utilização deste algoritmo foram encontradas as regras de associação correlacionando as perguntas e respostas atingindo o objetivo do trabalho.

As regras de associação que foram encontradas neste trabalho, podem ser analisadas junto com a FAUeD que possui um entendimento melhor das APO's, gerando conhecimento que possa ser utilizado em futuros projetos de arquitetura. Com a extração e o acúmulo de mais informações de diferentes tipos de habitações, locais e moradores, este trabalho possibilita a análise de padrões relacionados, oferecendo toda a base de conhecimento em KDD e Mineração de dados, que pode ser utilizada para tentar a aplicação de diferentes técnicas de mineração que encaixem no objetivo proposto, de ajudar a melhorar as necessidades das pessoas em suas moradias.

Para o futuro, este trabalho fornece uma base para extração de conhecimento de bases de dados com a utilização da tarefa de associação, na análise das regras criadas podem ser analisados outros critérios como lift e conviction que devem ser estudados para fazer a aplicação correta dos mesmos sobre as regras de associação. Outra futura possibilidade seria o emprego de outras tarefas de mineração de dados como a classificação, que também pode ser candidata a ser aplicada no processo de mineração, porém seria necessário avaliar e transformar os dados na etapa de pré-processamento para que sejam utilizados da forma correta no algoritmo de classificação que for escolhido.

Referências

- AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. *Mining Association Rules between Sets of Items in Large Databases*. [S.l.], 1993. Disponível em: <<http://arbor.ee.ntu.edu.tw/~chyun/dmpaper/agrama93.pdf>>. Citado na página 17.
- AMO, S. de. *Técnicas de Mineração de Dados*. [S.l.], 2006. Disponível em: <<http://www.deamo.prof.ufu.br/arquivos/JAI-cap5.pdf>>. Citado 2 vezes nas páginas 17 e 24.
- CORTES, S. C.; PORCARO, R. M.; LIFSCHITZ, S. *Mineração de Dados - Funcionalidades, Técnicas e Abordagens*. [S.l.], 2002. Disponível em: <ftp://obaluae.inf.puc-rio.br/pub/docs/techreports/02_10_cortes.pdf>. Citado na página 17.
- COSTA, M. *O processo de ETL na construção de conhecimento em uma aplicação de uma empresa seguradora*. 2009. Monografia (Bacharel em Ciência da Computação), UFJF (Universidade Federal de Juiz de Fora), MG, Brasil. Citado 2 vezes nas páginas 14 e 15.
- CUNHA, F. R.; ARAUJO, R. L. P. de; SANTOS, N. dos. Uso de técnicas de data mining no monitoramento de alunos on-line. *Cadernos do IME-Série Informática*, v. 15, 2003. Citado na página 16.
- FAYYAD, U. M. et al. *Advances in Knowledge Discovery and Data Mining*. Menlo Park: The MIT Press, 1996. Citado na página 13.
- GIUDICI, P.; FIGINI, S. Market basket analysis. *Applied Data Mining for Business and Industry, Second Edition*, Wiley Online Library, p. 175–191, 2009. Citado na página 17.
- GOMES, T. C. S. *Descoberta de Conhecimento Utilizando Mineração de Dados Educacionais Abertos*. 2016. Disponível em: <<http://200.17.137.109:8081/novobsi/bsi-na-ufrpe-recife/trabalhos-de-conclusao-de-curso/2015-2/Descoberta%20de%20conhecimento%20usando%20mineracao%20de%20dados%20educacionais%20abertos.pdf>>. Acesso em: 02 Jun. 2016. Citado na página 13.
- GONCALVES, E. Market basket analysis. v. 111, p. 28–38, 05 2013. Citado na página 18.
- HAN, J.; KAMBER, M.; PEI, J. *Data Mining Concepts and Techniques*. [S.l.], 2012. Disponível em: <<http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf>>. Citado na página 15.
- HAN, J.; PEI, J.; YIN, Y. Mining frequent patterns without candidate generation. In: ACM. *ACM sigmod record*. [S.l.], 2000. v. 29, n. 2, p. 1–12. Citado 2 vezes nas páginas 9 e 10.

HOFMANN, M.; KLINKENBERG, R. *RapidMiner: Data mining use cases and business analytics applications*. [S.l.]: CRC Press, 2013. Citado na página 9.

KIRA, G. *Aplicação de técnicas de mineração de dados à um recorte do repositório de dados históricos sobre eleições brasileiras do Tribunal Superior Eleitoral Brasileiro*. 2016. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/3020/1/CT_COBSI_2013_2_02.pdf>. Acesso em: 25 Maio 2016. Citado 2 vezes nas páginas 16 e 26.

MARIANO, M. *Comparação de algoritmos paralelos para a extração de regras de associação no modelo de memória distribuída*. 2011. Dissertação (Mestre em Ciência da Computação), UFMS (Universidade Federal de Mato Grosso do Sul), MS, Brasil. Citado 2 vezes nas páginas 4 e 23.

NANDI, J. C. et al. *O Algoritmo de Associação Frequent Pattern-Growth na Shell Orion Data Mining Engine*. [S.l.], 2015. Disponível em: <<http://arbor.ee.ntu.edu.tw/~chyun/dmpaper/agrama93.pdf>>. Citado na página 25.

OLIVEIRA, W. J. *DataWare House*. [S.l.]: Visual Books, 2009. Citado 2 vezes nas páginas 14 e 15.

PEREIRA, M. H.; ABREU, A. de S.; REIS, P. A. *Aplicação e Gerenciamento de Avaliações de Pós-Ocupação usando Interfaces Digitais*. [S.l.], 2017. Citado na página 27.

PHYU, T. N. Survey of classification techniques in data mining. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists*. [S.l.: s.n.], 2009. v. 1, p. 18–20. Citado na página 16.

RAPIDMINER. *RapidMiner Studio Manual*. [S.l.], 2014. Disponível em: <<https://docs.rapidminer.com/downloads/RapidMiner-v6-user-manual.pdf>>. Citado 2 vezes nas páginas 19 e 20.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introdução ao DATA MINING Mineração de Dados*. Rio de Janeiro: Ciência Moderna, 2009. Citado na página 16.