UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

GUSTAVO KIRA

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS À UM RECORTE DO REPOSITÓRIO DE DADOS HISTÓRICOS SOBRE ELEIÇÕES BRASILEIRAS DO TRIBUNAL SUPERIOR ELEITORAL BRASILEIRO

CURITIBA

2014

GUSTAVO KIRA

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS À UM RECORTE DO REPOSITÓRIO DE DADOS HISTÓRICOS SOBRE ELEIÇÕES BRASILEIRAS DO TRIBUNAL SUPERIOR ELEITORAL BRASILEIRO

Trabalho de Conclusão de Curso, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Bacharelado em Sistemas de Informação - da Universidade Tecnológica Federal do Paraná - UTFPR, Campus Curitiba, como requisito para obtenção do título de Bacharel.

Orientador: Dr. Eng. Celso A. A. Kaestner

CURITIBA

2014

AGRADECIMENTOS

Gostaria de agradecer a todos que, de alguma maneira, contribuíram com este trabalho. Em especial ao professor Celso, pelo voto de confiança.

If you mine the data hard enough, you can also find messages from god.
– Dogbert

RESUMO

KIRA, Gustavo. APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS À UM RECORTE DO REPOSITÓRIO DE DADOS HISTÓRICOS SOBRE ELEIÇÕES BRASILEIRAS DO TRIBUNAL SUPERIOR ELEITORAL BRASILEIRO. 115 f. – Departamento Acadêmico de informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

Este estudo tem como objetivo usar o repositório de dados eleitorais históricos do Tribunal Superior Eleitoral como um estudo de caso para aplicação de técnicas de descoberta de conhecimento em banco de dados. Para a execução do projeto, foram escolhidas seis técnicas de classificação (C4.5, Knn-3, Knn-5, Máquina de Suporte Vetorial, Perceptron de Múltiplas Camadas, Bayes Ingênuo) e uma técnica de associação (Apriori), representado duas das quatro tarefas clássicas da Mineração de Dados. Para as técnicas de classificação, foram usados os dados referentes aos cargos de Deputado Federal, Deputado Estadual e Deputado Distrital, todos do ano 2010. Já para a tarefa de associação, foram usados os dados sobre os cargos de Vereador e Prefeito, ambos de 2012.

Palavras-chave: Mineração de Dados, Dados Eleitorais, Classificação, Associação

ABSTRACT

KIRA, Gustavo. DATA MINING TECHNIQUES APPLIED AT THE ELECTORAL SUPERIOR COURT'S BRAZILIAN HISTORICAL DATA REPOSITORY OF ELECTIONS. 115 f. – Departamento Acadêmico de informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

This study aims to apply a set of KDD (knowledge discovery in databases) tecniques at the historical election data repository from Superior Electoral Court (Tribunal Superior Eleitoral) as a case study. To complete this goal, five classification techiniques (C4.5, Knn-3, Knn-5, Support vector Machine, Multilayer Perceptron, Naive Bayes) and one association techinique (Apriori) were chosen. For the classification experiment, the data used was about the House elections from 2010. For the association experiment, the data used was from Mayor and City Concilmans from all cities, both from the year 2012.

Keywords: Data Mining, Election Data, Classification, Association

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

KDD	Knowledge Discovery in Databases
TSE	Tribunal Superior Eleitoral
MLP	Multi Layer Perceptron
K-NN	K-Nearest Neighbors
SVM	Support Vector Machine
GNU	Gnu is Not Unix
GPL	General Public Licence
ELT	Extract Load Transform
CSV	Comma Separeted Values
OLTP	Online Transaction Processing
OLAP	Online Analytical Processing
MVC	Model-View-Control
SQL	Structured Query Language

SUMÁRIO

1 INTRODUÇÃO

O livre acesso aos dados e informações públicas que o governo federal possui é um direito do cidadão brasileiro previsto na constituição. O inciso XXXIII do quinto artigo da constituição de 1988 define este direito: todos têm direito a receber dos órgãos públicos informações de seu interesse particular, ou de interesse coletivo ou geral, que serão prestadas no prazo da lei, sob pena de responsabilidade, ressalvadas aquelas cujo sigilo seja imprescindível à segurança da sociedade e do Estado (BRASIL, 1988).

Porém, mesmo que o livre acesso tenha sido descrito na constituição de 1988, a regulamentação do acesso à estes dados, ou seja, a forma e os limites deste direito, somente foram definidos em 2011, pela lei nº 12.527 (Lei de Acesso a Informação). O oitavo artigo desta lei prevê que toda informação de interesse coletivo ou geral, custodiada ou gerada por um órgão ou uma entidade pública, deve ser disponibilizada em local de fácil acesso. O segundo parágrafo deste mesmo artigo define o quão amplo deve ser o esforço para divulgação:

> "Para cumprimento do disposto no caput, os órgãos e entidades públicas deverão utilizar todos os meios e instrumentos legítimos de que dispuserem, sendo obrigatória a divulgação em sítios oficiais da rede mundial de computadores (internet)"(BRASIL, 2011)

Não somente órgãos e entidades públicas são obrigadas a disponibilizar os dados na internet, mas também devem disponibiliza-los de forma a garantir que possam ser usados. Assim, a redação da lei ainda prevê que os dados devem ser: de fácil compreensão, serem pesquisáveis, legíveis por máquina, dispostos em formatos de arquivos abertos, divulgar os detalhes da organização destes arquivos, garantir autenticidade e integridade, manter o dados atualizados. Quase todas as consideração sobre acessibilidade apresentadas no artigo da lei resolvem questões técnicas quanto à disponibilidade dos dados. Dificilmente são abordadas questões sobre a viabilidade de sua interpretação. Ou seja, mesmo disponíveis, precisam de algum tipo de manipulação para que se tornem informação.

A interpretação de dados fornecidos por repositórios não é alvo novo. Cientistas e pesquisadores usam e interpretam dados para criar modelos e teorias. Porém, o crescimento do ta-

manho das bases de dados, ocorrido principalmente devido aos baixos custos de armazenagem, fez com que a tarefa de transformar estes dados em informação ultrapassa-se as capacidades técnicas e cognitivas humanas (CARDOSO; MACHADO, 2008). Goldschmidt e Passos (2005) colocam o mesmo problema sob o argumento de que é inviável a análise de grande quantidade de dados pelo homem sem algum tipo de auxilio computacional. Indo um pouco além, Witten e Frank (2005) afirmam que o hiato entre a capacidade de gerar e entender dados cada vez aumenta mais. Pode-se entender que o hiato não só existe, como também tem a tendência de aumentar. Uma das formas de lidar com este problema é o uso de técnicas de Descoberta de Conhecimento em Banco de Dados (KDD, do *inglês knowledge discovery in databases*). Fayyad et al. (1996b) definem o termo como: "o processo não trivial de identificação de padrões no dados que sejam válidos, novos, potencialmente úteis e entendíveis (tradução nossa)".

De acordo com este ponto de vista, KDD pode ser entendida como uma ferramenta que usa a própria computação para resolver uma questão que ela mesma criou. Ou seja, "KDD é uma tentativa de contornar um problema que a era da informação digital tornou um fato para todos nós: a sobrecarga de dados (tradução nossa)" (PIATETSKY-SHAPIRO, 1991).

Diversas áreas fazem proveito das técnicas de KDD, como setores de Vendas, Marketing, Administração, Finanças, Investimentos, Astronomia, Saúde, Agropecuária, Telecomunicações, etc. Em tese, qualquer setor que disponha de um banco com muitos dados e uma capacidade razoável de processamento, pode ser uma área aplicável de KDD.

Dentro destas variadas aplicações de KDD, podemos destacar algumas que visam algum tipo de contribuição para a sociedade, separando daquelas que tem um objetivo comercial ou privado. Esta distinção se mostra útil, pois este perfil de aplicação de descoberta de conhecimento em banco de dados é o mesmo perfil desejado para este estudo.

Na literatura especializada, especificamente em *data mining*, tem-se três trabalhos recentes que se encaixam dentro deste contexto. No trabalho descrito em Vianna (2010), as técnicas de mineração de dados (uma etapa do KDD) são aplicadas em um repositório de índices médicos criado a partir de três sistemas de informação (SINASC/SIM e SIMI), a fim de procurar padrões sobre mortalidade infantil. Enquanto em Malucelli (2010) temos o uso de KDD para tentar procurar regras de classificação que ajudem a tomada de decisão sobre políticas e ações de saúde. Boschi et al. (2011) aplicam técnicas de mineração de dados com o objetivo de analisar o comportamento espaço temporal da precipitação pluvial no Estado do Rio Grande do Sul, em um dado período de tempo. A base de dados fonte para o estudo é proveniente da ANA (Agência Nacional de Água), um órgão federal.

Vianna (2010) e Boschi et al. (2011), mostram que a mineração de dados em repo-

sitório públicos é aplicada mesmo antes da criação da lei que facilita este acesso. Mesmo o tema não sendo de todo inédito, a aplicação de KDD em uma base de dados pública por si só pode ser considerado uma contribuição de cunho social. Pois, se o acesso à informação é tida como um requisito para luta contra a corrupção, melhora da gestão pública e inclusão social Brasil (2010), como isso acontecerá se os dados, da maneira como são disponibilizados, não são compreensíveis? Para Fayyad et al. (1996a, p. 15) os dados em si ("crus"), não possuem muito valor. Do ponto de vista deste trabalho, a aplicação de técnicas de KDD, além de possibilitar a descoberta de padrões não aparentes, pode representar um passo adiante no processo de divulgação da informação pública.

Levando em conta o ganho social que a aplicação de técnicas de mineração de dados pode trazer, para este projeto, foi escolhido fazer um recorte dos dados disponíveis no repositório sobre as eleições brasileiras, disponível no sitio do TSE (Tribunal Superior Eleitoral). Esta base de dados foi escolhida, pois está relativamente bem organizada, o que facilita sua extração e organização. Um outro fator decisivo é sua qualidade em ser de interesse coletivo, uma vez que, seus resultados diretos definem o poder executivo e legislativo. Por fim, não foi encontrada na literatura especializada (em mineração de dados especificamente) nenhum estudo que utilizasse esta base para o mesmo fim deste trabalho.

Uma vez que os dados disponibilizados pelo tribunal são referentes as eleições para cargos executivos e legislativos, o foco deste projeto será investigar a possibilidade de se encontrar características que indiquem a condição de eleito ou não eleito de um candidato.

1.1 OBJETIVO GERAL

Aplicar técnicas e ferramentas de Mineração de Dados à um repositório de dados eleitorais público, com o objetivo de estudar a presença de padrões nos dados disponíveis.

1.2 OBJETIVO ESPECÍFICOS

- Definir recortes do repositório de dados eleitorais do TSE que sejam compatíveis com as tarefas de mineração;
- Efetuar a limpeza e a normalização dos dados coletados, tratamento de instâncias com valores faltantes e de valores inconsistentes;
- Transformar os dados para formato de entrada da ferramenta de mineração de dados WEKA (formato arff);

- Aplicar as tarefas de mineração: busca de regras de associação, classificação à base de dados;
- Escolher os algoritmos adequados à tarefa de mineração; e
- Analisar e avaliar os resultados do processo

1.3 ESTRUTURA DA MONOGRAFIA

Em Referencial Teórico, foram descritos os conceitos relevantes para este trabalho. Este capítulo foi dividido em 4 itens, sendo os três primeiros diretamente ligados a descoberta de conhecimento (KDD, Mineração de dados,e Técnicas de mineração de dados) e o último sobre o paradigma de armazenamento de dados usado (Data Warehouse).

Em Estudo de Caso, encontra-se a descrição do objeto de estudo deste trabalho (Repositório de dados do TSE). Os dados disponibilizados pelo TSE foram analisados sob um vies computacional, não sendo feito nenhum tipo de julgamento sobre o domínio do problema. Neste capítulo, também encontra-se a descrição do escopo do trabalho.

Em Materiais e Métodos, são apresentados os materiais usados para esta pesquisa. Optou-se por descrever apenas os softwares usados, pois sua relevância para o projeto é muito mais significativas do que as contribuições de hardware.

Em Projeto, encontra-se a descrição da estrutura de integração dos softwares usados neste trabalho. O intuito deste capítulo é descrever como foram integradas as ferramentas escolhidas para chegar ao resultado esperado.

Em Experimento, está a descrição e os resultados dos experimentos executados. Para a classificação, além da aplicação dos algoritmos, foi feito um estudo sobre a importância dos atributos para o resultado do classificador. Quanto a tarefas de associação,

Em Conclusão, encontra-se uma discussão sobre os resultados e a metodologia empregada neste estudo. Neste capítulo encontra-se algumas sugestões para trabalhos futuros.

E, por fim, em Referencias, estão listadas as referencias usadas neste trabalho.

2 REFERENCIAL TEÓRICO

2.1 KDD

A Descoberta de Conhecimento em Banco de Dados (KDD do inglês knowledge discovery in databases) foi um termo usado pela primeira vez em 1989, no título de um workshop ocorrido na IJCAI-89 (PIATETSKY-SHAPIRO, 1991). Uma definição simples do que seria o KDD é dada por Fayyad et al. (1996b): "a descoberta de conhecimento em banco de dados é o processo não trivial de identificação de padrões no dados que sejam válidos, novos, potencialmente úteis e entendíveis (tradução nossa)". Quoniam et al. (2001) define a descoberta de conhecimento em banco de dados como o "processo da descoberta de novas correlações, padrões e tendências significativas por meio da análise minuciosa de grandes conjuntos de dados estocados". Já Galvao e Marin (2009), colocam o KDD como "o processo de extração de informação a partir de dados registrados numa base de dados, um conhecimento implícito, previamente desconhecido, potencialmente útil e compreensível". Cardoso e Machado (2008) colocam a Descoberta de Conhecimento em Banco de Dados como um "processo não trivial de identificação de padrões em um conjunto de dados", mas salientam que este processo deve ter as seguintes características: 1) validade – os novos dados devem possuir um grau de certeza ou probabilidade; 2) novidade – os padrões devem ser novos, não devem ter sido detectados por outras formas de extração de conhecimento; 3) utilidade potencial – os padrões devem ser, de alguma maneira, útil; e 4) assimilaridade – os padrões resultantes devem ser assimiláveis ao conhecimento humano.

Mesmo não sendo exatamente iguais, as definições apresentadas de KDD possuem grandes similaridades entre si. Em três delas, podemos encontrar explicitamente as ideias de uso de grande volume de dados e transformação do conjunto de dados em informação utilizável de alguma maneira. Goebel e Gruenwald (1999) oferecem uma definição do processo de KDD que, de uma forma simplista, acaba por englobar as definições anteriores: "KDD é o processo de tornar dados de baixo-nível em conhecimento de alto-nível (tradução nossa)".

Para Fayyad et al. (1996a), o KDD engloba nove passos: 1) entender o domínio do

problema: usar o conhecimento à priori sobre a questão; 2) criar o conjunto de dados para ser usado: selecionar o subconjunto de dados no qual o processo de descoberta será aplicado; 3) limpeza e pré-processamento de dados: inclui remover ruído, tratar dados faltantes e questões relacionadas ao gerenciamento do banco de dados; 4) projeção e redução de dados: preparar os dados a fim de servir como entrada para a o processamento, visando uma eficiência maior do processo; 5) escolher a função de mineração de dados: escolher o formato do modelo de mineração a ser usado: sumarização, classificação, regressão e agrupamento. 6) escolher os algoritmos de mineração: escolher os métodos que serão usados na busca por padrões; 7) Mineração de dados: procura de padrões nos dados; 8) interpretação: interpretar os resultados da mineração e, se necessário, retornar alguns passos e transformar os achados úteis em informação entendível; 9) usar o conhecimento descoberto: documentar e reportar o conhecimento gerado para os atores interessados. A figura 1, ilustra o processo, porém algumas das etapas enumeradas foram reunidas sob o mesmo nome:

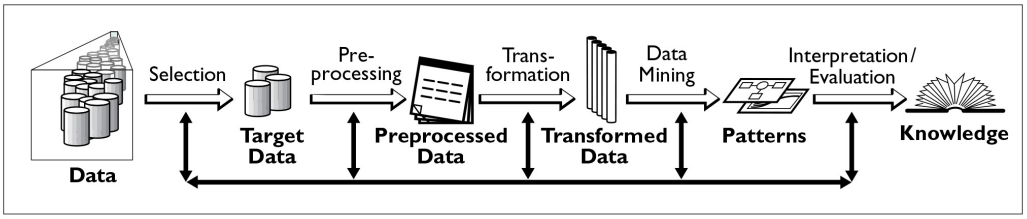

Figura 1: Esquema de mineração de dados segundo Fayyad, Piatetsky-Shapiro, e Smyth. Fonte: Fayyad et al. (1996a)

Deve-se deixar claro que, para os três autores, o KDD se refere a um processo maior que o *data mining*. Esta colocação é importante, pois muitas vezes as mineração de dados e a descoberta de conhecimento são usados como sinônimos entre si (CARDOSO; MACHADO, 2008) (GALVAO; MARIN, 2009).

2.2 MINERAÇÃO DE DADOS

A mineração de dados (do inglês *data mining*), pode ser considerada uma das etapas mais importantes do KDD. De acordo com Fayyad et al. (1996b), a mineração, seria a etapa de aplicação de algoritmos específicos para extrair padrões do dados.

Steinbach et al. (2006) dividem a mineração de dados em quatro tarefas: Modelos preditivos (classificação e regressão), Associação, Agrupamento e Detecção de anomalias. A classificação e a regressão, envolvem a construção de um modelo para uma variável-alvo que seja uma função das variáveis de entrada. Associação é usada para descobrir padrões fortemente

associados à características presentes nos dados. Agrupamento, refere-se a tarefa de procurar características que façam os itens de um grupo serem mais similares entre si em relação com as características apresentadas por outros grupos. Detecção de Anomalias, procura identificar elementos que são significativamente diferentes do resto apresentado nos dados. A seguir, serão detalhadas as quatro tarefas citadas anteriormente.

2.2.1 CLASSIFICAÇÃO

Classificação é definida, por Steinbach et al. (2006), como: "a tarefa de aprender um modelo de classificação f que associe cada atributo de uma entrada x a um rótulo de classe pré-definida y". Na classificação, a entrada de dados é uma coleção de registros. Estes registros, também chamados de instâncias ou exemplos, podem ser caracterizados como uma tupla(x, y), em que x é um conjunto de atributos e y é atributo especial chamado rótulo de classe. Em muitos casos a rotulação, isto é, a associação de uma classe a cada instância, deve ser feita de forma manual por um especialista humano. O conjunto de x de atributos pode conter valores contínuos ou discretos, enquanto o y pode receber somente valores discretos. Esta é uma característica chave que distingue a classificação de dados da regressão, uma vez que nesta, a variável y, assume valores contínuos (STEINBACH et al., 2006).

De acordo com (STEINBACH et al., 2006), o modelo de classificação pode ser descritivo ou preditivo. O modelo descritivo serve como uma ferramenta que explica quais atributos podem definir uma classe em um conjunto de instancias. O modelo preditivo, tentar "prever" a classe a qual pertence uma nova instancia que não fazia parte do conjunto inicial de exemplos x. Neste modelo, o conjunto original é usado como referência para a tarefa de classificação.

De uma maneira geral, um problema de classificação possui: um conjunto de treinamento, um algoritmo de aprendizado, e um conjunto de testes. A partir da aplicação do algoritmo sobre a base de treinamento se obtém um procedimento ou modelo de classificação. Este modelo quando aplicado à base de teste produz uma matriz de confusão sobre a qual podem ser computadas diversas métricas de performance, tais como a taxa de erro (STEINBACH et al., 2006). Tanto a bases de treinamento quanto a de testes devem estar previamente rotuladas.

O modelo gerado pelo algoritmo deve "encaixar" bem no conjunto de entrada de dados e prever corretamente a classe a qual pertence uma entrada nunca vistas antes (STEINBACH et al., 2006). Um bom objetivo de um algoritmo de aprendizado, é construir modelos com uma boa capacidade de generalização, ou seja, modelos que predizem com certa confiabilidade um rótulo de classe para uma dada entrada.

Erros cometidos por um modelo de classificação, de acordo com Steinbach et al. (2006), podem ser de dois tipos: erros de treinamento e erros de generalização. Erros de treinamento correspondem aos erros de classificação do modelo quando aplicado ao conjunto de treinamento. Já o erro de generalização é o erro encontrado quando o modelo de classificação é aplicado a instâncias desconhecidas. Quando um modelo possui uma taxa de erros de treinamento e erros de generalização altos, ele se encontra em uma situação chamada de model *underffiting*. A situação oposta ao *underffiting*, é chamada de model *overffiting*. No *overffiting*, o modelo ter um erro de treinamento muito baixo enquanto possui um erro de generalização alto, ou seja, o modelo se adapta bem demais ao conjunto de treino Steinbach et al. (2006).

Um conjunto de treinamento é composto por entradas conhecidas e rotuladas dentro de um conjunto finito de classes. Este conjunto é usado para construir o modelo de classificação, que será aplicado posteriormente a um conjunto de entradas sem rótulos conhecidos. O segundo conjunto é chamado de conjunto de teste (STEINBACH et al., 2006).

Uma forma de medir a performance de um modelo de classificação é contar quantas entradas do conjunto de testes o modelo conseguiu prever corretamente e incorretamente. A tabulação destes dados em forma de matriz, pode ser chamada de matriz de confusão. Nas linhas da matriz, temos as classes das instâncias de teste, e as colunas representam a classe que foi predita pelo modelo. Assim, podemos dizer que cada elemento e_{ij} da matriz, possui a quantidade de entradas do conjunto de testes que fazem parte da classes i, mas que foram classificadas como j. Ou seja, a diagonal da matriz de confusão mostra quantas instâncias foram classificadas corretamente, enquanto as outras, representam as classificações incorretas (STEINBACH et al., 2006).

	classe 1 (classificação)	classe 2 (classificação)
classe 1 (instância)	verdadeiro positivo	falso negativo
classe 2 (instância)	falso positivo	verdadeiro negativo

Figura 2: Matriz de Confusão

A matriz de confusão é útil para avaliar um modelo de classificação de acordo com sua eficiência. Porém, para comparar um modelo com outros, métricas que sumarizam as informações sobre o desempenho do modelo em um único valor podem ser usadas. A precisão, definida por Steinbach et al. (2006) como a razão entre o número de previsões corretas sobre o número total de previsões. Uma outra medida que pode ser usada, é a taxa de erro, definida por

Steinbach et al. (2006) como a razão entre o número de previsões erradas sobre o total de previsões. Técnicas de classificação são apropriadas para predizer ou descrever conjunto de dados que podem ser classificados de forma binária ou nominal. Elas não são muito indicadas para lidar com rótulos de classe que possuem um relacionamento (ou ordenamento) natural entre si. Por exemplo, valores que possuem uma hierarquia (alto, médio e baixo) e classes que possuem subclasses (homens e macacos são primatas, e estes subclasse de mamíferos) (STEINBACH et al., 2006).

2.2.2 ASSOCIAÇÃO

A associação pode ser útil para descobrir relacionamentos interessantes e escondidos em transações presentes em grandes conjuntos de dados. Estas associações podem ser representadas como Regras de Associação ou um conjunto de itens frequentes (STEINBACH et al., 2006). Formalmente, uma tarefa de associação possui um conjunto de itens $I = i_1, i_2, i_3, ..., i_n$ e um conjunto de transações $T = t_1, t_2, t_3, ..., t_n$. Cada transação ti, possui um subconjunto de I chamado *itemset*. Um *itemset* com k termos é chamado de *k-itemset*. Por exemplo, o *itemset* {Cerveja, Fraldas, Leite} é considerado um 3-*itemset*. A largura de uma transação é definida pelo número de itens na transação. Uma medida importante do *itemset* é o *suport count*, que se refere ao número de transações que contém um dado *itemset*. Uma regra de associação pode ser definida, segundo Steinbach et al. (2006), como uma implicação de X para Y, tal que X e Y seja disjuntos. A força de uma regra de associação pode ser medida em termos de suporte e confiança. Suporte, determina o quão comum uma regra aparece em um item set, enquanto confiança determina o quão frequente itens em Y aparecem em transações contendo X. Matematicamente, pode-se definir:

Suporte, $S(X \rightarrow Y) = \sigma(X \cup Y)/N$

Confiança, $C(X \rightarrow Y) = \sigma(X \cup Y)/\sigma(X)$

Uma regra de associação com um suporte muito baixo, pode ser um indício de que a associação ocorre apenas ao acaso. Em uma outra perspectiva, o suporte baixo pode não ser interessante do ponto de vista do modelo de negócios, pois o custo em promover algum tipo de política para uma regra com suporte baixo pode não ser muito diferente do custo de promover uma regra com suporte alto. Dada estas razões, regras com suporte muito baixos são descartadas por ser irrelevantes para o processo de descoberta de regras. Já a confiança, mede o quanto é confiável a inferência feita pela regra. Para uma regra de X implica em Y, quanto maior a confiança, maior é chance de Y aparecer nas transações que contém X. De certa maneira, a confiança mostra a probabilidade de Y, dado X Steinbach et al. (2006). A tarefa de

mineração de regras de associação, de acordo com (STEINBACH et al., 2006, p. 330) pode ser definida formalmente como: "dado um conjunto de transações T, achar todas as regras que tenham suporte maior ou igual a *minsup* e confiança maior ou igual a *minconf*, em que *minsup* e *minconf* são parâmetros que correspondem aos limiares de suporte e confiança." (tradução nossa)

Ao se aplicar uma tarefa de mineração de regras de associação em grande conjuntos de dados, de acordo com Steinbach et al. (2006), devemos levar em conta duas questões: primeiro, a descoberta de padrões escondidos em transações dentro de grandes volumes de dados pode ter um custo computacional alto, e segundo, alguns dos padrões encontrados podem não ser relevantes, apenas ocorrendo ao acaso. Uma estratégia comum dos algoritmos de mineração de regras de associação é dividir o problema em duas partes: criação dos *itemsets* frequentes e criação de regras. Na criação de *itemsets* frequentes, o objetivo é achar todos os *itemsets* que satisfaçam a condição de possuir uma confiança maior que o *minconf*. Já na criação de regras, o objetivo é extrair todas as regras com alta confiança dos *itemsets* gerados na criação de *itemsets* frequentes. Estas regras são chamadas de regras fortes (STEINBACH et al., 2006).

2.2.3 AGRUPAMENTO

O agrupamento (do inglês, *Clustering*) deve separar os dados em grupos (ou clusters) levando em conta somente as descrições dos dados e suas relações. O objetivo do agrupamento é de que cada objeto dentro de um grupo seja similar à outros objetos do grupo e, ao mesmo tempo, diferente de objetos de outros grupos (STEINBACH et al., 2006). Pode-se entender o agrupamento como uma forma de classificação na qual os rótulos de classe (grupos) são criados pelo processo e não conhecidos a priori (como ocorre na classificação). Como a sua forma de classificação não requer conhecimento prévio das categorias a serem classificadas, o agrupamento é visto como uma forma de classificação não supervisionada, em contrapartida a classificação apresentada anteriormente, que pode ser chamada de classificação supervisionada (STEINBACH et al., 2006).

De acordo com Steinbach et al. (2006), existem várias formas de agrupamentos: hierárquico ou particionado; exclusivo, sobreposto ou nebuloso; completo ou parcial. Um conjunto de grupos é considerado particionado quando seus objetos pertencem cada um a um grupo somente. Caso um conjunto de grupos possua subgrupos, ele é considerado hierárquico. O agrupamento hierárquico é apresentado sob forma de árvore, em que cada nó é um grupo, e que, normalmente, cada folha é um objeto ou um grupo unitário com um objeto.

Quando cada objeto é colocado dentro de um grupo, não existindo ao mesmo tempo em

mais de um, podemos chamar o agrupamento de exclusivo. Caso existam objetos que pertençam a mais de um cluster, o agrupamento deixa de ser exclusivo e passa a ser sobreposto, pois existe uma sobreposição dos grupos. O agrupamento nebuloso define que um objeto pertence a um grupo sob uma condição de pertencimento que pode assumir valores entre 0 e 1. Uma imposição adicional é de que a soma de todos os pertencimento do objeto sejam iguais a 1. Este tipo de agrupamento se mostra interessante para evitar classificações arbitrárias assumindo que o objeto é próximo de um grupo, quando na verdade é de vários (STEINBACH et al., 2006).

Um agrupamento é chamado de completo quando relaciona cada objeto a um clusters pelo menos, enquanto o parcial não o faz. Um dos motivos para não assegurar um grupo para um objeto, esta na condição deste em não ser bem definido dentro do conjunto de dados analisado, podendo ser mais propenso até mesmo a ser um ruído (STEINBACH et al., 2006).

2.2.4 DETECÇÃO DE ANOMALIAS

Na detecção de anomalias, o objetivo da tarefa é encontrar objetos que sejam diferentes de outros objetos. Esta tarefa é aplicada em detecção de fraudes, detecção de invasões em sistemas computacionais e distúrbios em ecossistemas.

Algumas causas comuns de anomalias, de acordo com Steinbach et al. (2006) são: dados de classes diferentes, variação natural e erros de medição. Um objeto pode ser diferente de outro por ser um objeto de uma classe diferente. Por exemplo, na detecção de fraude em cartões de crédito, um conjunto de usos normal pertence à uma classe diferente de um conjunto fraudulento de usos. Assim, um objeto que venha de uma classe diferente da classe que a maioria dos objetos em um conjunto de dados, pode ser considerado uma anomalia ou um objeto atípico (*outlier*) (STEINBACH et al., 2006).

Por variação natural, entende-se que os dados são obtidos de processos aleatórios, de forma que o conjunto de dados pode ser modelado por uma distribuição estatística. Erros de medição ou erros de coleta de dados, podem ser uma outra fonte de anomalias. Nesta categoria, podemos colocar: erros humanos, erros de equipamentos de coleta de dados e presença de ruído. O objetivo nestes casos é eliminar as anomalias para que o conjunto de dados possa ser melhor aproveitado. Normalmente, esta é uma tarefa de pré-processamento de dados e é comum ser associada à limpeza de dados. (STEINBACH et al., 2006).

Para Steinbach et al. (2006), existem três formas básicas para encarar um problema de detecção de anomalias: supervisionado, não-supervisionado e semi-supervisionado. Na detecção supervisionada, é preciso usar um conjunto de treino que contenha tanto instancias

anômalas quanto normais. Todas as instancias anômalas não necessariamente são da mesma classe, pois nada impede que exista mais de um tipo de anomalia. A detecção supervisionada de anomalias é semelhante um esquema de classificação para classes raras (STEINBACH et al., 2006).

A detecção de anomalias não-supervisionada é muito usada quando não se tem informações sobre as classes das instancias. Neste casos, é atribuído um valor para cada instancia que representa o seu grau de anormalidade. Deve-se tormar cuidado com a presença de muitas anomalias similares, pois em conjunto podem influenciar a a atribuição de grau (STEINBACH et al., 2006).

A detecção semi-supervisionada consiste em um conjunto de treino classificado, porém sem nenhuma informação sobre suas instancias anômalas. O objetivo é definir um valor ou classificar instancias anômalas com informação sobre instancias normais (STEINBACH et al., 2006).

2.3 ALGORITMOS PARA MINERAÇÃO DE DADOS

Nesta seção serão apresentados conceitos ligados diretamente aos métodos usados nesta pesquisa. Como será apresentado mais adiante, foi escolhido trabalhar somente com associação e classificação. Para a classificação serão aplicadas as técnicas conhecidas como C4.5, K-nn, Perceptron de Multiplas Camadas, Classificador Bayesiano Ingênuo e Máquina de Suporte Vetorial. Para a associação, será usado somente um método chamado Apriori.

2.3.1 C4.5

Uma forma intuitiva de classificação de um padrão, é uma sequência de questões na qual a próxima pergunta depende da resposta da atual. Esta sequência de questões podem ser modeladas sob a forma de uma Árvore de Decisão (DUDA et al., 2001)

Uma Árvore de Decisão, pode ser definida como uma forma específica de árvore, que, de acordo com Steinbach et al. (2006), possui três tipos de nós: 1) um nó raiz, que não possui arestas chegando nele, e possui zero ou mais arestas saindo de si. 2) nós internos, que possuem exatamente um nó de chegada e dois ou mais nós de saída. 3) folhas ou nós terminais, que tem exatamente uma aresta de chegada e nenhuma de saída.

Em uma árvore de decisão, cada nó folha representa um rótulo de classe possível do conjunto de dados. Todos os nós que não são folhas (raiz e internos) contém testes que separam

as entradas de acordo com suas características (STEINBACH et al., 2006). O algoritmo C4.5 pertence a uma sucessão de classificadores baseados em árvores de decisão que remetem ao trabalho de Hunt, do começo dos anos 60. Seus predecessores são os algoritmos ID3 e C4. Existe já uma versão nova do algoritmo chamado de C5, porém este está disponível somente através da RuleQuest Research (KOHAVI; QUINLAN, 1999).

De acordo com Kohavi e Quinlan (1999), a entrada do C4.5 consiste em uma coleção de instancias de treinamento, que possuem atributos (ou variáveis independentes) as quais podem ser definidas como tuplas $A = A_1, A_2, ..., A_k$ e uma classe (ou variável dependente). Os atributos de uma instância podem ser contínuos ou discretos. A classe é uma variável discreta C, podendo assumir valores os valores $C_1, C_2, ..., C_z$. O objetivo do treinamento com as instâncias é aprender uma função que mapeie os valores dos atributos a um conjunto de classes.

A maioria dos algoritmos baseados em arvores de decisão possuem duas fases: construção da árvore de decisão e simplificação da árvore de decisão. Existem duas operações principais na construção de uma árvore de decisão: 1) Avaliação dos pontos de separação de cada nó interno da árvore e identificação do melhor ponto de separação. 2) a criação de partições usando o melhor ponto de separação Goldschmidt e Passos (2005) . Para Kohavi e Quinlan (1999), a estratégia de construção da árvore de decisão do C4.5 usa de busca gulosa para selecionar um teste candidato que maximize uma heurística para a escolha do critério de separação. Dois critérios são usados para tanto: ganho de informação e razão de ganho. O Ganho de informação considerando a partição de dados associada ao nó de análise, pode ser definido como Goldschmidt e Passos (2005):

$$info(S) = -\sum_{j=1}^{k} \frac{freq(C_j,S)}{|S|} \times log_2 \left(\frac{freq(C_j,S)}{|S|} \right)$$

Onde, S - representa a partição da base de dados; $freq(C_j, S)$ - representa o número de vezes em que a classe C_j acontece em S; $|S|$ - denota o número de casos do conjunto S; k - indica o número de classes distintas. O Ganho de informação de cada atributo, considerando a partição de dados associada ao nó de análise, pode ser definido como Goldschmidt e Passos (2005):

$$info_x(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times info(T_i)$$

Onde, T – representa a quantidade de ocorrências na partição em análise; T_i - representa a quantidade de ocorrências de uma classe contidas no conjunto T. A seleção do atributo que resulta em maior ganho de informação é dada por:

$$gain(X) = info(T) - info_x(T)$$

Em que gain representa a razão de ganho de informação. Ao escolher o atributo que

irá particionar a base (aquele com o maior ganho de informação), a base é particionada e o processo é repetido a cada novo nó gerado Goldschmidt e Passos (2005).

2.3.2 PERCEPTRON DE MÚLTIPLAS CAMADAS

Um Perceptron de Múltiplas Camadas (MLP do inglês *Multi Layer Perceptron*) é um modelo de arquitetura de rede que faz parte de uma família de algoritmos e estruturas de dados inspirados em sistemas neurais biológicos. Estes modelos, junto com certos paradigmas de aprendizado, são elementos que fazem parte de uma área de estudo conhecida como Redes Neurais Artificiais.

As Redes Neurais Artificiais oferecem um modelo computacional diferente do modelo oferecido pelas Máquinas de von Neumann. A arquitetura de uma máquina de von Neumann possui um bom desempenho em computação numérica e manipulação simbólica, ao mesmo tempo que não se mostra apropriado para executar algumas tarefas triviais para o cérebro humano. Uma vez que as Redes Neurais são baseadas em um outro tipo de paradigma diferente (um similar ao cérebro humano), espera-se que elas resolvam problemas que uma máquina de von Neumann tenha dificuldades. Entre as características de redes neurais, podemos destacar: acentuado paralelismo, computação distribuída, capacidade de aprendizado, capacidade de generalização, adaptividade, capacidade de processamento de informação contextual inerente, tolerância a falhas e baixo consumo de enegia (JAIN; MAO, 1996).

Haykin (1999, p. 28) define uma rede neural artificial como:

> "... é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos: 1) o conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem. 2) forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizados para armazenar o conhecimento adquirido.

De um ponto de vista conceitual, a arquitetura de uma rede neural artificial, pode ser vista como um grafo conectado através de arestas ponderadas. Assim, podemos dividir as redes neurais em dois grandes grupos: *feed-foward networks* (modelo do qual o Perceptron de Múltiplas Camadas faz parte), em que não existem ciclos no grafo e *feedback networks*, em que os ciclos nos grafos representam os *feedbacks* de informação (JAIN; MAO, 1996).

A arquitetura de um Perceptron de Múltiplas Camadas, tipicamente, consiste em um conjunto de unidades sensoriais divididas em uma camada de entrada, uma ou mais camadas

ocultas de nós computacionais e uma camada de saída de nós computacionais. O sinal de entrada se propaga para frente, camada por camada (HAYKIN, 1999).

Em um MLP, cada neurônio da rede inclui uma função de ativação não-linear e derivável. Um neurônio pode ser definido como uma unidade que computa uma soma ponderada de sinais de entrada e gera uma saída de acordo. Matematicamente, pode ser definido da seguinte maneira (adotando que uma entrada para a rede é um vetor $x_i \in X$ com n termos):

$y = \theta\left(\sum_{j=1}^{n} w_j x_j - u\right)$, em que y é a saída do neurônio, θ é a função de ativação do neurônio, w_j é o peso associado ao j-ésimo termo e x_j é o valor do $j-$ésimo e u o limiar de ativação. Dentre as várias funções de ativação possíveis, uma das mais utilizadas é a função sigmóide: $\frac{1}{1+e^{-\beta x}}$ (JAIN; MAO, 1996).

Existem três tipos de camadas de neurônios em um MLP. A camada de entrada, é responsável por receber os valores dos sinal usados na rede neural. Já as camadas ocultas (uma rede pode conter uma ou mais camadas ocultas) permitem a rede aprender padrões significativosa, podendo armazená-los em sua organização. A camada de saída é responsável por devolver os resultados do sinal de entrada. Uma outra característica importante de Perceptron de Múltiplas Camadas, é seu alto grau de conectividade, determinado pelas sinapses da rede (HAYKIN, 1999).

Este tipo de rede tem sido aplicado com certo sucesso na resolução de problemas difíceis, através do seu treinamento de forma supervisionada com um algoritmo conhecido como "algoritmo de retropropagação de erro (do inglês, *error back-propagation*). Basicamente, este algoritmo possui dois passos: um para frente, a propagação, e um para trás a retropropagação. No primeiro, um sinal é propagado camada por camada até gerar suas saídas. Neste processo os pesos dos neurônios são todos fixos. No segundo passo, os pesos sinápticos são todos ajustados de acordo com uma regra de correção de erro (HAYKIN, 1999).

O sinal de erro na saída do neurônio j pode ser definido como:

$e_j = d_j - y_j$. Onde d_j é o valor esperado na saída do neurônio e y_j é o sinal no estado atual. Com esta equação, pode-se definir o erro instantâneo para um neurônio j como $\frac{1}{2}e_j^2$. Consequentemente, podemos obter o erro total da camada de saída da rede somando todos os erros instantâneos de seus neurônios: $E = \frac{1}{2}\sum_{j\in C} e_j^2$ (HAYKIN, 1999).

O campo local induzido na entrada do neurônio j pode ser escrito como: $v_j = \sum_{i=0}^{m} w_{ji} y_i$. Onde m é a quantidade de entradas associadas ao neurônio j. Esta equação permite deduzir que a saída associada ao neurônio j seja: $y_j = \theta(v_j)$ (HAYKIN, 1999).

A correção Δw aplicada ao peso sináptico w_{ij} é proporcional a derivada parcial $\frac{\partial E}{\partial w_{ij}}$,

sendo definida como: $\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$, onde η é o parâmetro da taxa de aprendizagem do algoritmo de retropropagação. Resolvendo a derivada parcial $\frac{\partial E}{\partial w_{ij}}$, pode-se rescrever a função delta da seguinte maneira: $\Delta w_{ij} = \eta \delta_j y_j$, onde $\delta_j = e_j \theta'(v_j)$ (HAYKIN, 1999).

Desta maneira, pode-se identificar dois casos possíveis para o ajuste sinóptico: quando o neurônio está na camada de saída ou em uma camada escondida. O caso da camada de entrada não é levado em conta, pois não existem pesos associados as entradas, uma vez que estes são os valores das instancias em si (HAYKIN, 1999) (JAIN; MAO, 1996).

Para o caso de um neurônio j estar na camada de saída da rede, uma vez obtido o valor de e_j, pode-se calcular diretamente o valor de Δw_j, através do gradiente local δ_j. Já para o caso do neurônio j estar em uma camada oculta, o calculo de Δw_j é um pouco mais complicado, pois não se pode obter diretamente o valor de e_j.

Seja k um neurônio da camada posterior aquela da qual j faz parte, o calculo do diferencial δ_j é dado pela seguinte equação: $\delta_j = \theta'(v_j \sum_k \delta_k w_{kj})$. É importante notar que o δ_j, quando o neurônio faz parte de uma camada oculta, depende da soma ponderada dos δ dos neurônios da camada posterior a sua, para os quais j é uma conexão de origem.

Jain e Mao (1996), apresenta sete passos resumidos para a execução do algoritmo de retropropagação:

1. inicializar os pesos com pequenos valores randômicos;
2. escolher aleatoriamente um uma entrada $x_n \in X$;
3. propagar o sinal para frente através da rede;
4. computar o valor de delta na camada de saída da rede δ_k, onde k é um neurônio da última camada;
5. computar o valor de delta das camadas escondidas, propagando o erro para trás δ_j, onde j é um neurônio de uma camada escondida;
6. atualizar os pesos ΔW_{ij};
7. repetir o passo 2 em diante até o erro se tornar menor que eu valor específico, ou o máximo de iterações for alcançado.

2.3.3 K-NN

O K-NN (k-Nearest Neighbors ou em português, K-Vizinhos mais próximos) é considerado um método de classificação baseado em instâncias. É classificado desta maneira, pois, em seu funcionamento são levados em conta as instâncias já registradas na base de dados de referência (GOLDSCHMIDT; PASSOS, 2005).

O seu funcionamento é descrito da seguinte maneira, por Goldschmidt e Passos (2005) Goldschmidt e Passos (2005): 1) Calcula-se a distância do novo registro a cada um dos registros existentes na base de dados de referencia. A métrica de distância deve ser definida antes da aplicação do método. 2) São identificados os k registros mais próximos do novo registro (o valor de k deve ser escolhido previamente). 3) Determina a classe mais frequente nos itens identificados no item 2 e esta classe será a atribuída ao novo registro. 4) Comparação entre o registro real e o registro atribuído pelo classificador. Este item somente existe se o registro tiver uma classe atribuída previamente e seja objetivo da tarefa avaliar o desempenho do classificador. Caso contrário, o método se encerra no item 3.

Para Steinbach et al. (2006) Steinbach, Tan e Kumar (2006), o K-NN, tem como caracteristicas os seguintes itens: 1) O K-NN faz parte de um tipo de técnica mais geral conhecida como “Aprendizado Baseado em Instâncias”. Este grupo de técnicas não precisam manter um modelo abstrato derivado dos dados disponíveis. 2) Sendo o K-NN um “aprendiz preguiçoso”, ele não requer um modelo classificatório. Porém, a tarefa de classificação pode ser custosa, devido ao fato de que o algoritmo exige uma comparação da nova instância com as outras já existentes. 3) O K-Vizinhos mais próximos faz suas predições baseadas em informações locais, enquanto outros tipos de classificadores (árvores de decisão ou classificadores baseados em regras) o fazem basedos em informações globais. Por causa da forma como lida com este “espaço”, o K-NN se mostra sensível a ruído. 4) K-NN pode criar limites de decisão com formas arbitrárias se comparado com outros tipos de classificadores como árvores de decisão ou classificadores baseados em regras. Seus limites de decisão também são bem variados devido a sua sensibilidade ao conjunto de treino usado. Uma forma de diminuir essa variação, é usar valores de k mais altos. 5) Métodos baseados na vizinhança mais próxima, podem gerar classificações errôneas se os dados não forem tratados corretamente e se o valor de k não for adequado. Variáveis com valores numéricos muito diferentes, podem acabar por “dominar” o processo de medição de distância. É recomendável a normalização das entradas para evitar um problema do gênero.

2.3.4 CLASSIFICADOR BAYESIANO INGÊNUO

O classificador bayesiano ingênuo (do inglês, Naive-Bayes), é considerado um método de mineração de dados fundamentado em princípios estatísticos. Dentre os vários métodos derivados da estatística, este, em especial, como seu nome indica, é derivado do Teorema de Bayes (GOLDSCHMIDT; PASSOS, 2005). O Teorema de Bayes, tem a seguinte formulação:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Onde, $P(Y|X)$ é a probabilidade de um evento pertencente à Y ocorrer, caso X tenha ocorrido. $P(X|Y)$ é a probabilidade do evento X ocorrer, dado a ocorrência do evento Y. $P(Y)$ a probabilidade independente de Y ocorrer e $P(X)$ a probabilidade do evento X ocorrer (STEINBACH et al., 2006). A noção do significado do teorema de Bayes pode ser apresentada também da seguinte maneira (DUDA et al., 2001) (DUDA, HART e STORCK 2001):

$$prob.aposteriori = \frac{prob.condicional \times prob.apiori}{evidencia}$$

Onde, a probabilidade a posteriori é a probabilidade alterada com as informações sobre as condições especificadas. A probabilidade condicional é a probabilidade inversa da posteriori, ou seja inverte-se a relação causa e efeito, admitindo que existe uma relação entre ambas. A probabilidade a priori não é nada mais do que a probabilidade de Y ocorrer independente de outras condições e por fim, evidência é um fator para normalizar a probabilidade a posteriori, de forma que a soma de todas suas possibilidades somem 1 (DUDA et al., 2001) (DUDA, HART e STORCK 2001). Baseado nestas noções, o classificador bayesiano ingênuo estima a probabilidade condicional de uma dada classe, assumindo que seus atributos são condicionalmente independentes (STEINBACH et al., 2006). A condição de independência, assumida pelo classificador, pode ser definida como:

$$P(X|Y = y) = \prod_{i=2}^{d} P(X_i|Y = y)$$

Onde, Y é o conjunto de rótulos de classes e X o conjunto de atributos composto de d atributos (STEINBACH et al., 2006). Ou seja, o classificador calcula a probabilidade de uma entrada pertencer a uma dada classe y, levando em conta as probabilidade dos atributos encontrados nas instancias vistas.

Para atributos nominais ou categóricos, a probabilidade condicional $P(Xi = xi|Y = y)$, é estimada de acordo com a fração de instancias classificadas como y, que possuem o atributo x_i. Para atributos contínuos, existem duas maneiras de se estimar a probabilidade condicional dos atributos: 1) pode-se discretizar os valores, e aplicar o mesmo processo para os atributos categóricos. Porém, aa probabilidade condicional é sensível a estratégia de discrtização esco-

lhida (ver Redução de Dados Horizontal). 2) Pode-se assumir que os dados respeitam uma dada distribuição, como por exemplo uma distribuição gaussiana e usar o conjunto de treino para estimar seus parâmetros. Com posse destes dados, pode-se estimar a probabilidade condicional (STEINBACH et al., 2006).

Steinbach et al. (2006) Steinbach, Tan e Kumar (2006), definem três caracteristicas que um classificador bayesiano ingênuo possui: 1) São robustos quanto a valores não disponíveis e ruídos em atributos isolados, pois ao estimar a probabilidade condicional dos dados, ambas caracteristicas são “abafadas”. 2) São robustos a valores irrelevantes. Se Xi for um valor atributo irrelevante, então P(Xi|Y) se torna quase uniformemente distribuido. 3) Atributos correlacionados diminuem a eficiência de um classificador bayesiano ingênuo, pois violam a premissa da independência dos atributos.

2.3.5 MÁQUINA DE SUPORTE VETORIAL

A ideia geral de uma máquina de vetores de suporte (SVM, do inglês support vector machine) é construir um hiperplano que funcione como uma superfície de decisão. Esse hiperplano deve ser construído de uma forma a maximizar a margem que separa os padrões classificados com rótulos diferentes durante a fase de treino.

Sendo o conjunto de treino definido como: $D = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, onde x_i é um vetor $n-$dimensional e y_i pode assumir os galres de 1 ou -1 indicando a classe para qual x_i pertence (y_i funciona como um rótulo de classe, porém dada a base matemática da forma como funciona uma máquina de vetor de suporte). A função de classificação de uma SVM $F(x)$ tem a seguinte forma: $F(x) = w \cdot x - b$. Onde, w é o peso do vetor e b é um "bias"que será computado pelos SVM no processo (HWANJO; KIM, 2012).

Pode-se escrever a condição de classificação da seguinte maneira: $y_1(w \cdot x_i - b) >= 0, \forall (x_i, y_i) \in D$. Se existir uma função que classifique todos os pontos corretamente, pode-se dizer que o conjunto D é linearmente separável. Como a distância de um ponto x_i até o hiperplano de separação ($F(x)$) é dada por: $\frac{|F(x_i)|}{||w||}$, pode-se reformular a condição de classificação da seguinte maneira: $y_1(w \cdot x_i - b) >= 1, \forall (x_i, y_i) \in D$, fazendo com que a margem de separação (distância do hiperplano até o ponto mais próximo de uma dada classe) seja dado por $\frac{1}{||w||}$, pois quando x_i for o mais próximo, a função irá retornar 1 (HWANJO; KIM, 2012).

Duda et al. (2001) definem os vetores de suporte como as instâncias de treino que definem o hiperplano de separação ótimo (ou um x_i para o qual $F(x)$ retorne 1). São as instâncias equidistantes do hiperplano e que permitem maximizar a margem de separação. Também são

os padrões mais difíceis de se classificar.

Ao usar os multiplicadores de Lagrange para maximizar a margem, chega-se na seguinte forma para a função de classificação: $F(x) = \sum_i \alpha_i y_i x_i \cdot x - b$. Onde α são os multiplicadores de Lagrange. Porém, para trabalhar com um conjunto D que não seja linearmente separável, é preciso adaptar a forma da classificação para que possam ser minimizadas as quantidades de entradas classificadas erroneamente.

Com um mapeamento apropriado para uma dimensão suficientemente maior, quaisquer dados de duas categorias podem ser separados linearmente por um hiperplano (DUDA et al., 2001). A fim de evitar o mapeamento de todas as instâncias de treino para o novo espaço dimensional, operação custosa de um ponto de vista computacional, é usada uma técnica chamada de truque de núcleo (*kernel trick*).

A base do truque de núcleo é o teorema de Mercer. Este teorema afirma que uma função núcleo pode sempre ser expressada como um produto interno de uma dimensão qualquer. Assim, ao invés de se calcular o produto interno em um espaço dimensional maior, calcula-se somente a operação do kernel, que pode ser usada como substituta para a operação de otimização usada. Desta maneira, pode-se rescrever a função de classificação usando uma função kernel: $F(x) = \sum i \alpha_i y_i K(x_i, x) - b$. Como $K(\cdot)$ é calculado no espaço original, não é necessário executar a transformação de D para o espaço vetorial destino explicitamente. Existem três funções consideradas núcleos populares: polinomial $K(a,b) = (a \cdot b + 1)$, radial basis function(rbf) $K(a,b) = exp(-\gamma \|a-b\|^2)$, sigmoid $K(a,b) = tanh(ka \cdot b + c)$ (HWANJO; KIM, 2012).

Tanto o truque de núcleo quanto a maximização da margem de separação do hiperplano são duas características importantes do funcionamento de uma SVM. Um outro fator relevante sobre máquinas de vetor de suporte, é sua habilidade em não precisar do conhecimento do domínio do problema para funcionar (HAYKIN, 1999).

Originalmente, uma SVM é uma máquina classificadora modelada para problemas de classificação binários(lineares). Sua generalização para problemas com múltiplas classes, pode ser alcançada com a combinação de vários classificadores binários (HWANJO; KIM, 2012).

2.3.6 APRIORI

Dos métodos descritos nesta seção, o Apriori é o único que não é um algoritmo de classificação. Na verdade, o Apriori é um método clássico usado para mineração de regras de associação, inclusive é a base de diversos outros métodos tais como: GSP, DHP, Partition, DIC, Eclat, MaxEclat, Clique e MaxClique. (GOLDSCHMIDT; PASSOS, 2005)

Este método é baseado no seguinte princípio de mesmo nome: “Se um *itemset* é frequente, então todos os seus *subsets* devem ser também frequentes.” (STEINBACH et al., 2006, p. 333). Esta definição permite evitar a exploração de boa parte do espaço de busca, pois ao definir um itemset como infrequente, todas as suas derivações também o serão, implicando que tais itemsets não precisam ser visitados. Esta estratégia somente é possível devido a propriedade chamada de antimonotonicidade presente na medição do suporte. Steinbach et al. (2006) Steinbach, Tan e Kumar (2005) definem a antimonotonicidade da seguinte maneira:

Seja I um conjunto de itens e $J = 2^I$ o conjunto de todas as possíveis combinações de I, então a medida f é monotônica se: $\forall X, Y \in J : (X \subseteq Y) \rightarrow f(X) \leq f(Y)$ O que implica que f será anti-monotônica se: $\forall X, Y \in J : (X \subseteq Y) \rightarrow f(Y) \leq f(X)$

Desta maneira, qualquer medida que possuía a propriedade de antimonotonicidade, pode ser incorporada no método de mineração para diminuir o espaço de busca (STEINBACH et al., 2006). Esta é a definição formal da heurística descrita como o princípio Apriori. O algoritmo Apriori, pode ser decomposto em duas etapas: 1) Encontrar todos os conjuntos de itens frequentes. 2) A partir dos conjuntos frequentes, gerar as regras de associação(GOLDSCHMIDT; PASSOS, 2005).

Inicialmente o usuário, deve definir os valores de suporte e confiança mínimos (ambos definidos no item 2.2.2) a serem considerados pelo Apriori. Com base nestes valores, o algoritmo define os itens sets mais frequentes e depois gera as regras de associação(GOLDSCHMIDT; PASSOS, 2005).

Steinbach et al. (2006) Steinbach, Tan e Kumar (2005) afirmam que o Apriori age da seguinte maneira para gerar os conjuntos de itemsets frequentes: Seja C_k o conjunto de candidatos k-itemsets e F_k o conjunto frequente k-$itemsets$, primeiramente, o algoritmo “passa” por todos os dados definindo o suporte para cada um deles. Após o fim deste passo, o conjunto de todos os 1-$itemsets$ frequentes, $F1$, serão conhecidos. Depois, serão gerados, de forma iterativa, novos candidatos k-$itemsets$ usando os (k-1)-$itemsets$ frequentes encontrados em uma iteração anterior. Para contar o suporte dos candidatos, o algoritmo é obrigado a “passar” pelos dados mais uma vez. Após contar os suportes, são eliminados todos os $itemsets$ candidatos que possuem um valor de suporte menor que o definido como mínimo no começo do processo. O algoritmo termina quando não existem mais $itemsets$ candidatos a serem gerados.

Existem duas características que são importantes no processo de gerar os conjuntos de itemsets frequentes do Apriori. Primeiro, o processo é level-wise, ou seja ele atravessa cada nível do reticulado por vez. Segundo, ele usa uma estratégia de “gerar e testar”. Nada mais do que o processo de criar novos itemsets, baseados nos itemsets frequentes da iteração anterior

(STEINBACH et al., 2006).

Diferente da medição de suporte, a confiança não possui a propriedade de monotonicidade. Porém, pode-se usar o seguinte teorema para o seu calculo: “Se uma regra $X \rightarrow Y - X$ não satisfaz o limiar de confiança, então qualquer regra $X' \rightarrow Y - X'$, em que X' é um subconjunto de X, deve não satisfazer o limiar de confiança também.” (STEINBACH et al., 2006).

O Apriori uma abordagem level-wise para gerar as regras de associação, em que cada nível é definido pela quantidade de itens que pertencem a dada regra. O processo de gerar as regras é similar ao processo de gerar os *itemsets* candidatos, porém não é necessário o passo que computa a confiança de cada regra candidata, pois pode-se usar o a contagem de suporte definida durante a geração dos *itemsets* frequentes (STEINBACH et al., 2006).

2.4 DATA WAREHOUSE

Para este projeto, os dados a serem minerados, serão armazenados em um sistema de banco de dados relacional, mas que irá funcionar como um *data warehouse*. Um *data warehouse* é definido por Silberschatz (1999) como "um repositório de informações coletadas em diversas fontes, armazenadas sob um esquema único, em um só local”. Pode-se discordar desta afirmação, pois mesmo que vários *data warehouse* tenham realmente fontes de dados heterogêneas, este tipo de origem não é obrigatório.

Para Kimball (1996), existem seis objetivos para um *data warehouse* : 1) fornecer um acesso rápido aos dados organizacionais. 2) oferecer consistência de dados. 3) Os dados são oferecidos em diversas formas (slice e dice). 4) Além dos dados, um *data warehouse* deve fornecer ferramentas de consulta, análise e apresentação. 5) devem estar publicados somente dados confiáveis. 6) a qualidade dos dados no *data warehouse* pode impulsionar uma re-engenharia no modelo de negócios de uma organização. A definição de Kimball (1996) é direcionada ao ambiente corporativo. Desta maneira, para este projeto, o item seis dos objetivos propostos pelo autor será desconsiderada.

Uma grande diferença que entre a arquitetura de um *data warehouse* e uma base de dados regular, é quanto ao seu esquema lógico. Diferente dos sistemas de banco de dados relacionais regulares, um *datamart* pode se apresentar com um esquema lógico desnormalizado. Kimball (1996) ressalta a natureza assimétrica na qual uma tabela é dominante no esquema dimensional. Este esquema dimensional pode ser de duas formas: estrela ou floco-de-neve.

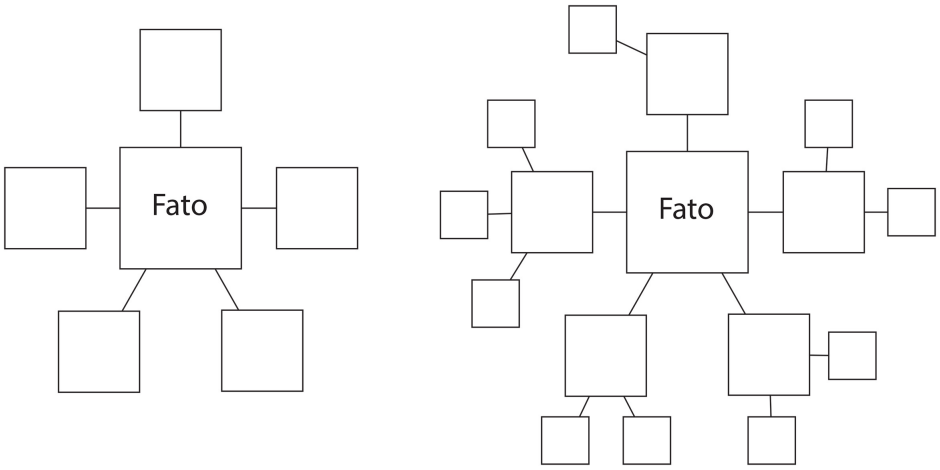

Figura 3: Primeira imagem é uma representação de um esquema Estrela e a segunda, um esquema Floco de Neve

3 ESTUDO DE CASO

Neste capítulo, será discutido as características gerais dos dados disponíveis. Apresentar a magnitude do tamanho da base de dados que o TSE disponibiliza, mostra o que existe dentro dela, assim como os dados que não estão disponibilizados. Em seguida, será feita uma análise mais técnica dos dados, de acordo com uma classificação simples proposta por Pyle, como indicado em Goldschmidt e Passos (2005). E por último, usando das informações levantadas nestas duas etapas, será definido um escopo de atuação para este trabalho.

3.1 REPOSITÓRIO DE DADOS HISTÓRICOS ELEITORAIS DO TSE

O repositório de dados eleitorais disponibilizado pelo TSE, pode ser dividido em quatro grandes grupos, de acordo com a própria organização do sítio do tribunal: Eleitorado, Candidatos, Resultados e Prestação de Contas.

Em Eleitorado, constam informações do perfil do eleitorado das eleições de 1994 à 2012, até o grão da zona eleitoral (TSE, 2013). Nesta seção também estão disponíveis informações mais atuais sobre o eleitorado, atualizadas à cada mês.

Os dados sobre o perfil do eleitorado estão disponíveis em arquivos em formato txt, separados por ano da eleição e tem o seu nome no seguinte formato: PERFIL_ELEITORADO_ <ANO ELEIÇÃO>.txt (exemplo: perfil_eleitorado_2002.txt). No campo "grau de escolaridade" desta base de dados, existe uma ressalva feita pelo próprio TSE quanto a precisão do campo. Como esta é um informação dada pelo eleitor, muitas vezes o seu valor é desatualizado. Isto se deve ao fato de que uma vez cadastrado o grau de escolaridade, o eleitor, na maioria dos casos, não tem um interesse em atualizá-la no cartório eleitoral. (TSE, 2014) Na tabela a seguir, temos a quantidade total de registros disponíveis por ano. É fácil notar que existe um "degrau" na quantidade de registros disponíveis antes e depois do ano 2000. Vale lembrar que estes são os valores dos registros disponíveis na base eleitoral do TSE, e que estes valores não necessariamente significam a um salto na quantidade de eleitores existentes entre 1998 e 2000.

Tabela 1: Quantidade de registros sobre perfil de eleitores por ano

ano	Quantidade de registros
1994	98129
1998	107194
2000	106479
2002	854195
2004	861621
2006	854455
2008	871744
2010	871940
2012	882980
total	5508737

Em Candidatos, temos dados sobre o as candidaturas, declarações de bens dos candidatos, dados sobre os partidos, coligações, vagas por cargo e unidade eleitoral. Esta seção é subdividida em: Candidatos, Bens de Candidatos, Legendas e Vagas. No subitem Candidatos, existem dados referentes a candidatura em uma eleição. Os dados estão separados por ano de eleição e subdivididos por estado. Os arquivos estão disponibilizados no formato txt, com a seguinte nomenclatura: CONSULTA_CAND_<ANO ELEIÇÃO>_<SIGLA UF>.txt (exemplo: consulta_cand_2012_PR.txt) Os dados mais antigos sobre as candidaturas estão bem incompletos. Por exemplo, em 1994, os dados de 13 estados (CE, DF, ES, MG, MT, PA, PE, PI, PR, RJ, RN e RO) simplesmente não estão disponíveis. A tabela a seguir mostra o total de registros disponíveis sobre as candidaturas por eleição, e novamente pode-se notar que os dados de 1994 e 1996 possuem muito menos entradas que os demais anos.

Tabela 2: Quantidade de registros sobre as candidaturas por ano

ano	Quantidade de registros
1994	5177
1996	9755
1998	15126
2000	383740
2002	19901
2004	401789
2006	20790
2008	383530
2010	22578
2012	451543
total	1713929

Os arquivos que fazem parte do subitem chamado Bens de Candidatos, associam vários tipos de bens a candidatos em uma eleição. É importante ressaltar que neste caso só existem dados de 2006 para frente. Os registros estão disponíveis em formato de arquivo txt, separados por eleição e, dentro de cada ano eleitoral, separados novamente por estado. Sua Nomenclatura possui o seguinte formato: BEM_CANDIDATO_<ANO ELEIÇÃO>_<SIGLA UF>.txt (exemplo: bem_candidato_2012_PR.txt). A tabela a seguir, mostra os totais de entradas de bens de

candidatos disponibilizadas por eleição:

Tabela 3: Quantidade de registros sobre bens de candidatos nas eleições por ano

ano	Quantidade de registros
2006	86946
2008	796114
2010	104597
2012	892979
total	1880636

No subitem Legendas, dentro de Candidatos, temos os dados referentes a relação entre uma candidatura, partidos e coligações. Os dados estão disponíveis por estado, mas nos anos que ocorrem eleições presidenciais, existe um arquivo (BR) que disponibiliza os dados da esfera federal. Nas eleições de 2012, existem dados de 14 estados somente (AC, AL, AM, AP, BA, CE, ES, GO, MA, MS, MT, PA, PB e PE). EM 2010, existe um campo não documentado que possui valores numéricos. Por fim, em 1994 existem dados sobre 14 estados e BR(AC, AL, AM, AP, BA, BR, GO, MA, MS, RR, RS, SC, SE, SP, TO). Os arquivos possuem a seguinte nomenclatura: CONSULTA_LEGENDAS_<ANO ELEIÇÃO>_<SIGLA UF> (exemplo: consulta_legendas_2010_PR.txt). Na tabela a seguir, constam o total de registros disponíveis para cada eleição:

Tabela 4: Quantidade de registros sobre as legendas por ano

ano	Quantidade de registros
1994	1016
1996	1831
1998	2660
2000	17734
2002	4864
2004	17304
2006	4833
2008	97022
2010	6519
2012	102115
total	355898

Em Vagas, outro subitem de Candidatos, assim como Legendas, existem informações sobre os cargos disponíveis na eleição em questão. Os dados estão disponíveis desde 1994, separados por ano da eleição e separados novamente por estado. Os arquivos estão no formato txt, com a seguinte nomenclatura: CONSULTA_VAGAS_<ANO ELEIÇÃO>_<SIGLA UF>. (exemplo: consulta_vagas_2012_PR.txt). No ano de 1994, existem dados de somente 15 estados (AC, AL, AM, AP, BA, GO, MA, MS, RO, RR, RS, SC, SE, SP e TO). Já no ano de 1996, os dados disponíveis são referentes somente as capitais dos estados. Em nenhum ano com eleições presidenciais, constam como vagas o cargo de presidente e vice-presidente. A tabela a seguir, mostra o total de entradas de vagas nos anos das eleições:

Tabela 5: Quantidade de registros sobre vagas em eleições por ano

ano	Quantidade de registros
1994	60
1996	244
1998	108
2000	11118
2002	108
2004	11124
2006	108
2008	11297
2010	117
2012	16704
total	50988

Em Resultados, existem dados sobre os resultados da eleições. Nesta seção, temos detalhes sobre as votações nominais e votações por partido, ambos divididos em zonas eleitorais e em seus respectivos municípios. Temos também resultados de votação por seção eleitoral e detalhes da apuração da eleição, tanto por zona e município quanto por seção eleitoral. Assim, temos como subitens desta área: votação nominal por município e zona, votação partido, por município e zona, votação por seção eleitoral, detalhe de apuração por município e zona e detalhe de apuração por seção eleitoral. Em Detalhe por Votação no Município e Zona, existem dados sobre como se deram, de forma geral, os votos nas zonas eleitorais, tais como: quantidade de votos em branco, de votos nominais, etc. Os dados estão disponíveis em formato txt, divididos por ano de eleição e novamente divididos por estado. Estes arquivos possuem a seguinte nomenclatura: DETALHE_VOTACAO_MUNZONA_<ANO ELEIÇÃO>_<SIGLA UF> (exemplo: detalhe_votacao_munzona_2012_PR.txt). Nesta seção, o ano de 1996 possui poucos dados se comparado com os disponíveis nos outros anos, como pode ser visto na tabela abaixo.

Tabela 6: Quantidade de registros sobre detalhe por votação no município e zona

ano	Quantidade de registros
1994	14353
1996	600
1998	26247
2000	12570
2002	27665
2004	12618
2006	27665
2008	12929
2010	39348
2012	12837
total	186832

Em Votação por Seção, existem dados sobre como se deram os votos nas seções eleitorais, tais como: quantidade de votos em branco, de votos nominais, etc. Os dados estão disponíveis em formato txt, divididos por ano de eleição e novamente divididos por estado. Estes

arquivos possuem a seguinte nomenclatura: DETALHE_VOTACAO_SECAO_<ANO ELEIÇÃO>_<SIGLA UF> (exemplo: detalhe_votacao_secao_2012_PR.txt). Nesta coleção de dados, no ano de 2000, os dados de Tocantins e Exterior não estão disponíveis. Em 2002, os dados sobre a votação no Exterior também não estão disponíveis. Por fim, em 2006, os dados de Santa Catarina não estão disponíveis. Na tabela abaixo, estão os totais dos registros disponíveis por ano:

Tabela 7: Quantidade de registros sobre votação por seção

ano	Quantidade de registros
1994	276382
1996	74653
1998	1864913
2000	616675
2002	2723486
2004	732519
2006	3045510
2008	784471
2010	3254507
2012	815082
total	14188198

Em Votação do Partido por Município e Zona, temos a quantidade de votos dos candidatos dos partidos em uma zona eleitoral. Os dados estão disponíveis em formato txt, separados por ano e por estado. Os arquivos possuem com a seguinte nomenclatura: VOTACAO_PARTIDO_MUNZONA_<ANO_ELEICAO>_<SIGLA_UF>.txt. (exemplo: detalhe _votacao_munzona_2012_PR.txt). Este conjunto de dados possui vários dados faltantes. Em 1994, 1998, 2002, 2006 e 2010 não possuem dados referentes ao âmbito nacional. Em 2002, também não esta disponível os dados sobre a votação no exterior. No ano de 2010, existe uma coluna não documentada com um valor numérico. Em 2012, o campo referente ao nome da coligação, ao invés de seu valor, possui o valor da coluna anterior. Por fim, a ausência mais relevante é que, nos anos anteriores à 2008, não existem dados sobre os votos de legenda. Em 2008, também não é encontrado os dados referentes ao estado de Alagoas. Na tabela abaixo, estão os totais dos registros disponíveis por ano:

Em Votação do candidato por município zona, temos os dados sobre a votação dos candidatos por município e zona. Estes dados estão disponíveis com a seguinte nomenclatura: VOTACAO_CANDIDATO_MUNZONA_<ANO_ELEICAO>_<SIGLA_UF> (exemplo: votacao_candidato_munzona_2012_PR.txt). No ano de 1994, não é possível encontrar os dados do estado de São Paulo, em 1996, não existem os dados referentes à Roraima, e 1998 não possui os dados sobre o Acre. Em 2006, não estão disponíveis os dados sobre as coligações dos candidatos. Na tabela abaixo, estão os totais dos registros disponíveis por ano:

Tabela 8: Quantidade de registros sobre votação de partido por município e zona

ano	Quantidade de registros
1994	144037
1996	5950
1998	337955
2000	78581
2002	397645
2004	91668
2006	398310
2008	94446
2010	379886
2012	113273
total	2041751

Tabela 9: Quantidade de registros sobre votação de candidato por município e zona

ano	Quantidade de registros
1994	399922
1996	73036
1998	1541667
2000	596167
2002	1903010
2004	64348
2006	2074808
2008	646730
2010	2098603
2012	857169
total	10255460

Em Prestação de Contas, estão os dados sobre as receitas e despesas de campanha dos candidatos, partidos e comitês nas eleições. Existem dados disponíveis para as eleições ocorridas em 2006, 2008, 2010 e 2012. As despesas do candidato estão separados por ano. Porém, os campos de cada ano não necessariamente possuem os mesmo nomes. Apesar de serem as mesmas categoria de dados, cada ano possui a sua própria estrutura de organização. Na tabela abaixo, estão os totais de registros disponíveis por ano:

A base de dados eleitorais possui alguns “buracos” em seu conteúdo. Mesmo sendo uma pequena parcela do total, a identificação destas parcelas é importante, pois se o objetivo do processo é a aplicação de técnicas de KDD, a ausência destes dados pode significativa, no processo de interpretação. No próximo tópico, serão apresentados os dados disponíveis de forma técnica. As tabelas originais, com a descrição de dados fornecida pelo próprio sitio, encontram-se nos anexos deste trabalho.

3.2 ORGANIZAÇÃO DO REPOSITÓRIO

De acordo com Pyle apud Goldschmidt e Passos (2005), podemos classificar as variáveis de três formas: nominais ou categóricas, discretas e contínuas. As variáveis nominais são

aquelas usadas para nomear ou atribuir rótulos a objetos não existindo uma relação de ordenação entre seus valores. Por exemplo, “solteiro”, não pode ser considerado menor ou maior que “viúvo”. Variáveis discretas, se assemelham as nominais, porém possuem um ordenamento e este por sua vez, possui algum tipo de significado. Os dias da semana, por exemplo, podem ser considerados discretos. Domingo vem depois de Sábado que vem depois da Sexta-feira. E, as variáveis contínuas, são valores que possuem uma relação de ordem entre si, podendo se finitos ou infinitos. Idade, renda são dois exemplos de variáveis normalmente definidas como contínuas.

Nas tabelas a seguir, foram retirados os atributos redundantes e não relacionados com o processo eleitoral, tais como a data que o arquivos foi “baixado”. As entidades listadas abaixo, representam o formato disponibilizado para o público pelo sítio do TSE, e não o esquema usado pelo tribunal. Além disso, foram listadas apenas as entidades que são "bem comportadas"ao longo dos anos. As que tem seus atributos e nomes "mutantes", não participam desta listagem.

A descrição dos atributos foi feita de forma superficial, sem entrar no domínio do problema. Esta etapa do projeto tem como objetivo fornecer uma base para o entendimento da complexidade dos dados a serem tratados. No apêndice, estão listados todos os possíveis valores que podem ser assumidos pelos atributo nominais de cada entidade, exatamente com constam na base de dados.

3.2.1 BENS DE CANDIDATO

O atributo "Ano Eleição", representa o ano em que ocorreu a eleição cujo o bem foi declarado. A "Descrição da Eleição", define qual o tipo da eleição ocorrida. Em "Sigla da Unidade Federativa", está a sigla da unidade federativa da qual os dados são provenientes. Por fim, em "Descrição do Tipo de Bem do Candidato", encontra-se a descrição de como o TSE classifica o bem declarado pelo candidato. Na tabela 11, encontra-se a classificação de acordo com Pyle apud Goldschmidt e Passos (2005).

Tabela 10: Classificação dos atributos da tabela Bens de Candidato

codigo	Tipo da variável	Quantidade
ANO_ELEICAO	Nominal	4
DESCRICAO_ELEICAO	Nominal	3
SIGLA_UF	Nominal	2
SQ_CANDIDATO	Numérico	-
DS_TIPO_BEM_CANDIDATO	Nominal	51
VALOR_BEM	Numérico	-

3.2.2 CONSULTA CANDIDATURAS

Em candidaturas, estão os dados referentes aos candidatos e candidaturas. O atributo "Ano Eleição", representa o ano em que ocorreu a eleição cujo o bem foi declarado. A "Descrição da Eleição", define qual o tipo da eleição ocorrida. Em "Sigla da Unidade Federativa", está a sigla da unidade federativa da qual os dados são provenientes. O atributo "Sigla ue"é o código da unidade eleitoral (pode ser um estado ou município, dependendo da amplitude da eleição). O campo "descrição ocupação"representa todas as profissões declaradas na eleição pelos candidatos, possuindo 334 valores.

Tabela 11: Classificação dos atributos da tabela Consulta de Candidaturas

codigo	Tipo da variável	Quantidade
ANO_ELEICAO	Nominal	8
NUM_TURNO	Nominal	2
DESCRICAO_ELEICAO	Nominal	8
SIGLA_UF	Nominal	27
SIGLA_UE (*)	Nominal	-
DESCRICAO_CARGO	Nominal	13
NOME_CANDIDATO	Nominal	-
SEQUENCIAL_CANDIDATO (*)	Nominal	-
NUMERO_CANDIDATO	Nominal	-
NOME_URNA_CANDIDATO	Nominal	-
DES_SITUACAO_CANDIDATURA	Nominal	16
SIGLA_PARTIDO	Nominal	39
COMPOSICAO_LEGENDA	Nominal	-
DESCRICAO_OCUPACAO	Nominal	334
DATA_NASCIMENTO	Data	-
NUM_TITULO_ELEITORAL_CANDIDATO	Nominal	-
DESCRICAO_SEXO	Nominal	3
DESCRICAO_GRAU_INSTRUCAO	Discreto	17
DESCRICAO_ESTADO_CIVIL	Nominal	6
DESCRICAO_NACIONALIDADE	Nominal	6
SIGLA_UF_NASCIMENTO	Nominal	28
NOME_MUNICIPIO_NASCIMENTO	Nominal	-
DESPESA_MAX_CAMPANHA	Contínuo	-
DESC_SIT_TOT_TURNO	Nominal	21

3.2.3 CONSULTA LEGENDAS

O atributo "Ano Eleição", representa o ano em que ocorreu a eleição cujo o bem foi declarado. A "Descrição da Eleição", define qual o tipo da eleição ocorrida. "Num Turno"indica se a eleição ocorreu no primeiro ou segundo turno, "Sigla UF"representa a sigla do estado ao qual os dados se referem, "Sigla ue"é o código da unidade eleitoral (pode ser um estado ou município, dependendo da amplitude da eleição), "Nome ue"representa o nome da unidade eleitoral ao qual os dados se refere, "Descricao cargo"refere-se ao cargo para o qual os dados são válidos. Em "Tipo Legenda"é definido se a candidatura ao cargo é de partido isolado ou

coligação. "Sigla Coligação"e "Nome Coligacao", identificam a coligação. Em "Composicao Coligacao"estão os nomes de todos os partidos que compõem a coligação, separados por "/". Na tabela 12, encontra-se a classificação de acordo com om Pyle apud Goldschmidt e Passos (2005).

Tabela 12: Classificação dos atributos da tabela Consulta Legendas, de acordo com Pyle apud Goldschmidt e Passos (2005)

codigo	Tipo da variável	Quantidade
ANO_ELEICAO	Nominal	10
NUM_TURNO	Nominal	2
DESCRICAO_ELEICAO	Nominal	4
SIGLA_UF	Nominal	27
SIGLA_UE (*)	Nominal	51
NOME_UE	Nominal	-
DESCRICAO_CARGO	Nominal	14
TIPO_LEGENDA	Nominal	2
SIGLA_PARTIDO	Nominal	44
SIGLA_COLIGACAO	Nominal	-
NOME_COLIGACAO	Nominal	-
COMPOSICAO_COLIGACAO	Nominal	-

3.2.4 DETALHE VOTAÇÃO MUNICÍPIO ZONA

O atributo "Ano Eleição", representa o ano em que ocorreu a eleição cujo o bem foi declarado. A "Descrição da Eleição", define qual o tipo da eleição ocorrida. "Num Turno"indica se a eleição ocorreu no primeiro ou segundo turno, "Sigla UF"representa a sigla do estado ao qual os dados se referem, "Sigla ue"é o código da unidade eleitoral (pode ser um estado ou município, dependendo da amplitude da eleição), "Nome Município"representa o nome do município ao qual os dados se refere, "Numero Zona"refere-se ao código da zona eleitoral, "Descricao cargo"refere-se ao cargo para o qual os dados são válidos. Todos os dados de quantidade são referentes a votação em questão. Na tabela 13, encontra-se a classificação de acordo com Pyle apud Goldschmidt e Passos (2005).

3.2.5 PERFIL ELEITORADO

Em "período"estão os anos para os quais os dados desta entidade fazem referencia, "Sexo"possui três valores: masculino, feminino e "não definido". O não definido representa uma falta de informação e não uma terceira classificação de gênero. Em "faixa etária"e "grau de escolaridade"os atributos são discretos, funcionando como se fossem bandas que subdividem a classe como um todo.

Tabela 13: Cclassificação dos atributos da tabela Detalhe Votação Município Zona, de acordo com Pyle apud Goldschmidt e Passos (2005)

codigo	Tipo da variável	Quantidade
ANO_ELEICAO	Nominal	10
NUM_TURNO (*)	Nominal	2
DESCRICAO_ELEICAO (*)	Nominal	3
SIGLA_UF	Nominal	30
SIGLA_UE (*)	Nominal	-
NOME_MUNICIPIO	Nominal	-
NUMERO_ZONA (*)	Nominal	-
DESCRICAO_CARGO	Nominal	10
QTD_APTOS	Contínuo	-
QTD_SECOES	Contínuo	-
QTD_SECOES_AGREGADAS	Contínuo	-
QTD_APTOS_TOT	Contínuo	-
QTD_SECOES_TOT	Contínuo	-
QTD_COMPARECIMENTO	Contínuo	-
QTD_ABSTENCOES	Contínuo	-
QTD_VOTOS_NOMINAIS	Contínuo	-
QTD_VOTOS_BRANCOS	Contínuo	-
QTD_VOTOS_NULOS	Contínuo	-
QTD_VOTOS_LEGENDA	Contínuo	-
QTD_VOTOS_ANULADOS_APU_SEP	Contínuo	-

Tabela 14: Classificação dos atributos da tabela Perfil Eleitorado, de acordo com Pyle apud Goldschmidt e Passos (2005)

codigo	Tipo da variável	Quantidade
PERIODO	Nominal	10
UF	Nominal	30
MUNICIPIO	Nominal	-
NR_ZONA	Nominal	-
SEXO	Nominal	3
FAIXA_ETARIA	Discreto	13
GRAU_DE_ESCOLARIDADE	Discreto	16
QTD_ELEITORES_NO_PERFIL	Contínuo	-

3.2.6 VOTAÇÃO NOMINAL POR MUNICÍPIO E ZONA

O atributo "Ano Eleição", representa o ano em que ocorreu a eleição cujo o bem foi declarado. A "Descrição da Eleição", define qual o tipo da eleição ocorrida. "Num Turno"indica se a eleição ocorreu no primeiro ou segundo turno, "Sigla UF"representa a sigla do estado ao qual os dados se referem, "Sigla ue"é o código da unidade eleitoral (pode ser um estado ou município, dependendo da amplitude da eleição), "Nome Município"representa o nome do município ao qual os dados se refere, "Numero Zona"refere-se ao código da zona eleitoral, "Descricao cargo"refere-se ao cargo para o qual os dados são válidos. Todos os dados de quantidade são referentes a votação em questão.

Os atributos "Numero Zona", "Numero Candidato"e "Sq candidato"são nominais e não é apresentada uma quantia, pois são muito numerosos. Já o total de votos é o atributo que

representa a votação em si.

Na tabela 15, encontra-se a classificação de acordo com om Pyle apud Goldschmidt e Passos (2005).

Tabela 15: Classificação dos atributos da tabela Votação Nominal Por Município e Zona, de acordo com Pyle apud Goldschmidt e Passos (2005)

codigo	Tipo da variável	Quantidade
ANO_ELEICAO	Nominal	8
NUM_TURNO (*)	Nominal	2
DESCRICAO_ELEICAO (*)	Nominal	3
SIGLA_UF	Nominal	30
SIGLA_UE (*)	Nominal	-
NOME_MUNICIPIO	Nominal	-
NUMERO_ZONA (*)	Nominal	-
NUMERO_CAND (*)	Nominal	-
SQ_CANDIDATO (*)	Nominal	-
NOME_CANDIDATO	Nominal	-
DESCRICAO_CARGO	Nominal	10
DESC_SIT_CAND_SUPERIOR	Nominal	3
DESC_SIT_CANDIDATO	Nominal	22
DESC_SIT_CAND_TOT	Nominal	15
SIGLA_PARTIDO	Nominal	43
NOME_COLIGACAO	Nominal	-
COMPOSICAO_LEGENDA	Nominal	-
TOTAL_VOTOS	Contínuo	-

3.3 ESCOPO DO TRABALHO

Devido ao grande volume de dados presente no repositório, é necessário definir um recorte para a aplicação das tarefas propostas neste trabalho. Assim, para a seleção, foram levados em conta a compatibilidade do possível recorte com as tarefas definidas no começo do projeto. Este processo também é importante para que o projeto possa ser concluído dentro do período esperado.

Ao se tratar de eleições, a condição mais importante com certeza é o resultado da candidatura. Verificar se existe algum padrão que possa predizer a condição de eleito ou não de um candidato é pode ser definida como o objetivo da tarefa de classificação deste trabalho. Assim, o recorte para esta tarefa, deve ter como princípio a compatíbilidade com tal objetivo.

As tabelas apresentadas na primeira seção deste capítulo permitem inferir que quanto mais antigos os registros menos dados estão disponíveis. Uma das possíveis explicações para este fato está no próprio processo de informatização. É natural que quanto mais antigo, menos dados estejam em formato digital. Esta hipótese é respaldada pelas atualizações no repositório ocorridas durante este trabalho. Os dados mais antigos aos poucos estão sendo digitalizados por

um processo manual (TSE,2013).

Outro fator relevante para efetuar um recorte é a heterogeneidade dos dados disponíveis. A cada ano, principalmente as informações financeiras, possuem uma estrutura diferente. Isto torna a junção entre os anos uma tarefa difícil sem a ajuda de um especialista da área.

Levando em conta estes itens, optou-se por não consolidar os dados financeiros em um grande *datawarehouse*, mas sim, trabalhar com *datamarts* específicos para cada ano.

Outro fator restritivo são os cargos. Como a condição de eleito pode ser definida de alguma maneira como dependente do cargo ao qual o candidato concorre, uma condição inicial da tarefa de classificação será a separação dos candidatos por cargo pretendido. Com esta restrição, junto com a separação por anos, os cargos executivos acabam por apresentar uma quantidade muito menor de elementos comparados com os cargos do poder legislativo (salvo prefeitos, que será comentado mais adiante). Assim, a tarefa de classificação neste trabalho sera restrita ao poder legislativo.

Até o momento presente, as duas últimas eleições foram as do ano de 2010 e do ano de 2012. Nestes dois anos estão presentes as os pleitos de todos os cargos eletivos dos poderes executivo e legislativo das esferas estaduais, municipais e federais, uma vez que as eleições municipais são alternadas com as estaduais e federais a cada dois anos. Assim, como dito anteriormente, estes dois anos serão os base para este trabalho, uma vez que quanto mais atuais, melhor estão organizados e disponíveis os dados.

Destes dois anos, serão escolhidos os cargos de Deputado Federal, Deputado Estadual, Deputado Distrital, Vereador e Prefeito como recorte para a mineração. O cargo de Senador foi excluído da tarefa pelo mesmo motivo que os cargos executivos. Prefeito foi incluído, pois mesmo sendo um cargo executivo, é numeroso o bastante para ser usado como conteúdo para mineração.

Assim, de forma arbitrária, foram separados os cargos de Deputado Federal, Deputado Estadual e Deputado Distrital (equivalente ao cargo de deputado estadual, porém referente ao Distrito Federal) de 2010 para serem usados na tarefa de classificação de dados. Já os cargos de Vereador e Prefeito, serão usados na tarefa de levantamento de regras de associação.

4 MATERIAIS E MÉTODOS

4.1 METODOLOGIA

A metodologia de trabalho foi teve como base os nove passos descritos por Fayyad et al. (1996a) para a execução da Descoberta de Conhecimento em Base de Dados:

Primeiro, entender o domínio do problema: usar o conhecimento à priori sobre a questão. No estudo de caso deste trabalho, o entendimento foi feito analisando superficialmente o significado dos dados disponíveis. Foram usados documentos disponibilizados pelo próprio TSE, e ocasionalmente foram consultadas referencias adicionais sobre direito eleitoral.

Segundo, criar o conjunto de dados para ser usado: selecionar o subconjunto de dados no qual o processo de descoberta será aplicado. A base de dados do TSE encontra-se bem organizada, porém não está perfeita. Existem alguns anos que não possuem todos os dados disponíveis. Levando em conta as características da base disponível foi feito um recorte nos dados para que o processo seja mais acurado (melhor descrito em um capítulo sobre o projeto).

Terceiro, limpeza e pré-processamento de dados: remover os ruídos da base, tratar dados faltantes e questões relacionadas ao gerenciamento do banco de dados. Após remover os conjuntos de dados que podem distorcer a descoberta de conhecimento, foi decidido como tratar os casos irregulares nos dados restantes.

Quarto, projeção e redução de dados: preparar os dados a fim de servir como entrada para a o processamento, visando uma eficiência maior do processo. No caso deste trabalho, esta etapa foi caracterizada, principalmente, pela conversão dos dados para o formato .arff, formato de entrada padrão do WEKA (software para mineração e análise de dados).

Quinto, escolher a função de mineração de dados: escolher a tarefa de mineração a ser realizada: como foi descrito anteriormente, foram utilizadas as tarefas de associação e de classificação.

Sexto, escolher os algoritmos de mineração: escolher os métodos que serão usados na busca por padrões. Nesta etapa foram selecionados cinco algoritmos (*j48, naive-bayes, knn,*

multilayer perceptron e support vector machine) disponíveis na ferramenta de mineração de dados WEKA.

Sétimo, mineração de dados: procura de padrões nos dados. Fase que consiste na aplicação dos algoritmos em si. Nesta fase, foram usados o WEKA e uma aplicação JAVA para obter resultados.

Oitavo, interpretação: interpretar os resultados da mineração e, se necessário, retornar alguns passos e transformar os achados úteis em informação entendível. Esta fase, será uma verificação do processo. Foi feita uma interpretação superficial das informações obtidas, somente para verificar se o "conhecimento" levantado era, no mínimo, plausível.

Nono, usar o conhecimento descoberto: documentar e reportar o conhecimento gerado para os atores interessados. Esta etapa descrita por Fayyad et al. (1996a) não faz parte dos objetivos deste projeto. No máximo, pode ser considerado com o um possível desdobramento, sendo cabível para trabalhos futuros.

4.2 MATERIAIS

Nesta seção, serão apresentados os materiais usados para esta pesquisa. Optou-se por descrever apenas os *softwares* usados, pois sua relevância para o projeto é muito mais significativas do que as contribuições de *hardware*.

4.2.1 WEKA

O projeto WEKA (Waikato Environment for Knowledge Analysis) foi fundado pelo governo da Nova Zelândia em 1993 e, na época, tinha o seguinte objetivo:

> *"The programme aims to build a state-of-the-art facility for developing techniques of machine learning and investigating their application in key areas of the New Zealand economy. Specifically we will create a workbench for machine learning, determine the factors that contribute towards its successful application in the agricultural industries, and develop new methods of machine learning and ways of assessing their effectiveness." (HALL et al., 2009)*

O resultado deste projeto é o pacote Weka, um software com algoritmos de aprendizagem de máquina, pré-processamento e visualização de dados. Sua primeira versão pública foi lançada em 1996. (HALL et al., 2009)

Definições posteriores do Weka, apresentam o objetivo do projeto como: fornecer uma coleção de algoritmos de aprendizagem de máquina e pré-processamento de dados para pesquisadores e interessados (HALL et al., 2009). Pode-se notar que, pelo menos em publicações sobre mineração de dados, o viés industrial e desenvolvimentista nacional neozelandes ficou em segundo plano.

O software disponibiliza uma coleção de algoritmos de aprendizagem de máquina, atualizados e divididos em: algoritimos de regressão, classficação, regras de associação, seleção de atributos e pré-processamento de dados. Além disso, o Weka foi projetado para ser rápido, flexível e dar suporte para um processo experimental de mineração de dados (HALL et al., 2009).

O pacote Weka, atualmente é desenvolvido em Java (em suas primeiras versões foram usados Prolog e C). Além disso, o Weka é um software livre e esta disponibilizado sob a licença publica geral do GNU (acrônimo recursive para Gnu is Not Unix) ou GNU GPL (do inglês, General Public Licence) (WEKA, 2013).

Devido essas características, o Weka foi o programa escolhido para executar os algoritmos de mineração de dados. Outro fator que pesou para esta decisão, é a comunidade que gira em torno do Weka. Além de existir uma boa documentação sobre o funcionamento do software, existem um grande número de publicações científicas que usam o Weka como plataforma de mineração de dados. Por fim, o grupo de aprendizado de máquina da universidade de Waikato, entidade responsável pelo Weka, possui inúmeras publicações sobre o assunto.

4.2.2 KETTLE

Kettle é um acrônimo recursivo para "Kettle E.T.T.L. Enviroment", um projeto criado para auxiliar as tarefas de Extração, Transformação, Transporte e Carregamento de Dados. Um dos componentes que fazem parte deste sistema, é o Spoon.

O Spoon é responsável por fornecer uma interface gráfica para montar os processos responsáveis por executar o ELT. Estes processos são executados por outras ferramentas do Kettle, o Kichen e o Pan. O processo que é normalmente utilizado envolve o agendamento de tarefas para serem executados periodicamente, pois este sistema foi feito para trabalhar com cargas de *data warehouses*.

Para este trabalho, o spoon foi usado em duas etapas: Primeiro, para passar os arquivos originais(CSV) para o esquema "original"da base de dados. E, segundo, passar os dados em "original"para o esquema "OLTP", fazendo as transformações necessárias para que este novo

esquema esteja em uma forma normal.

O Kettle permite trabalhar com *jobs* ou *transformations*. *Transformations* são operações atômicas que processam dados. Os *jobs* são combinações de *transformações* que podem receber dados dinâmicos.

Na figura 4, temos uma transformação usada neste trabalho. Seu objetivo era extrair os dados de uma fonte e reorganizá-los e salvá-los em outro tipo de esquema (Mais detalhes sobre esta operação como um todo é feita na descrição do projeto).

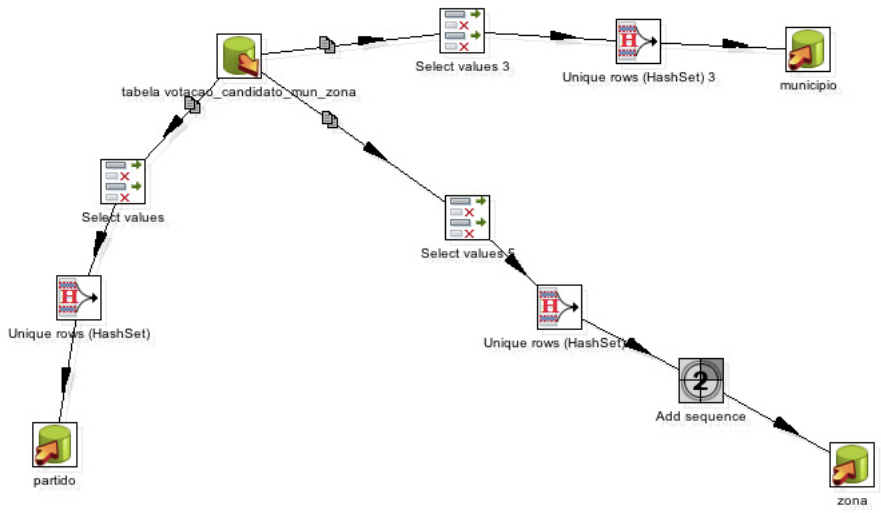

Figura 4: Exemplo de uma transformação usada neste trabalho

4.2.3 POSTGRESQL

O sistema gerenciador de banco de dados escolhido para este projeto foi o Postgresql. Sua escolha deve-se, principalmente, pelos seguintes fatores: Primeiro, por ser um programa de código aberto. Segundo, funciona em diferentes sistemas operacionais. Terceiro, o kettle e o weka fornecem uma forma de integração nativa (ambos são desenvolvidos em java).

Além destes fatores, pode-se também destacar que as especificações técnicas do postgresql atendem com facilidade os requisitos deste projeto. Por exemplo, a base de dados não possui um limite de tamanho, algo que garante a possibilidade de se trabalhar com todos os dados necessários. Uma tabela pode ter no máximo 32TB (o conjunto de *hardwares* usados para este trabalho não chega nem perto deste limite). O número máximo de linhas por tabela é ilimitado. Este é um atributo importante, pois existem tabelas que passam de um milhão de

registros.

A versão usada para o trabalho é a 9.3, sendo uma instancia para Mac Os X e duas para Linux. Outro recurso importante é o pgAdmin3, ferramenta usada para a administração das bases de dados.

4.2.4 JAVA

Como linguagem auxiliar de programação, foi escolhido o Java. Esta escolha deve-se, principalmente, ao fato de que a plataforma Weka é desenvolvida na mesma linguagem. Com isso, pode-se criar uma "camada"com o objetivo de extender as funcionalidades do Weka, adaptando-o para este estudo de caso. Nesta adaptação, não foram efetuadas nenhuma mudança no Weka, apenas foram criados métodos com o objetivo de automatizar a aplicação dos experimentos. Em um tópico mais a frente, serão explicados com mais detalhes o funcionamento desta camada.

5 PROJETO

Neste capítulo será apresentada a estrutura que possibilitou a execução dos experimentos. Em termos gerais, o projeto deste protótipo de sistema é a integração dos materiais apresentados no capitulo anterior.

5.1 ARQUITETURA E FLUXO DE TRABALHO

Como este não é um sistema feito para um uso específico, nenhum esquema clássico de engenharia de software foi usado como referência, uma vez que, seu objetivo é apenas ajudar nas tarefas de mineração definidas. Desta maneira, foram apenas seguidos alguns princípios de engenharia, pois o foco não era ter uma estrutura que fosse projetada para sua manutenção, mas sim, que possibilita-se a automatização para os testes.

Assim, foi definido um fluxo de trabalho que une os matérias através dos arquivos de saída e entrada de cada um deles. Para intuito de organização, cada uma dessas partes foram consideradas como um componente do sistema.

5.2 MODELAGEM DA BASE DE DADOS

Os dados existentes no repositório do TSE, foram divididos em três grupos: dados pessoais, dados financeiros e meta-dados. Em dados pessoais, estão os dados que tem uma relação direta com a pessoa que concorre. Por exemplo, idade, nome, ocupação, estado civil, etc. Em dados financeiros, estão todos os dados que possuem algum tipo de valor associado a ele. E, por fim, os meta-dados são dados derivados dos dados originais, ou seja são dados criados a partir dos dados disponíveis.

A opção por trabalhar cada ano com o seu *datamart* elimina a necessidade de se padronizar todos os dados em um *datawarehouse* de todos os anos. Esta forma de trabalho é interessante, uma vez que os dados financeiros se mostram muito difíceis de serem padronizados. O cálculo do valor relativo dos gastos é um cálculo relativamente complexo (inflação), a

diferença da quantidade de dados disponíveis também sugerem que os entre os anos, a forma como são declarados os dados também foi mudada. Ou seja, além do problema de equivalência de valores, existe a possibilidade de que a granulariedade usada em um ano não seja a mesma do outros, mesmo que os rótulos sejam iguais. Assim, foi escolhido não agrupar estes dados.

Como a quantidade de dados financeiros é bem grande, optou-se por trabalhar com suas consolidações. Então, para cada candidatura, foram somadas todas as receitas, todos os bens declarados e todas as despesas, criando os campos receita_total, despesa_total e bem_total.

Para este trabalho, foram criados dois meta-dados: nasceu no estado e eleito vezes. Nasceu no estado é um atributo boleado que simplesmente diz se o candidato nasceu ou não no estado que ele concorre a um cargo. Já eleito vezes, é um atributo mais interessante. Ele guardar a quantidade de vezes que o candidato foi eleito nos últimos anos. Para o ano de 2010, foram usados os anos de 2002 até 2008. Já para 2012, foram usados os anos de 2002 até 2010.

Para a base de dados OLAP, o resultado da candidatura foi agrupado em dois valores nominais: eleito e não eleito.

Estas foram as operações necessárias para passar os dados da base OLTP para a OLAP. Para cada algoritmo, muitas vezes são necessários tratamentos adicionais (normalização, remoção, etc), porém estes foram efetuados na fase de pré-treinamento.

Para a modelarem do esquema OLAP é necessária a definição de uma tabela fato que será o "centro"do esquema. Como um dos objetivos é investigar a condição de eleito de um dado candidato, assim foi natural a escolha da tabela candidatura como central. Para a sua composição, foi necessário desnormalização dos atributos da base OLTP.

As entidades bem, receita e despesa foram desmoralizadas de acordo com as relações existentes com seus respectivos tipos. Ou seja, um bem possuía um tipo (relação um para muitos do ponto de vista do tipo do bem) e pertencia a um candidato(um para muito do ponto de vista do candidato), agora um candidato possui somente um "bem OLAP". Este "bem", é uma agregação dos valores de todos os tipos de bens que o candidato possui. De um ponto de vista mais formal, pode-se dizer que a tabela tipo de cada entidade foi transformada em atributos da entidade correspondente.

Na figura 6, o modelo físico para a base de dados OLAP. Foi escolhida somente representar este nível, pois algumas destas tabelas possuem uma quantidade de atributos muito grande (despesas possui mais de 30). Um detalhamento maior, pode ser encontrado no apêndice, onde estão os *scripts* SQL.

Uma dos principais atributos de um *data mart* é sua componente temporal. Como

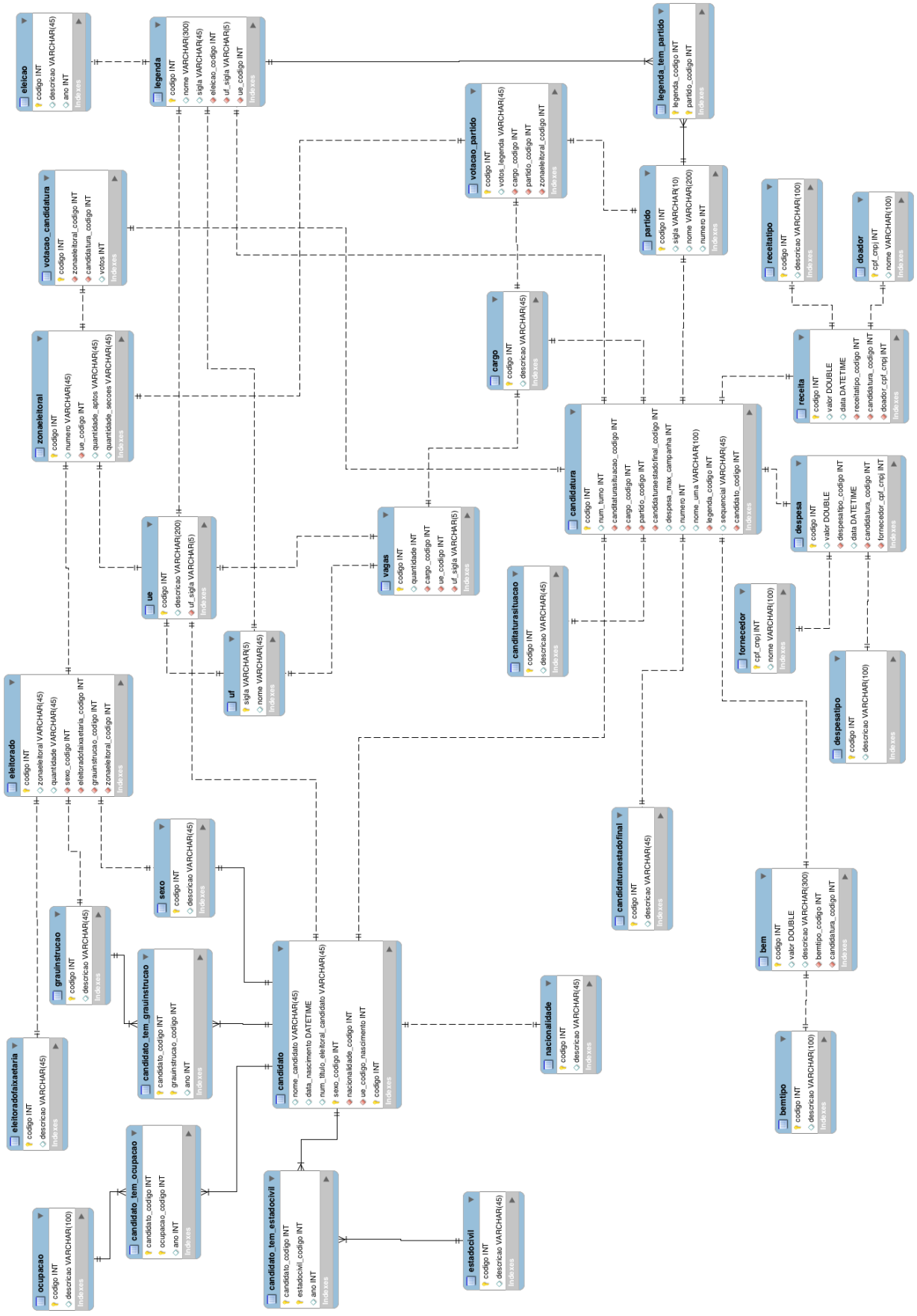

Figura 5: Diagrama entidade relacionamento dos dados de 2012 normalizados

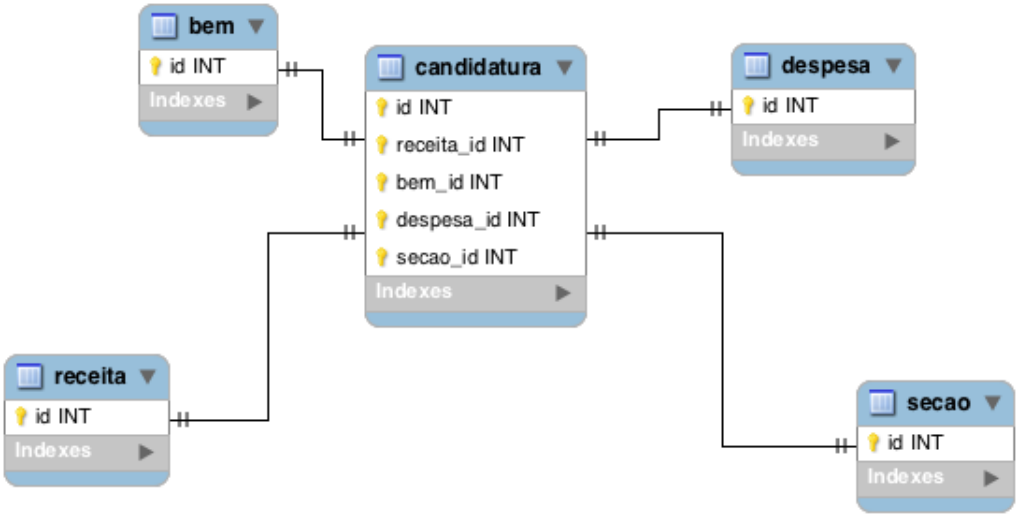

Figura 6: Esquema estrela físico usado para os anos de 2012 e 2010

neste trabalho o *data mart* é usado como repositório de dados para as tarefas de mineração, o ano (única dimensão temporal possível) foi apenas usada como um rótulo, por isso não foi incluída no *data mart*.

5.3 MODELAGEM DA APLICAÇÃO

A camada de aplicação do sistema, foi desenvolvida em Java. Pode-se dizer que esta camada é responsável por usar o Weka como uma biblioteca para executar os experimentos e fazer um controle de entrada e saída de arquivos. Para executar estas duas funções, a camada de aplicação foi dividida em cinco pacotes: ações, es, principal, run e utils.

No pacote ações, estão todas as classes que executam alguma operação dentro do sistema. Este pacote é mais próximo do bloco de *Controller* definido no padrão MVC. Porém, como este *software* não tem nenhum pacote que se assemelhe ao *View* ou *Model* do MVC, optou-se por não usar a mesma nomenclatura. Neste pacote encontram-se as classes de experimento (ExperimentoA e ExperimentoB). O primeiro é referente a aplicação da tarefa de classificação, o segundo é referente a seleção de atributos e o terceiro refere-se à árvore de decisão. O experimento do Apriori não foi feita pela aplicação, uma vez que o seu uso não necessita de nenhuma forma de automação.

O pacote es é responsável por ler e salvar (entrada e saída) arquivos em disco. Ele é responsável por ler os arquivos .arff e salvar os arquivos de saída .xls. Possui duas classes para executar esta tarefa.

O pacote run possui as classes que fazem a comunicação com os classificadores do

Weka. Todas as classes deste pacote são filhas da classe Run, a qual fornece todos os métodos necessários para fazer uma bateria de testes. Cada classe filha representa a aplicação de um método de Classificação. Por exemplo, a classe NaivesBayes aplica o algoritmo do classificado Bayesiano Ingênuo enquanto a classe Knn3 implementa o Três Vizinhos mais Próximos e o Knn5 implementa o Cinco Vizinhos mais Próximos.

O pacote principal apenas existe para abrigar a classe App. Esta classe é a classe inicial da aplicação, responsável por instanciar e chamar qualquer outra classe do projeto.

Em utils, estão todas as classes que devem, de alguma forma trabalhar com as funções do weka que não tem a ver com a classificação. Por exemplo, os filtros de instancias estão nesta classe, o formatado de saída de classificação também.

A figura 7 ilustra os pacotes da aplicação. Não estão representadas neste disgrama as relações inter-pacotes.

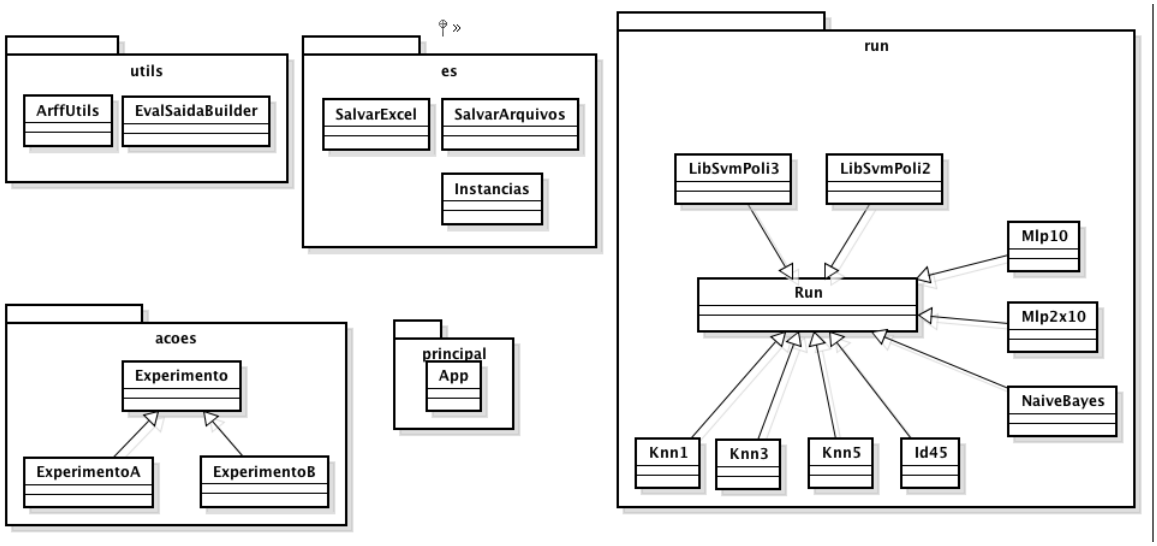

Figura 7: Diagrama de pacotes

5.4 IMPLEMENTAÇÃO

A organização física do sistema gerenciador de banco de dados foi executada de acordo com a seguinte plano: Foram criadas uma instancia de banco de dados para cada ano, nomeadas como “eleição<ANO >” (eleicao_2002, eleicao_2004, etc). Como o postgres permite particionar um banco de dados em esquemas (schemas), dentro de cada banco, foram criados quatro esquemas: oltp, olap, original e temp.

O oltp foi criado para armazenar o esquema físico do modelo entidade-relacionamento definido para o dado ano. Seu nome é proveniente da sigla homônima, usada para abreviar o

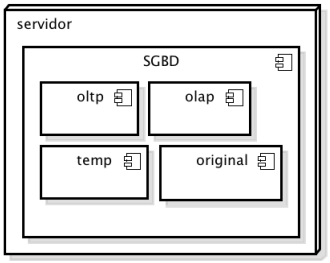

Figura 8: Representação em forma de componentes dos esquemas do SGBD referente a um ano dentro de um servidor físico

Online transactional processing.

No esquema olap, serão armazenados os dados transformados que servirão de base para a tarefa de mineração de dados. Assim, como o oltp, seu nome foi escolhido por ser o esquema que funcionará como um Online analitical processing (olap).

Em originais, foram guardados os dados originais provenientes do repositório do TSE. Este esquema não se encontra em nenhuma forma normal, pois tem como objetivo somente armazenar os dados de uma forma que seja fácil sua manipulação e rastreamento. Desta maneira, pode-se manipular os dados originais através de SQL, facilitando o processo de limpeza, algo que não é possível fazer com os arquivos originais, uma vez que estes estão em formato csv.

Em temp, foram colocados os dados derivados do oltp, que não são diretamente ligados com o olap. Neste esquema, ficam os dados consolidados que podem servir como parte de um processo de transformação para outros bancos de dados.

Para o acesso da aplicação (weka), foram criadas views que retornam os dados da maneira que serão usados no Weka. Com esta estrutura, pode-se dizer que cada ano tem seu data mart próprio.

O processo básico que envolve o sistema gerenciador de banco de dados, pode ser resumido em três passos principais: 1) Transferir os dados originais presentes nos arquivos csv do TRE para o esquema originais do banco de dados. 2) Transferir os dados do esquema originais para o esquema oltp colocando-os no mínimo, na terceira forma normal. 3) Executar o ETL do esquema oltp para o olap. 4) Criar as *views* para servir de entrada para o weka.

As transferencias dos arquivos do TRE para o esquema originais e deste para o oltp, foram feitos através do kettle. Com esta ferramenta foi possível organizar os dados nos esquemas e prepará-los para futuras transformações. Do oltp para o olap, foi usado SQL pois o kettle não tem nenhuma funcionalidade de "pivoteamento"de colunas e linhas, algo que era necessário para este projeto.

O Weka, por sua vez, oferece uma funcionalidade que permite fazer chamadas SQL diretas para um banco de dados através de um módulo JDBC. Para tanto, se faz necessário configurar o weka para que reconheça o conector JDBC reference a base a ser trabalhada. Foi através desta funcionalidade que o Weka, era responsável por fazer as chamadas de sql para carregar as views do esquema olap, já preparadas para o uso no Weka. Desta maneira, o weka fazia chamadas SQL e transformava os dados em arquivos .arff.

A aplicação em java é responsável por ler o arquivo .arff, executar o filtro de dados necessário para a tarefa escolhida, executar todos os algoritmos definidos (knn1, knn3, knn5, svmp2, svmp3, svmr, j48, mlp10 e mpl2x10) a quantidade de vezes necessária para gerar estatísticas. A saída da aplicação é um arquivo .xls com as métricas para a avaliação do sistema.

Por fim, o "sistema"foi implementado em duas máquinas sendo uma contento um banco de dados e as aplicações que se comportam como clientes, enquanto o outro possui apenas um banco de dados. Este formato foi escolhido pela limitação de *hardware* do projeto. Todos os clientes estão na mesma máquina, pois existe uma troca de arquivos grande entre eles.

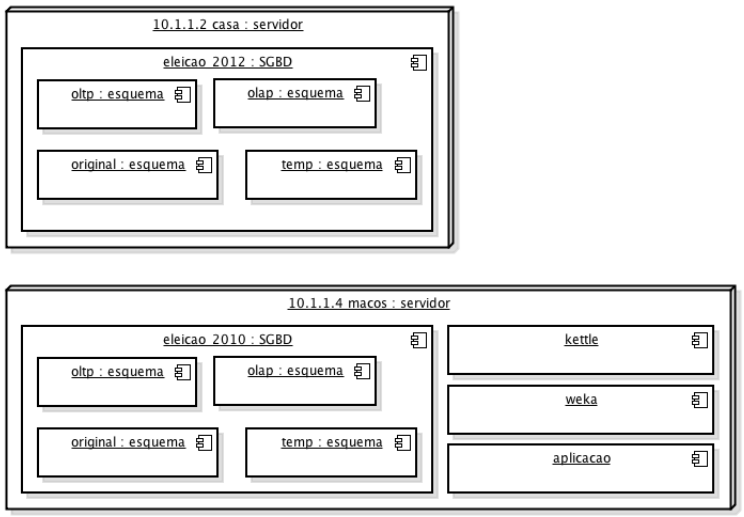

Figura 9: Diagrama de implantação do sistema

6 EXPERIMENTOS

Neste capítulo serão descritos a forma como foram organizados e executados os experimentos deste trabalho. Inicialmente, foram definidas duas abordagens baseadas nas tarefas de classificação e associação. Para classificação, além da aplicação dos algoritmos, foi feita também um estudo de importância de um atributo para o resultado do classificador.

6.1 TAREFA DE CLASSIFICAÇÃO

As tarefa de classificação, basicamente consistem em definir o cargo e ano que será analisado, preparar os dados para a tarefa e executar sistematicamente uma bateria de testes. A definição dos cargos foi feita durante da delimitação do escopo do estudo de caso (candidatos à deputados federais e estaduais(incluindo distritais) em 2010). A preparação dos dados ocorreu no processo de extração, transformação e carga dos arquivos disponibilizados pelo TSE para o conjunto de bases preparada para este estudo. A execução dos testes, é a etapa que sera descrita nesta seção do trabalho.

Levando em conta as diretrizes para a tarefa de classificação, foram preparadas dois conjuntos de testes para os dois cargos escolhidos. No primeiro conjunto, as instâncias usadas para o experimento eram constituídas somente dos dados pessoais disponibilizados pelo TSE. No segundo conjunto, as instâncias, além dos dados pessoais, possuem também a consolidação dos dados financeiros das campanhas dos candidatos mais os atributos criados neste estudo. A seguir, uma tabela que mostras quais são os atributos das instâncias.

Cada um dos conjuntos de instâncias dos dois cargos foi usado de entrada para dez testes de classificação. Os algoritmos usados são: Bayes ingênuo (naive-bayes), Perceptron de Múltiplas camadas com uma camada escondida de dez nós e com duas camadas escondidas de dez nós cada, Máquina de Suporte Vetorial com núcleo polinomial de grau dois, com núcleo polinomial de grau três e com núcleo radial, K-nn com valor de k = 1, k = 3 e k = 5 e o C4.5.

Como a quantidade de instâncias rotuladas como não eleitos é muito mais numerosa

que as de eleitos, para cada um dos testes de validação cruzada, foi feito um balanceamento. Basicamente, o balanceamento é uma forma de aproximar a quantidade de instancias das classes envolvidas na tarefa de mineração. Para cada cargo foram feitos tratamentos diferentes devido a quantidade de instancias também ser diferente. Assim, cada tratamento será explicado juntamente com a apresentação dos resultados de cada cargo.

Depois de selecionadas as instâncias, para cada um dos algoritmos, foram feitas dez rodadas de treino usando como método de teste a validação cruzada com cinco *folds*. De cada um dos *folds*, foram calculados os valores de cobertura, precisão de cada classe e kappa. Precisão, é a proporção entre as instâncias classificadas corretamente como uma dada classe x entre todas as classificadas como pertencentes a esta classe x. Cobertura, é a proporção entre as instâncias classificadas como uma dada classe x entre todas as corretamente pertencentes a esta classe x. Kappa, é uma medida normalizada (assume valores de zero a um) que mede o desempenho do classificador com relação a um classificador aleatório com a mesma proporção entre as classificações Weka (2013).

A seguir, os quadros com os valores de cobertura, precisão de cada classe e a medida kappa das instâncias de cada um dos cargos escolhidos para os testes. Os valores destas tabelas são as médias com seus respectivos desvios padrão segmentados por cada algoritmo.

6.1.1 BASE COM DADOS DOS CANDIDATOS À DEPUTADO FEDERAL EM 2010

O balanceamento das classes das instâncias dos candidatos à deputados federais, ocorreu da seguinte maneira: Primeiro, são mantidos todas as instâncias de não eleitos. Depois são sorteados aleatoriamente as instancias de não eleitos, porém mantendo uma proporção de 2:1 entre as duas classes. Na base de dados, existem 6129 candidatos ao cargo de deputado federal. Destes, 576 foram eleitos e 5553 não foram eleitos. Assim, no pré-processamento dos dados, foram mantidos as 576 instancias eleitas e foram sorteadas aleatoriamente, a cada fold, 864 instancias de não eleitos. A tabela 16 mostra os resultados dos testes dos grupos de instancias que possuem somente dados pessoais e a tabela 17, mostra o grupo com dados pessoais, mais totais financeiros e meta-dados.

No grupo que possui somente os dados pessoais (tabela 16), o melhor valor para para a precisão da classe eleito é o da máquina de suporte vetorial com núcleo radial (svm radial). Para a cobertura da mesma classe e a precisão dos não eleitos, o Bayes ingênuo (Naive-Bayes) tem o melhor valor. E a cobertura dos não eleitos, quem possui o maior valor é novamente o svm radial. Neste caso, houve uma polarização entre o Naive-Bayes e o SVM radial. Ambos possuem dois dos melhores valores de cada classe, porém, o SVM radial possui alguns dos

Tabela 16: Resultado da aplicação dos algoritmos de mineração no primeiro grupo de candidatos à deputado federal em 2010

	eleito		**não eleito**		**geral**
algoritmo	precisão	cobertura	precisão	cobertura	kappa
j48	0,81 (0,03)	0,65 (0,04)	0,79 (0,02)	0,89 (0,02)	0,56 (0,04)
knn-1	0,61 (0,02)	0,75 (0,04)	0,80 (0,03)	0,68 (0,03)	0,42 (0,05)
knn-3	0,60 (0,03)	0,74 (0,03)	0,79 (0,02)	0,67 (0,03)	0,39 (0,04)
knn-5	0,63 (0,02)	0,76 (0,04)	0,82 (0,03)	0,70 (0,03)	0,44 (0,04)
mlp 10	0,78 (0,03)	0,73 (0,05)	0,83 (0,02)	0,86 (0,02)	0,60 (0,05)
mlp 2x10	0,79 (0,09)	0,73 (0,12)	0,83 (0,05)	0,86 (0,10)	0,59 (0,08)
naive bayes	0,76 (0,04)	**0,79** (0,04)	**0,86** (0,02)	0,84 (0,03)	**0,63** (0,05)
svm grau 2	0,67 (0,04)	0,77 (0,04)	0,83 (0,02)	0,74 (0,05)	0,50 (0,05)
svm grau 3	0,47 (0,11)	0,54 (0,36)	0,73 (0,12)	0,64 (0,22)	0,17 (0,17)
svm radial	**0,83** (0,04)	0,52 (0,04)	0,75 (0,02)	**0,93** (0,02)	0,48 (0,05)

Tabela 17: Resultado da aplicação dos algoritmos de mineração no segundo grupo de candidatos à deputado federal em 2010

	eleito		**não eleito**		**geral**
algoritmo	precisão	cobertura	precisão	cobertura	kappa
j48	0,87 (0,02)	**0,99** (0,02)	**0,99** (0,01)	0,90 (0,02)	**0,86** (0,02)
knn-1	0,74 (0,03)	0,77 (0,04)	0,84 (0,02)	0,82 (0,03)	0,59 (0,05)
knn-3	0,75 (0,03)	0,78 (0,03)	0,85 (0,02)	0,82 (0,03)	0,60 (0,03)
knn-5	0,73 (0,03)	0,78 (0,03)	0,84 (0,02)	0,81 (0,03)	0,58 (0,04)
mlp 10	0,90 (0,02)	0,88 (0,03)	0,92 (0,02)	0,94 (0,02)	0,82 (0,03)
mlp 2x10	0,87 (0,05)	0,86 (0,05)	0,91 (0,03)	0,91 (0,05)	0,77 (0,04)
naive-bayes	**0,92** (0,03)	0,85 (0,04)	0,91 (0,02)	**0,95** (0,02)	0,81 (0,04)
svm grau 2	0,90 (0,02)	0,89 (0,03)	0,93 (0,02)	0,93 (0,02)	0,83 (0,03)
svm grau 3	0,79 (0,05)	0,84 (0,05)	0,89 (0,03)	0,84 (0,05)	0,67 (0,05)
svm radial	0,79 (0,04)	0,59 (0,04)	0,77 (0,02)	0,89 (0,02)	0,51 (0,04)

piores valores nos itens que ele não obteve o melhor valor. Assim, é natural que o seu valor do coeficiente kappa seja menor do que o do Naive-Bayes.

No segundo grupo (tabela 17), o desempenho do Naive-Bayes continua bom (possui o melhor valor para a precisão da classe eleita e o melhor valor para cobertura da classe de não eleitos), porém não melhor que o j48 (implementação em java do C4.5) que possui o melhor valor da cobertura dos eleitos e precisão dos não eleitos, além do valor mais alto do coeficiente kappa.

Algo que chama a atenção comparando os resultados dos dois grupos (tabela 16 e tabela 17), é a melhoria generalizada nos índices no segundo grupo. Aparantemente, a adição dos novos atributos se mostrou interessante para uma melhoria da tarefa de classificação (as possíveis causas são comentadas mais a frente no texto). Outro fator interessante, é o alto desempenho nos dois grupos do naive-bayes.

6.1.2 BASE COM DADOS DOS CANDIDATOS À DEPUTADO ESTADUAL OU DISTRITAL EM 2010

O balanceamento das classes das instancias dos candidatos à deputados estaduais e distritais, ocorreu da seguinte maneira: Primeiro, defini-se que serão usados 10% das instancias originais. Como temos 15525 instancias, serão sorteadas aleatoriamente 1545 instancias. Porém, este sorteio deve obedecer dois critérios: Primeiro, as instancias sorteadas não devem ser repetidas. Segundo, entre as 1545 instancias a proporção de classes eleitos para não eleitos, deve ser de 2:1. A tabela 18 mostra os resultados dos testes dos grupos de instancias que possuem somente dados pessoais e a tabela 19, mostra o grupo com dados pessoais, mais totais financeiros e meta-dados.

Tabela 18: Resultado do primeiro grupo aplicado à Deputados Estaduais

	eleito		**não eleito**		**geral**
algoritmo	precisão	cobertura	precisão	cobertura	kappa
j48	**0,78** (0,03)	0,55 (0,05)	0,75 (0,02)	**0,90** (0,02)	0,47 (0,04)
k-nn1	0,61 (0,02)	0,72 (0,03)	0,79 (0,02)	0,70 (0,02)	0,40 (0,03)
k-nn3	0,59 (0,02)	0,71 (0,02)	0,78 (0,02)	0,68 (0,02)	0,37 (0,03)
k-nn5	0,60 (0,02)	0,72 (0,03)	0,79 (0,01)	0,68 (0,02)	0,39 (0,03)
mlp 10	0,75 (0,02)	0,70 (0,03)	0,81 (0,02)	0,84 (0,02)	**0,54** (0,03)
mlp 2x10	0,76 (0,10)	0,65 (0,14)	0,79 (0,05)	0,83 (0,12)	0,49 (0,06)
naive-bayes	0,73 (0,02)	**0,72** (0,04)	**0,82** (0,02)	0,82 (0,02)	0,54 (0,04)
svm grau 2	0,57 (0,07)	0,54 (0,18)	0,71 (0,05)	0,71 (0,18)	0,25 (0,10)
svm grau 3	0,39 (0,09)	0,53 (0,36)	0,68 (0,09)	0,52 (0,26)	0,04 (0,11)
svm radial	0,76 (0,03)	0,56 (0,03)	0,75 (0,01)	0,88 (0,02)	0,46 (0,03)

Tabela 19: Resultado do segundo grupo aplicado à Deputados Estaduais

	eleito		**não eleito**		**geral**
algoritmo	precisão	cobertura	precisão	cobertura	kappa
j48	0,85(0,03)	**0,94**(0,03)	**0,96**(0,02)	0,90(0,03)	**0,82**(0,03)
k-nn1	0,72(0,03)	0,73(0,04)	0,83(0,02)	0,82(0,02)	0,55(0,04)
k-nn3	0,72(0,03)	0,74(0,04)	0,84(0,02)	0,82(0,03)	0,56(0,05)
k-nn5	0,74(0,03)	0,75(0,04)	0,83(0,02)	0,83(0,03)	0,57(0,04)
mpl 10	0,85(0,03)	0,84(0,03)	0,90(0,02)	0,90(0,02)	0,75(0,03)
mlp 2x10	0,84(0,06)	0,79(0,11)	0,88(0,05)	0,91(0,05)	0,70(0,07)
naive-bayes	**0,89**(0,02)	0,82(0,03)	0,89(0,02)	**0,94**(0,01)	0,77(0,03)
svm grau 2	0,87(0,03)	0,83(0,03)	0,90(0,02)	0,92(0,02)	0,76(0,03)
svm grau 3	0,60(0,12)	0,74(0,22)	0,82(0,09)	0,62(0,25)	0,35(0,17)
svm radial	0,76(0,04)	0,53(0,05)	0,74(0,02)	0,88(0,02)	0,44(0,05)

No grupo que possui somente os dados pessoais (tabela 18), o melhor valor para para a precisão da classe eleito é o C4.5 (J48). Para a cobertura da mesma classe e a precisão dos não eleitos, o Bayes ingênuo (Naive-Bayes) tem o melhor valor. E para o cobertura dos não eleitos, quem possui o maior valor é novamente o J48. Porém, mesmo que os dois algoritmos dominem os máximos dos índices usados, é interessante notar que o o Perceptron de Múltiplas Camadas com uma camada de dez neurônios possui uma valor tão alto quanto o do Bayes Ingênuo para

o coeficiente Kappa.

No segundo grupo (tabela 19), o desempenho do Naive-Bayes continua bom (possui o melhor valor para a precisão da classe eleita e o melhor valor para cobertura da classe de não eleitos), porém não melhor que o j48 (implementação em java do C4.5) que possui o melhor valor da cobertura dos eleitos e precisão dos não eleitos, além do valor mais alto do coeficiente kappa. É interessante notar que este resultado é muito similar ao encontrado para os candidatos a deputados federais de 2010.

Novamente, comparando os resultados dos dois grupos (tabela 18 e tabela 19), percebe-se a melhoria generalizada nos índices no segundo grupo. Mais uma vez, a adição dos novos atributos se mostrou interessante para uma melhoria da tarefa de classificação e, mais uma vez, deve-se destacar o desempenho do Naive-Bayes.

6.2 SELEÇÃO DE ATRIBUTOS

A seleção de atributos é um teste que visa verificar a relativa importância de um atributo para com o resultado da classificação. Para isso, deve-se ordenar os atributos das instâncias de acordo com alguma medida. A escolha desta medida, assim como o seu método de busca são importantes, pois o primeiro define o viés de avaliação, enquanto o segundo afeta o tempo computacional.

Neste trabalho, foi escolhido como o ganho de informação como avaliador de atributos. O ganho de informação é a mesma medida descrita no C4.5, onde era usada para bifurcar a árvore de decisão. Para a aplicação da seleção de atributos, foram escolhidos somente os conjuntos de instancias mais completos de cada um dos cargos. Neste conjunto, as instancias foram processadas da seguinte maneira: Primeiro, aplica-se a seleção de atributos para ordenar os atributos de forma decrescente pelo seu valor de ganho de informação. Depois disto, são escolhidos os N (onde N é um número arbitrário maior que zero e menor que o número de atributos em uma instância) primeiros atributos e todo o resto é descartado. Estes novos conjuntos de instancias, cada um com N atributos, foram usado de entrada para o algoritmo com melhor desempenho no teste de classificação. Este processo é repetido aumentando o valor de N, até que se chegue ao máximo possível. Com isso, pode-se usar a média do coeficiente kappa de cada rodada destas para criar uma curva. Esta curva deve descrever o comportamento da quantidade de atributos com relação aos resultados obtidos. Como os conjuntos originais escolhidos possuem 12 atributos, foi definido que 3 seria a quantidade de atributos adicionados a cada novo teste. Assim, para cada conjunto, foram rodados 10 testes usando a validação cruzada com 5

folds. A seguir, a curva de cada um dos testes, junto com uma análise mais específica para cada um dos casos.

6.2.1 BASE COM DADOS DOS CANDIDATOS À DEPUTADO FEDERAL EM 2010

No caso dos deputados federais de 2010, o algoritmo usado foi o C4.5 devido ao sei valor kappa nos testes de classificação. É interessante notar o ganho de desempenho que ocorrem ao passar de três para seis atributos. Em um estudo rápido, somente com o intuito de verificar se a hipótese de que o atributo eleito é o mais significativo dos três, foram rodados três teste com validação cruzada usando 10 folds, cada um com quatro atributos. Cada teste possuía instancias com os três atributos iniciais adicionados de um dos atributos extras do grupo de seis, um de cada vez. Os valores do kappa fora os seguintes: quando adicionado o atributo vezes eleito 0,87. Quando foi adicionado o atributo bem total, o valor foi de 0,67. Para partido sigla, o kappa ficou em 0,65. É interessante notar que esta ordenação dos atributos, colocando os financeiros e o eleito vezes entre os quatro primeiros corrobora com o ganho de eficiência apresentado quando estes atributos foram adicionados na classificação. A seguir, a tabela 20 mostrando os atributos que fazem parte dos grupos de instancias com tamanho N e a curva do valor médio de kappa e quantidade de atributos ordenados, para as instancias de candidatos à deputados federais em 2010.

Tabela 20: Atributos escolhidos na seleção de acordo com a quantidade escolhida

quantidade	atributos escolhidos
3	receita total, despesa total, ocupacao e resultado
6	receita total, despesa total, ocupacao, eleito vezes, bem total, partido sigla e resultado
9	receita total, despesa total, ocupacao, eleito vezes, bem total, partido sigla, grauinstrucao, estadocivil, ue e resultado
12	todos os atributos

6.2.2 BASE COM DADOS DOS CANDIDATOS À DEPUTADO ESTADUAL OU DISTRITAL EM 2010

Para os candidatos ao cargo de deputado estadual ou distrital em 2010, também foi usado o algoritmo C4.5, pois o valor do seu coeficiente kappa foi o mais alto na tarefa de classificação. As escolhas dos atributos foram semelhantes aos feitos para os deputados federais. O mesmo estudo efetuado para verificar a importância do atributo eleito vezes foi feito, porém, na primeira rodada já foi identificado que a árvore montada pelo algoritmo, neste caso usa somente a receita total para classificar as instancias. A seguir, a tabela 21 mostrando os atributos que fazem parte dos grupos de instancias com tamanho N e a curva do valor médio de kappa

e quantidade de atributos ordenados, para as instancias de candidatos à deputados estaduais ou distritais em 2010.

Tabela 21: Atributos escolhidos na seleção de acordo com a quantidade escolhida

quantidade	atributos escolhidos
3	receita total, despesa total, ocupacao e resultado
6	receita total, despesa total, ocupacao, eleito vezes, bem total, partido sigla e resultado
9	receita total, despesa total, ocupacao, eleito vezes, bem total, partido sigla, grauinstrucao, estadocivil, ue e resultado
12	todos os atributos

6.3 TAREFA DE ASSOCIAÇÃO

Para a tarefa de associação, foi usado somente o Weka, sem o complemento em java. Para ambos os cargos usados (candidatos à vereadores e prefeitos em 2012) foram criados,no mínimo, dois conjuntos de dados: um com alguns atributos pessoais de todos os candidatos àquele cargo e outro com os mesmo atributos, porém somente dos eleitos. Em cada conjunto, foram executadas duas vezes o Apriori com parâmetros diferentes: primeiro, com um suporte baixo e uma confiança alta, buscando regras que sejam específicas e válidas. A segunda combinação de parâmetros foi invertida: suporte relativamente alto com uma confiança mais baixa, buscando regras mais abrangentes.

Para todos os testes, foi definido que a quantidade máxima de regras retornadas pelo Weka seria de no máximo 100. Para os eleitos, foi criado um subgrupo somente com as mulheres eleitas. Este recorte não foi feito para o conjunto com todos as instancias devido a grande quantidade de dados. Por fim, os números entre parênteses são o índice que o Weka retornou e a lista completa de regras encontra-se nos apêndices.

6.3.1 BASE COM DADOS DOS CANDIDATOS À VEREADOR EM 2012

Para o teste com suporte baixo e confiança alta usando os candidatos à vereadores em 2012 (424296 instâncias), foram definidos um suporte mínimo de 0,01 e uma confiança mínima de 0,90. Das regras retornadas, podemos destacar três regras:

- se for do sexo feminino, não eleito (29)
- se for do sexo feminino, grau de instrução médio completo e nunca foi eleita antes, não eleita. (11)

- se o estado for São Paulo, e nunca foi eleito antes, não eleito (28)

Já para o teste com suporte alto e confiança baixa usando os candidatos à vereadores em 2012 (424296 instâncias), foram definidos um suporte mínimo de 0,30 e uma confiança mínima de 0,70. Das regras retornadas, podemos destacar três regras:

- se não eleito, nunca foi eleito. (3)
- se grau de instrução médio completo, nunca eleito. (14)
- se estado civil for casado e não foi eleito, homem. (28)

Para o teste com suporte baixo e confiança alta usando somente os vereadores eleitos em 2012 (59610 instâncias), foram definidos um suporte mínimo de 0,01 e uma confiança mínima de 0,90. Abaixo, todas a regras retornadas:

- se grau de instrução for fundamental incompleto e estado civil for casado, homem. (1)
- se grau de instrução for fundamental incompleto, homem. (2)
- se ocupação for agricultor, homem. (3)
- se grau de instrução for fundamental completo, homem. (4)
- se nunca foi eleito, homem. (5)
- se é vereador e tem entre 28,5 e 46,5 anos, homem(6).

Já para o teste com suporte alto e confiança baixa usando somente os vereadores eleitos em 2012 (59610 instâncias), foram definidos um suporte mínimo de 0,30 e uma confiança mínima de 0,70. Das regras retornadas, podemos destacar três regras:

- se nunca foi eleito, é homem. (1)
- se é casado, é homem. (2)
- se tem ensino médio completo, é homem. (3)
- se tem entre 28,5 e 46,5 anos e é casado, é homem (4)
- se tem entre 28,5 e 46,5 anos, é homem (5)

- se nunca foi eleito antes, é homem (6)

Por fim, o último teste com vereadores, quando usamos somente as mulheres eleitas vereador em 2012 (7921), foram definidos um suporte mínimo de 0,30 e uma confiança mínima de 0,70. As regras retornadas encontram-se a seguir:

- se tem entre 28,5 e 46,5 anos, nunca foi eleita. (1)
- se tem entre 28,5 e 46,5 anos, casada. (2)
- se nunca foi eleita, é casada. (3)
- se nunca foi eleita, tem entre 28,5 e 46,5 anos. (4)
- se casada, nunca foi eleita. (5)
- se casada, tem entre 28,5 e 46,5 anos. (6)

Para o teste com suporte baixo e confiança alta usando os vereadores eleitos em 2012 (59610 instâncias), foram definidos um suporte mínimo de 0,01 e uma confiança mínima de 0,90. Das regras retornadas, podemos destacar quatro regras:

6.3.2 BASE COM DADOS DOS CANDIDATOS À PREFEITO EM 2012

Para o teste com suporte baixo e confiança alta usando os candidatos à prefeito em 2012 (16010 instâncias), foram definidos um suporte mínimo de 0,01 e uma confiança mínima de 0,90. Das regras retornadas, podemos destacar cinco regras:

- se é empresário, é homem. (1)
- se estado civil é casado e foi eleito uma vez nos últimos dez anos, eleito. (2)
- se possui ensino médio completo e estado civil é casado, eleito (3)
- se tem 53,4 a 62 e estado civil é casado, homem. (4)
- se foi eleito uma vez nos últimos dez anos e foi eleito, homem. (5)

Já para o teste com suporte alto e confiança baixa usando todos os candidatos a prefeito em 2012 (16010 instâncias), foram definidos um suporte mínimo de 0,30 e uma confiança mínima de 0,70. Das regras retornadas, podemos destacar duas regras:

- se foi eleito uma vez nos últimos 10 anos, é homem. (1)
- se foi eleito, é homem. (2)

Para o teste com suporte baixo e confiança alta usando os prefeitos eleitos em 2012 (5771 instâncias), foram definidos um suporte mínimo de 0,01 e uma confiança mínima de 0,90. Das regras retornadas, podemos destacar três regras:

- se estado for Rio Grade do Sul e estado civil é casado, homem. (7)
- se ocupação for empresário e estado civil é casado, homem. (11)
- se tiver sido eleito duas vez nos últimos 10 anos, homem. (30)

Já para o teste com suporte alto, e confiança mais relaxada, foi usado como valores: 0,3 para suporte e 0,7 para confiança. Assim, podemos destacar três regras levantadas:

- se estado civil é casado e tiver sido eleito uma vez nos últimos 10 anos, homem. (1)
- se tiver sido eleito uma vez nos últimos 10 anos, homem. (2)
- se sexo for masculino e tiver sido eleito uma vez nos últimos 10 anos, estado civil é de casado. (8)

Já para o conjunto com somente com mulheres eleitas (685 instancias), foram usados 0,1 para suporte e 0,7 para confiança. Neste teste foram retornados 13 regras e podemos destacar as seguintes regras:

- se a idade for entre 39,4 e 46,2 anos e não tiver sido eleita nos 10 anos anteriores, o estado civil é de casada. (3)
- se a ocupação for a de prefeito e tiver sido eleita uma vez nos últimos 10 anos, o estado civil é de casada. (7)

6.4 ARVORE DE DECISÃO

A árvore de decisão não é um dos objetivos deste trabalho. Porém, como sua composição é necessária para o algoritmo C4.5 e seu resultado é interessante, foi resolvido mostrar a sua composição. Para deputados federais, a arvore tem somente uma divisão, baseada na receita total do candidato. Abaixo, a arvore para os candidatos ao cargo de deputados federais em 2010:

- receita total $<=$ 61025.33: não eleito (1307.0/32.0)
- receita total $>$ 61025.33: eleito (1348.0/200.0)

7 CONCLUSÃO

O objetivo deste trabalho era usar técnicas de Mineração de dados para procurar padrões no repositório de dados eleitorais do TSE. Com a aplicação das tarefas definidas, foi possível verificar algumas informações interessantes, que não necessariamente constavam claramente na massa de dados. Assim, neste capitulo, serão discutidos os resultados alcançados, tanto de um ponto de vista técnico quanto do ponto de vista do domínio do problema. Espera-se com isso mostrar que os objetivos iniciais foram atingidos e também discutir as limitações e possíveis desdobramentos deste trabalho.

7.1 DISCUSSÃO

Ao longo deste trabalho, o repositório foi atualizado algumas vezes com novos dados. Isto, junto com a evolução natural da qualidade que os dados são disponibilizados, leva a crer que, em um futuro próximo, o tipo de trabalho desenvolvido neste projeto possa ser reaplicado com mais facilidade. Isto é relevante, pois a etapa de organização dos dados foi uma das etapas que mais consumiram tempo neste trabalho.

Uma da premissas do projeto era mostrar que a mineração de dados pode ser usada como ferramenta para um entendimento dos dados. Neste trabalho foram percebidas as limitações desta abordagem, assim como seus pontos fortes.

A tarefa de classificação teve resultados interessantes. Como quase todos os valores giram em torno de 85% de precisão, pode-se considerar que o modelo proposto para a tarefa de mineração se mostrou eficiente para o processo.

Algo que merece ser citado e é passível de uma investigação, é o desempenho do SVM. Ao utilizar a implementação disponibilidade pelo Weka, várias vezes o aviso de que o limite de iteração foi atingindo foi mostrado. Assim, o desempenho dos SVM pode ter sido limitado por este comportamento do algoritmo (1000000 de iterações).

Um conclusão que não faz parte dos objetivos iniciais é sobre a arquitetura da apli-

cação. Uma das possíveis considerações sobre esta parte do projeto, seria quanto a utilidade do java como linguagem de programação. Como dois dos programas principais eram nesta linguagem, a integração através do java se mostrou util para tarefas de mineração de dados.

O desempenho do Apriori, dependendo do ponto de vista, pode ser considerado satisfatório. Muitas das regras encontradas refletem a assimetria da participação de mulheres na política brasileira, em todas as esferas de poder. Por exemplo, a regra "se for do sexo feminino, não eleito"encontrada no teste com todos os candidatos ao cargo de vereadores em 2012 é um bom exemplo disso. Outro resultado interessante é quanto ao estado do Rio Grande do Sul, onde a regra “e estado for Rio Grade do Sul e estado civil é casado, homem.” também colabora para exemplificar a questão. A questão da tradição também existe, porém a tarefa de classificação fornece indícios melhores quanto a sua importância. O modo de funcionamento do Apriori, talvez não seja o melhor para a forma em que os dados eleitorais são organizados. Dependendo da configuração, o algoritmo retornou regras válidas porém "naturais". Por exemplo, o Apriori consegui dizer que advogados tem ensino superior. Iso é interessante, porém para este estudo não tem muita relevância. E o fato do Apriori não tratar bem este tipo de acontecimento, é relevante.

A árvore de decisão serve como uma forma de estudo sobre a questão eleitoral. O fato de delta ter apenas um nível e este nível ser a receita total dos candidatos, serve como indicio da força do poder financeiro dentro das eleições. Resultado este que corrobora com boa parte dos outros experimento deste trabalho.

Descobriu-se que a tradição é uma característica importante para a tarefa de classificação. Ela se encontra parcialmente representada pela quantidade de vezes que o candidato foi eleito em eleições anteriores. Este atributo se mostrou importante para a tarefa de classificação. Outro atributo que poderia ser trabalho é a tradição de famílias na política. Porém, para isto seriam necessários dados adicionais aos disponíveis.

Vale a pena citar que os atributos financeiros são muito importantes para definir a condição de eleito nas tarefas de classificação aplicadas. Iso pode ser visto tanto no experimento de classificação quanto seleção de atributos. Porém, deve-se deixar claro que, os resultados apresentados aqui são validos somente para este caso específico. Não existe nenhuma pretensão de generalização no escopo deste trabalho.

De um ponto de vista pessoal, pode-se dizer que este trabalho também teve uma função didática. Foi possível aprender mais a fundo o funcionamento da mineração de dados e suas tarefas associadas. Também ficou claro que a etapa de limpeza e preparação de dados é uma etapa custosa e trabalhosa. Quanto as duas tarefas aplicadas neste trabalho (associação e

classificação), foi possível entender a diferença no tipo de conhecimento gerada por cada uma das duas.

7.2 TRABALHOS FUTUROS

Como possíveis desdobramentos deste trabalho, podemos destacar quatro linhas que poderiam se mostrar interessantes. Primeiro, a aplicação de outros algoritmos de mineração de dados ao repositório de dados eleitorais. Segundo, investigar outras formas de resolver o problema de balanceamento das instancias. Terceiro, estudar novos meta-atributos que possam ser significativos para a classificação. Quarto, aplicar as tarefas clássicas de mineração que não foram aplicadas. E, por fim, consulta com especialistas do problema. Aplicar outros algoritmos na base de dados é uma continuação lógica do trabalho proposto. Pode-se afirmar isto, uma vez que a base deste estudo foi a aplicação de diferentes técnicas de classificação no problema proposto. Mesmo sendo simples, existe uma grande quantidade de algoritmos que podem ser aplicados. Podendo ser desde técnicas mais próximas de regressões ou até mesmo classificadores baseados em regras difusas.

O balanceamento das instâncias de treino/teste é um desafio interessante ligado ao repositório de dados eleitorais. A grande diferença entre a quantidade de eleitos e não eleitos (ou as possíveis variações de resultados de uma candidatura) são uma característica estrutural destes dados. Isto tem uma causa bem clara: existem muito mais candidatos do que cargos eletivos disponíveis. Esta questão ganha importância quando se trata de classificar dados, pois muitos dos algoritmos de classificação, exigem que os rótulos das instancias sejam balanceados. Algumas formas de tentar resolver este problema, envolvem em explorar outras formas de balanceamento (outras formas de resampling) ou buscar classificadores que podem funcionar bem quando as classes não estão bem equilibradas.

A criação de novos atributos tanto provenientes da base de dados do TSE quanto de outras bases, também seria um desdobramento interessante. Neste trabalho, foram criados dois atributos, porém somente um se mostrou interessante para a classificação (vezes eleito). Outros temas ligados ao domínio do problema poderiam ser explorados, tais como a herança política de um candidato ou se existe algum tipo de influência do executivo no legislativo.

A grande limitação deste estudo foi a ausência de um especialista de domínio do problema para auxiliar o processo como um todo. A presença deste ator possibilitaria uma identificação de vários itens que poderiam ser interessantes de ser investigados. Por exemplo, quanto as meta métricas criadas para este estudo, um especialista de domínio poderia guiar esta tarefa.

Outra questão que torna evidente a possibilidade de contribuição deste profissional, seria na fase de validação das informações levantadas.

Por fim, este trabalho somente aplicou duas das quatro tarefas clássicas da mineração de dados. O agrupamento e detecção de anomalias não foram usadas pela limitação de tempo/recursos deste trabalho, fazendo com que, inicialmente, não exista nenhum impedimento técnico para sua aplicação. Seria preciso apenas definir as diretrizes para ambas as tarefas.

REFERÊNCIAS

BOSCHI, R. S.; OLIVEIRA, S. R. d. M.; ASSAD, E. D. Técnicas de mineração de dados para análise da precipitação pluvial decenal no rio grande do sul. **Eng. Agríc**, 2011.

BRASIL, C. d. . **Constituição da República Federativa do Brasil**. [S.l.: s.n.], 1988.

BRASIL, C. G. d. U. Política brasileira de acesso à informações públicas. 2010.

BRASIL, P. d. R. **Lei n 12.527 de 18 de novembro de 2011**. [S.l.: s.n.], 2011.

CARDOSO, O. N. P.; MACHADO, R. T. M. Gestão do conhecimento usando data mining: estudo de caso na universidade federal de lavras. **Rev. Adm. Pública**, 2008.

DUDA, R. O.; HARD, P. E.; STORK, D. G. **Pattern Classification**. [S.l.: s.n.], 2001.

FAYYAD, U.; PIATESKY-SHAPIRO, G.; SMYTH, P. The kdd processfor extracting useful knowledge from volumes of data. **Communications of the ACM November Vol. 39**, v. 39, n. 11, 1996.

FAYYAD, U. et al. (Ed.). **Advances in Knowledge Discovery and Data Mining**. [S.l.]: MIT Press, 1996.

GALVAO, N. D.; MARIN, H. d. F. Técnica de mineração de dados: uma revisão da literatura. 2009.

GOEBEL, M.; GRUENWALD, L. A survey of data mining and knowleadge discovery software tools. 1999.

GOLDSCHMIDT, R.; PASSOS, E. **Data Mining: um guia prático**. Rio de Janeiro: [s.n.], 2005.

HALL, M. et al. The weka data mining software: An update. **SIGKDD Explorations**, 2009.

HAYKIN, S. **Redes Neurais**. [S.l.]: bookman, 1999.

HWANJO, Y.; KIM, S. Svm tutorial: Classification, regression, and ranking. 2012.

JAIN, A. K.; MAO, J. Artificial neural networks: A tutorial. 1996.

KIMBALL, R. **Data Warehouse Toolkit**. São Paulo: Makron books, 1996.

KOHAVI, R.; QUINLAN, R. Decision tree discovery. **Handbook of Data Mining and Knowledge Discovery**, 1999.

MALUCELLI, A. Classificação de micro áreas de risco com uso de mineração de dados. **Rev. Saúde Pública**, 2010.

PIATETSKY-SHAPIRO, G. Knowledge discovery in real databases: A report on the ijcai-89 work- shop. **AI Magazine**, v. 11, 1991.

QUONIAM, L. et al. Inteligência obtida pela aplicação de data mining em base de teses francesas sobre o brasil. **Ci. Inf. [online]**, 2001.

SILBERSCHATZ, A. **Sistema de Banco de Dados**. [S.l.]: Pearson Education do Brasil, 1999.

STEINBACH, M.; TAN, P.; KUMAR, V. **Introduction to Data Mining**. [S.l.]: Pearson Education, 2006.

TSE. **repositório de dados eleitorais**. 2013. Disponível em: <http://www.tse.jus.br/eleicoes/repositorio-de-dados-eleitorais>.

TSE. **leia-me**. 2014.

VIANNA, R. C. X. F. Mineração de dados e características da mortalidade infantil. **Cad. Saúde Pública**, 2010.

WEKA. **Weka 3: Data Mining Software in Java**. 2013. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/>.

WITTEN, I.; FRANK, E. **Data Mining: Practical Machine Learning Tools and Techniques**. [S.l.]: Morgan Kaufmann, 2005.

APÊNDICE A – COMANDOS SQL PARA A CARGA DO ESQUEMA OLAP

A.1 CRIAR_BEM_OLAP_2012.SQL

```
create table olap.bem as
select
sum(case when bemtipo_codigo = 1 then valor else 0 end) as
BEM_predio_residencial ,
sum(case when bemtipo_codigo = 2 then valor else 0 end) as
BEM_predio_comercial ,
sum(case when bemtipo_codigo = 3 then valor else 0 end) as
BEM_galpao ,
sum(case when bemtipo_codigo = 11 then valor else 0 end) as
BEM_apartamento ,
sum(case when bemtipo_codigo = 12 then valor else 0 end) as
BEM_casa ,
sum(case when bemtipo_codigo = 13 then valor else 0 end) as
BEM_terreno ,
sum(case when bemtipo_codigo = 14 then valor else 0 end) as
BEM_terra_nua ,
sum(case when bemtipo_codigo = 15 then valor else 0 end) as
BEM_sala_ou_conjunto ,
sum(case when bemtipo_codigo = 16 then valor else 0 end) as
BEM_construcao ,
sum(case when bemtipo_codigo = 17 then valor else 0 end) as
BEM_benfeitorias ,
sum(case when bemtipo_codigo = 18 then valor else 0 end) as
BEM_loja ,
sum(case when bemtipo_codigo = 19 then valor else 0 end) as
BEM_outros_bens_imoveis ,
```

```
sum(case when bemtipo_codigo = 21 then valor else 0 end) as
BEM_veiculo_automotor_terrestre ,
sum(case when bemtipo_codigo = 22 then valor else 0 end) as
BEM_aeronave ,
sum(case when bemtipo_codigo = 23 then valor else 0 end) as
BEM_embarcacao ,
sum(case when bemtipo_codigo = 24 then valor else 0 end) as
BEM_bem_relacionado_com_atividade_autonoma ,
sum(case when bemtipo_codigo = 25 then valor else 0 end) as
BEM_joia_quadro_obj_arte_colecao_etc ,
sum(case when bemtipo_codigo = 26 then valor else 0 end) as
BEM_linha_telefonica ,
sum(case when bemtipo_codigo = 29 then valor else 0 end) as
BEM_outros_bens_moveis ,
sum(case when bemtipo_codigo = 31 then valor else 0 end) as
BEM_acoes ,
sum(case when bemtipo_codigo = 32 then valor else 0 end) as
BEM_quotas_ou_quinhoes_capital ,
sum(case when bemtipo_codigo = 39 then valor else 0 end) as
BEM_outras_particit_societarias ,
sum(case when bemtipo_codigo = 41 then valor else 0 end) as
BEM_poupanca ,
sum(case when bemtipo_codigo = 45 then valor else 0 end) as
BEM_aplicacao_renda_fixa ,
sum(case when bemtipo_codigo = 46 then valor else 0 end) as
BEM_ouro_ativofinanceiro ,
sum(case when bemtipo_codigo = 47 then valor else 0 end) as
BEM_mercados_futuros ,
sum(case when bemtipo_codigo = 49 then valor else 0 end) as
BEM_outras_aplicacoes_e_investimentos ,
sum(case when bemtipo_codigo = 51 then valor else 0 end) as
BEM_credito_de_emprestimo ,
sum(case when bemtipo_codigo = 52 then valor else 0 end) as
BEM_credito_de_alienacao ,
sum(case when bemtipo_codigo = 53 then valor else 0 end) as
```

```
BEM_planopait_caderneta_de_peculio ,
sum(case when bemtipo_codigo = 54 then valor else 0 end) as
BEM_poupanca_para_imovel ,
sum(case when bemtipo_codigo = 59 then valor else 0 end) as
BEM_outros_creditos_poupancas ,
sum(case when bemtipo_codigo = 61 then valor else 0 end) as
BEM_conta_corrente_pais ,
sum(case when bemtipo_codigo = 62 then valor else 0 end) as
BEM_conta_corrente_exterior ,
sum(case when bemtipo_codigo = 63 then valor else 0 end) as
BEM_moeda_nacional ,
sum(case when bemtipo_codigo = 64 then valor else 0 end) as
BEM_moeda_estrangeira ,
sum(case when bemtipo_codigo = 69 then valor else 0 end) as
BEM_outros_depositos_vista ,
sum(case when bemtipo_codigo = 71 then valor else 0 end) as
BEM_fif ,
sum(case when bemtipo_codigo = 72 then valor else 0 end) as
BEM_fundo_de_aplicacao_quotas ,
sum(case when bemtipo_codigo = 73 then valor else 0 end) as
BEM_fundo_de_capitalizacao ,
sum(case when bemtipo_codigo = 74 then valor else 0 end) as
BEM_fundos_de_acoes ,
sum(case when bemtipo_codigo = 79 then valor else 0 end) as
BEM_outros_fundos ,
sum(case when bemtipo_codigo = 91 then valor else 0 end) as
BEM_licenca_concessoes_especiais ,
sum(case when bemtipo_codigo = 92 then valor else 0 end) as
BEM_titulo_de_clube ,
sum(case when bemtipo_codigo = 93 then valor else 0 end) as
BEM_direito_de_autor_inventor_patente ,
sum(case when bemtipo_codigo = 94 then valor else 0 end) as
BEM_direito_lavra ,
sum(case when bemtipo_codigo = 95 then valor else 0 end) as
BEM_consorcio_nao_contemplado ,
```

```
sum(case when bemtipo_codigo = 96 then valor else 0 end) as
BEM_leasing ,
sum(case when bemtipo_codigo = 97 then valor else 0 end) as
BEM_vgbl ,
sum(case when bemtipo_codigo = 99 then valor else 0 end) as
BEM_outros_bens_e_direitos ,
sum(valor) as
BEM_total,
sequencial as sequencial
from oltp.bem
group by candidatura_sequencial;
```

A.2 CRIAR_CANDIDATURA_OLAP_2012.SQL

```
–drop table olap.candidatura;
create table olap.candidatura as
SELECT
co.tituloeleitoral,
ca.sequencial,
uf ,
co.nome,
c.descricao as cargo,
ca.ue_codigo as ue,
ca.numero,
ca.partido_sigla,
ca.uf,
ca.turno,
ca.despesa_maxima_campanha,
o.descricao as ocupacao,
2010-cast((’19’ || to_char(to_date(co.nascimento,’dd-mm-yy’),’yy’)) as int) as idade,
s.descricao as sexo,
g.descricao as grauinstrucao,
e.descricao as estadocivil,
n.descricao as nacionalidade,
co.uf_nascimento,
```

```
co.municipionascimento_descricao,
el.eleito as eleito_vezes,
(case
when r.codigo = 1 then 'eleito'
when r.codigo = 2 then 'eleito'
when r.codigo = 3 then 'eleito'
else 'não eleito'
end) as resultado
FROM oltp.candidatura as ca
join oltp.resultado as r on
ca.resultado_codigo = r.codigo
join oltp.cargo as c on
ca.cargo_codigo = c.codigo
join oltp.candidato as co on
ca.tituloeleitoral = co.tituloeleitoral
join oltp.sexo as s on
co.sexo_codigo = s.codigo
join oltp.estadocivil as e on
co.estadocivil_codigo = e.codigo
join oltp.grauinstrucao as g on
co.grauinstrucao_codigo = g.codigo
join oltp.ocupacao as o on
co.ocupacao_codigo = o.codigo
join oltp.nacionalidade as n on
co.nacionalidade_codigo = n.codigo
left join temp.eleitos_quantidade as el on
co.tituloeleitoral = el.tituloeleitoral
```

A.3 CRIAR_RECEITA_OLAP_2012.SQL

```
–drop table olap.receita;
create table olap.receita as
select
sum(case when receitatipo_codigo = 'Recursos de pessoas físicas' then valor else 0 end)
```

```
as RECEITA_recursos_pessoas_fisicas,
sum(case when receitatipo_codigo = ‘Recursos de pessoas jurídicas’ then valor else 0 end)
as RECEITA_recursos_de_pessoas_jurididas,
sum(case when receitatipo_codigo = ‘Recursos próprios’ then valor else 0 end)
as RECEITA_recursos_proprios,
sum(case when receitatipo_codigo = ‘Recursos de outros candidatos/comitês’ then valor else 0
end)
as RECEITA_recursos_de_outros_candidatos_ou_comites,
sum(case when receitatipo_codigo = ‘Recursos de partido político’ then valor else 0 end)
as RECEITA_recursos_de_partido_politico,_
sum(case when receitatipo_codigo = ‘Comercialização de bens e/ou realização de eventos’ then
valor else 0 end)
as RECEITA_doacoes_relativas_a_comercializacao,
sum(case when receitatipo_codigo = ‘Recursos de origens não identificadas’ then valor else 0
end)
as RECEITA_recursos_de_origens_nao_identificaveis,
sum(case when receitatipo_codigo = ‘Rendimentos de aplicações financeiras’ then valor else 0
end)
as RECEITA_rendimento_de_aplicacao_financeira,
sum(case when receitatipo_codigo = ‘Recursos de doações pela Internet’ then valor else 0 end)
as RECEITA_doacoes_pela_internet,
sum(valor) as receita_total,
candidatura_codigo as candidatura_codigo_original
from oltp.receita
group by candidatura_codigo;
```

A.4 CRIAR_SECAO_OLAP_2012.SQL

```
–drop table olap.secao;
create table olap.secao as SELECT
ue_codigo,
uf_sigla,
sum(quantidade_aptos) as aptos,
sum(quantidade_comparecimentos) as comparecimentos,
sum(quantidade_abstencoes) as abstencoes,
```

```
sum(quantidade_nominais) as nominais,
sum(quantidade_brancos) as brancos,
sum(quantidade_nulos) as nulos,
sum(quantidade_legenda) as legenda,
sum(quantidade_anulados) as anulados
FROM oltp.secaoeleitoral join
oltp.municipio m on m.codigo = ue_codigo
group by ue_codigo,nome,uf_sigla;
```

A.5 CRIAR_VIEW_DEPFED_2012.SQL

```
– DROP VIEW olap.deputado_federal_dadospessoais;
CREATE OR REPLACE VIEW olap.vereadores_dadospessoais AS
SELECT candidatura.tituloeleitoral, candidatura.sequencial, candidatura.uf,
candidatura.nome, candidatura.cargo, candidatura.ue, candidatura.numero,
candidatura.partido_sigla, candidatura.turno,
candidatura.despesa_maxima_campanha, candidatura.ocupacao,
candidatura.idade, candidatura.sexo, candidatura.grauinstrucao,
candidatura.estadocivil, candidatura.nacionalidade,
candidatura.uf_nascimento, candidatura.municipionascimento_descricao,
COALESCE(v.comparecimentos,0) AS votacao_comparecimentos,
CASE
WHEN candidatura.ue::text <> candidatura.uf_nascimento::text THEN 1
ELSE 0
END AS nasceu_no_estado,
COALESCE(candidatura.eleito_vezes,0) as eleito_vezes,
candidatura.resultado
FROM olap.candidatura
LEFT JOIN olap.secao v ON candidatura.ue::text = v.ue_codigo::text
WHERE candidatura.cargo::text = 'VEREADOR'::text;
ALTER TABLE olap.vereadores_dadospessoais
OWNER TO postgres;
```

A.6 CRIAR_BEM_OLAP_2010.SQL

```
–drop table olap.bem_estrela;
create table olap.bem as
select
sum(case when bemtipo_codigo = 1 then valor else 0 end)
as BEM_predio_residencial ,
sum(case when bemtipo_codigo = 2 then valor else 0 end)
as BEM_predio_comercial ,
sum(case when bemtipo_codigo = 3 then valor else 0 end)
as BEM_galpao ,
sum(case when bemtipo_codigo = 11 then valor else 0 end)
as BEM_apartamento ,
sum(case when bemtipo_codigo = 12 then valor else 0 end)
as BEM_casa ,
sum(case when bemtipo_codigo = 13 then valor else 0 end)
as BEM_terreno ,
sum(case when bemtipo_codigo = 14 then valor else 0 end)
as BEM_terra_nua ,
sum(case when bemtipo_codigo = 15 then valor else 0 end)
as BEM_sala_ou_conjunto ,
sum(case when bemtipo_codigo = 16 then valor else 0 end)
as BEM_construcao ,
sum(case when bemtipo_codigo = 17 then valor else 0 end)
as BEM_benfeitorias ,
sum(case when bemtipo_codigo = 18 then valor else 0 end)
as BEM_loja ,
sum(case when bemtipo_codigo = 19 then valor else 0 end)
as BEM_outros_bens_imoveis ,
sum(case when bemtipo_codigo = 21 then valor else 0 end)
as BEM_veiculo_automotor_terrestre ,
sum(case when bemtipo_codigo = 22 then valor else 0 end)
as BEM_aeronave ,
sum(case when bemtipo_codigo = 23 then valor else 0 end)
as BEM_embarcacao ,
sum(case when bemtipo_codigo = 24 then valor else 0 end)
```

```
as BEM_bem_relacionado_com_atividade_autonoma ,
sum(case when bemtipo_codigo = 25 then valor else 0 end)
as BEM_joia_quadro_obj_arte_colecao_etc ,
sum(case when bemtipo_codigo = 26 then valor else 0 end)
as BEM_linha_telefonica ,
sum(case when bemtipo_codigo = 29 then valor else 0 end)
as BEM_outros_bens_moveis ,
sum(case when bemtipo_codigo = 31 then valor else 0 end)
as BEM_acoes ,
sum(case when bemtipo_codigo = 32 then valor else 0 end)
as BEM_quotas_ou_quinhoes_capital ,
sum(case when bemtipo_codigo = 39 then valor else 0 end)
as BEM_outras_particit_societarias ,
sum(case when bemtipo_codigo = 41 then valor else 0 end)
as BEM_poupanca ,
sum(case when bemtipo_codigo = 45 then valor else 0 end)
as BEM_aplicacao_renda_fixa ,
sum(case when bemtipo_codigo = 46 then valor else 0 end)
as BEM_ouro_ativofinanceiro ,
sum(case when bemtipo_codigo = 47 then valor else 0 end)
as BEM_mercados_futuros ,
sum(case when bemtipo_codigo = 49 then valor else 0 end)
as BEM_outras_aplicacoes_e_investimentos ,
sum(case when bemtipo_codigo = 51 then valor else 0 end)
as BEM_credito_de_emprestimo ,
sum(case when bemtipo_codigo = 52 then valor else 0 end)
as credito_de_alienacao ,
sum(case when bemtipo_codigo = 53 then valor else 0 end)
as BEM_planopait_caderneta_de_peculio ,
sum(case when bemtipo_codigo = 54 then valor else 0 end)
as BEM_poupanca_para_imovel ,
sum(case when bemtipo_codigo = 59 then valor else 0 end)
as BEM_outros_creditos_poupancas ,
sum(case when bemtipo_codigo = 61 then valor else 0 end)
as BEM_conta_corrente_pais ,
```

```
sum(case when bemtipo_codigo = 62 then valor else 0 end)
as BEM_conta_corrente_exterior ,
sum(case when bemtipo_codigo = 63 then valor else 0 end)
as BEM_moeda_nacional ,
sum(case when bemtipo_codigo = 64 then valor else 0 end)
as BEM_moeda_estrangeira ,
sum(case when bemtipo_codigo = 69 then valor else 0 end)
as BEM_outros_depositos_vista ,
sum(case when bemtipo_codigo = 71 then valor else 0 end)
as BEM_fif ,
sum(case when bemtipo_codigo = 72 then valor else 0 end)
as BEM_fundo_de_aplicacao_quotas ,
sum(case when bemtipo_codigo = 73 then valor else 0 end)
as BEM_fundo_de_capitalizacao ,
sum(case when bemtipo_codigo = 74 then valor else 0 end)
as BEM_fundos_de_acoes ,
sum(case when bemtipo_codigo = 79 then valor else 0 end)
as BEM_outros_fundos ,
sum(case when bemtipo_codigo = 91 then valor else 0 end)
as BEM_licenca_concessoes_especiais ,
sum(case when bemtipo_codigo = 92 then valor else 0 end)
as BEM_titulo_de_clube ,
sum(case when bemtipo_codigo = 93 then valor else 0 end)
as BEM_direito_de_autor_inventor_patente ,
sum(case when bemtipo_codigo = 94 then valor else 0 end)
as BEM_direito_lavra ,
sum(case when bemtipo_codigo = 95 then valor else 0 end)
as BEM_consorcio_nao_contemplado ,
sum(case when bemtipo_codigo = 96 then valor else 0 end)
as BEM_leasing ,
sum(case when bemtipo_codigo = 97 then valor else 0 end)
as BEM_vgbl ,
sum(case when bemtipo_codigo = 99 then valor else 0 end)
as BEM_outros_bens_e_direitos ,
sum(valor)
```

```
as BEM_total,
candidatura_codigo
as candidatura_codigo_original
from oltp.bem
group by candidatura_codigo;
```

A.7 CRIAR_CANDIDATURA_OLAP_2010.SQL

```
drop table olap.candidatura;
create table olap.candidatura as
SELECT
ca.codigo,
co.tituloeleitoral,
ca.sequencial,
co.nome,
c.descricao as cargo,
ca.ue_codigo as ue,
ca.numero,
ca.partido_sigla,
ca.turno,
ca.despesa_maxima_campanha,
o.descricao as ocupacao,
2010-cast((’19’ || to_char(to_date(co.nascimento,’dd-Mon-yy’),’yy’)) as int) as idade,
s.descricao as sexo,
g.descricao as grauinstrucao,
e.descricao as estadocivil,
n.descricao as nacionalidade,
co.uf_nascimento,
co.municipionascimento_descricao,
el.eleito as eleito_vezes,
(case
when r.codigo = 1 then ’eleito’
when r.codigo = 5 then ’eleito’
else ’não eleito’
end) as resultado
```

```
FROM oltp.candidatura as ca
join oltp.resultado as r on
ca.resultado_codigo = r.codigo
join oltp.cargo as c on
ca.cargo_codigo = c.codigo
join oltp.candidato as co on
ca.tituloeleitoral = co.tituloeleitoral
join oltp.sexo as s on
co.sexo_codigo = s.codigo
join oltp.estadocivil as e on
co.estadocivil_codigo = e.codigo
join oltp.grauinstrucao as g on
co.grauinstrucao_codigo = g.codigo
join oltp.ocupacao as o on
co.ocupacao_codigo = o.codigo
join oltp.nacionalidade as n on
co.nacionalidade_codigo = n.codigo
join temp.eleitos_quantidade as el on
co.tituloeleitoral = el.tituloeleitoral
```

A.8 CRIAR_DESPESA_OLAP_2010.SQL

```
–drop table olap.despesa;
create table olap.despesa as
select

sum(case when despesatipo_codigo = 3 then valor else 0 end)
as DESPESA_baixas_estimaveis_dinheiro ,
sum(case when despesatipo_codigo = 5 then valor else 0 end)
as DESPESA_publicicada_placas_estandartes_faixas ,
sum(case when despesatipo_codigo = 7 then valor else 0 end)
as DESPESA_com_pessoal ,
sum(case when despesatipo_codigo = 9 then valor else 0 end)
as DESPESA_producao_radio_video_tv ,
```

```
sum(case when despesatipo_codigo = 10 then valor else 0 end)
as DESPESA_taxas_encargos_bancarios ,
sum(case when despesatipo_codigo = 11 then valor else 0 end)
as DESPESA_publicidade_impressos ,
sum(case when despesatipo_codigo = 12 then valor else 0 end)
as DESPESA_pagina_internet ,
sum(case when despesatipo_codigo = 13 then valor else 0 end)
as DESPESA_servico_terceiros ,
sum(case when despesatipo_codigo = 14 then valor else 0 end)
as DESPESA_cessao_locacao_veiculos ,
sum(case when despesatipo_codigo = 15 then valor else 0 end)
as DESPESA_combustiveis_lubrificantes ,
sum(case when despesatipo_codigo = 18 then valor else 0 end)
as DESPESA_locacao_cessao_moveis ,
sum(case when despesatipo_codigo = 19 then valor else 0 end)
as DESPESA_locacao_cessao_imoveis ,
sum(case when despesatipo_codigo = 20 then valor else 0 end)
as DESPESA_producao_jingles_vinhetas_slogans ,
sum(case when despesatipo_codigo = 21 then valor else 0 end)
as DESPESA_publicidade_carro_som ,
sum(case when despesatipo_codigo = 22 then valor else 0 end)
as DESPESA_publicidade_jornal_revista ,
sum(case when despesatipo_codigo = 23 then valor else 0 end)
as DESPESA_telefone ,
sum(case when despesatipo_codigo = 24 then valor else 0 end)
as DESPESA_alimentacao ,
sum(case when despesatipo_codigo = 25 then valor else 0 end)
as DESPESA_agua ,
sum(case when despesatipo_codigo = 26 then valor else 0 end)
as DESPESA_preinstalacao_comite ,
sum(case when despesatipo_codigo = 27 then valor else 0 end)
as DESPESA_material_expediente ,
sum(case when despesatipo_codigo = 28 then valor else 0 end)
as DESPESA_postais ,
sum(case when despesatipo_codigo = 29 then valor else 0 end)
```

```
as DESPESA_diversas ,
sum(case when despesatipo_codigo = 30 then valor else 0 end)
as DESPESA_energia_eletrica ,
sum(case when despesatipo_codigo = 31 then valor else 0 end)
as DESPESA_bens_permanentes ,
sum(case when despesatipo_codigo = 32 then valor else 0 end)
as DESPESA_deslocamento_transporte ,
sum(case when despesatipo_codigo = 33 then valor else 0 end)
as DESPESA_impostos_taxas ,
sum(case when despesatipo_codigo = 34 then valor else 0 end)
as DESPESA_evento_promocao_candidatura ,
sum(case when despesatipo_codigo = 35 then valor else 0 end)
as DESPESA_reembolso_eleitores ,
sum(case when despesatipo_codigo = 36 then valor else 0 end)
as DESPESA_pesuisa_testes_eleitorais ,
sum(case when despesatipo_codigo = 37 then valor else 0 end)
as DESPESA_publicidade_telemarketing ,
sum(case when despesatipo_codigo = 38 then valor else 0 end)
as DESPESA_doacoes_outros_partidos_candidatos ,
sum(case when despesatipo_codigo = 39 then valor else 0 end)
as DESPESA_encargos_sociais ,
sum(case when despesatipo_codigo = 41 then valor else 0 end)
as DESPESA_comicios ,
sum(case when despesatipo_codigo = 42 then valor else 0 end)
as DESPESA_multas_eleitorais ,
sum(valor)
as despesa_total,
candidatura_codigo
as candidatura_codigo_original
from olap.despesa
group by candidatura_codigo;
```

A.9 CRIAR_RECEITA_OLAP_2010.SQL

```
–drop table olap.receita;
create table olap.receita as
select
sum(case when receitatipo_codigo = 2 then valor else 0 end)
as RECEITA_recursos_pessoas_fisicas,
sum(case when receitatipo_codigo = 6 then valor else 0 end)
as RECEITA_recursos_de_pessoas_jurididas,
sum(case when receitatipo_codigo = 9 then valor else 0 end)
as RECEITA_recursos_proprios,
sum(case when receitatipo_codigo = 11 then valor else 0 end)
as RECEITA_recursos_de_outros_candidatos_ou_comites,
sum(case when receitatipo_codigo = 12 then valor else 0 end)
as RECEITA_recursos_de_partido_politico,
sum(case when receitatipo_codigo = 13 then valor else 0 end)
as RECEITA_doacoes_relativas_a_comercializacao,
sum(case when receitatipo_codigo = 16 then valor else 0 end)
as RECEITA_recursos_de_origens_nao_identificaveis,
sum(case when receitatipo_codigo = 19 then valor else 0 end)
as RECEITA_rendimento_de_aplicacao_financeira,
sum(case when receitatipo_codigo = 20 then valor else 0 end)
as RECEITA_doacoes_pela_internet,
sum(valor) as receita_total,
candidatura_codigo as candidatura_codigo_original
from oltp.receita
group by candidatura_codigo;
```

A.10 CRIAR_SECAO_OLAP_2010.SQL

```
drop table olap.secao;
create table olap.secao as
SELECT uf_sigla,
sum(quantidade_aptos) as aptos,
sum(quantidade_comparecimentos) as comparecimentos,
```

```
sum(quantidade_abstencoes) as abstencoes,
sum(quantidade_nominais) as nominais,
sum(quantidade_brancos) as brancos,
sum(quantidade_nulos) as nulos,
sum(quantidade_legenda) as legenda,
sum(quantidade_anulados) as anulados
FROM oltp.secaoeleitoral join
oltp.municipio m on m.codigo = ue_codigo
group by uf_sigla;
```

APÊNDICE B – SAÍDAS DO WEKA PARA AS REGRAS DE ASSOCIAÇÃO

B.1 VEREADORES

B.1.1 TODOS RESTRITO

=== Run information ===

Scheme: weka.associations.Apriori
-N 100 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Relation: QueryResult
weka.filters.unsupervised.attribute.Remove-R71
weka.filters.unsupervised.instance.NonSparseToSparse
weka.filters.unsupervised.attribute.RemoveUseless-M99.0
weka.filters.unsupervised.attribute.Remove-R2,4
weka.filters.unsupervised.attribute.Remove-R10
weka.filters.unsupervised.attribute.Remove-R10-11
weka.filters.unsupervised.attribute.NumericToNominal-R10
weka.filters.unsupervised.attribute.Remove-R55-58
weka.filters.unsupervised.attribute.Remove-R10-21
weka.filters.unsupervised.attribute.Remove-R31-42
weka.filters.unsupervised.attribute.Remove-R10-30
weka.filters.unsupervised.attribute.Remove-R2
weka.filters.supervised.attribute.Discretize-Rfirst-last
weka.filters.unsupervised.attribute.Remove-R9

Instances: 424296

Attributes: 11 uf, partido_sigla, despesa_maxima_campanha, ocupacao, idade, sexo, grauinstrucao, estadocivil, votacao_comparecimentos, eleito_vezes, resultado

=== Associator model (full training set) ===

Apriori

=======

Minimum support: 0.1 (42430 instances)

Minimum metric <confidence>: 0.9
Number of cycles performed: 18
Generated sets of large itemsets:
Size of set of large itemsets L(1): 21
Size of set of large itemsets L(2): 59
Size of set of large itemsets L(3): 49
Size of set of large itemsets L(4): 21
Size of set of large itemsets L(5): 3
Best rules found:
1. sexo = FEMININO estadocivil = SOLTEIRO(A) resultado = não eleito 47381 ==> eleito_vezes = (-inf-0.5] 46776 conf: (0.99) lift: (1.12) lev: (0.01) [5082] conv:(9.38)
2. idade = (28.5-46.5] sexo = FEMININO resultado = não eleito 64099 ==> eleito_vezes = (-inf-0.5] 62750 conf: (0.98) lift: (1.11) lev: (0.01) [6344] conv:(5.7)
3. sexo = FEMININO estadocivil = SOLTEIRO(A) 49263 ==> eleito_vezes = (-inf-0.5] 48093 conf: (0.98) lift: (1.11) lev: (0.01) [4743] conv:(5.05)
4. sexo = FEMININO grauinstrucao = ENSINO MÉDIO COMPLETO resultado = não eleito 48576 ==> eleito_vezes = (-inf-0.5] 47412 conf: (0.98) lift: (1.11) lev: (0.01) [4666] conv:(5)
5. sexo = FEMININO resultado = não eleito 126734 ==> eleito_vezes = (-inf-0.5] 123544 conf: (0.97) lift: (1.11) lev: (0.03) [12022] conv:(4.77)
6. ocupacao = OUTROS resultado = não eleito 58011 ==> eleito_vezes = (-inf-0.5] 56505 conf: (0.97) lift: (1.11) lev: (0.01) [5457] conv:(4.62)
7. sexo = FEMININO estadocivil = SOLTEIRO(A) eleito_vezes = (-inf-0.5] 48093 ==> resultado = não eleito 46776 conf: (0.97) lift: (1.13) lev: (0.01) [5439] conv:(5.13)
8. grauinstrucao = ENSINO MÉDIO COMPLETO estadocivil = SOLTEIRO(A) resultado = não eleito 48171 ==> eleito_vezes = (-inf-0.5] 46686 conf: (0.97) lift: (1.1) lev: (0.01) [4297] conv:(3.89)
9. estadocivil = SOLTEIRO(A) resultado = não eleito 119912 ==> eleito_vezes = (-inf-0.5] 115909 conf: (0.97) lift: (1.1) lev: (0.02) [10390] conv:(3.59)
10. sexo = FEMININO estadocivil = CASADO(A) resultado = não eleito 60297 ==> eleito_vezes

= (-inf-0.5] 58280 conf: (0.97) lift: (1.1) lev: (0.01) [5220] conv:(3.59)

11. sexo = FEMININO grauinstrucao = ENSINO MÉDIO COMPLETO eleito_vezes = (-inf-0.5] 49087 ==> resultado = não eleito 47412 conf: (0.97) lift: (1.12) lev: (0.01) [5221] conv:(4.11)

12. idade = (28.5-46.5] estadocivil = SOLTEIRO(A) resultado = não eleito 66394 ==> eleito_vezes = (-inf-0.5] 63866 conf: (0.96) lift: (1.09) lev: (0.01) [5441] conv:(3.15)

13. sexo = FEMININO estadocivil = SOLTEIRO(A) 49263 ==> resultado = não eleito 47381 conf: (0.96) lift: (1.12) lev: (0.01) [5039] conv:(3.68)

14. sexo = FEMININO eleito_vezes = (-inf-0.5] 128504 ==> resultado = não eleito 123544 conf: (0.96) lift: (1.12) lev: (0.03) [13093] conv:(3.64)

15. idade = (28.5-46.5] sexo = FEMININO 68694 ==> eleito_vezes = (-inf-0.5] 65856 conf: (0.96) lift: (1.09) lev: (0.01) [5407] conv:(2.9)

16. sexo = FEMININO grauinstrucao = ENSINO MÉDIO COMPLETO 51257 ==> eleito_vezes = (-inf-0.5] 49087 conf: (0.96) lift: (1.09) lev: (0.01) [3982] conv:(2.83)

17. ocupacao = OUTROS 62949 ==> eleito_vezes = (-inf-0.5] 60181 conf: (0.96) lift: (1.09) lev: (0.01) [4787] conv:(2.73)

18. sexo = FEMININO 134655 ==> eleito_vezes = (-inf-0.5] 128504 conf: (0.95) lift: (1.08) lev: (0.02) [10011] conv:(2.63)

19. sexo = MASCULINO estadocivil = SOLTEIRO(A) resultado = não eleito 72531 ==> eleito_vezes = (-inf-0.5] 69133 conf: (0.95) lift: (1.08) lev: (0.01) [5308] conv:(2.56)

20. idade = (28.5-46.5] sexo = FEMININO eleito_vezes = (-inf-0.5] 65856 ==> resultado = não eleito 62750 conf: (0.95) lift: (1.11) lev: (0.01) [6146] conv:(2.98)

21. uf = SP resultado = não eleito 66686 ==> eleito_vezes = (-inf-0.5] 63404 conf: (0.95) lift: (1.08) lev: (0.01) [4722] conv:(2.44)

22. sexo = FEMININO estadocivil = SOLTEIRO(A) 49263 ==> eleito_vezes = (-inf-0.5] resultado = não eleito 46776 conf: (0.95) lift: (1.18) lev: (0.02) [7187] conv:(3.89)

23. sexo = FEMININO estadocivil = CASADO(A) eleito_vezes = (-inf-0.5] 61389 ==> resultado = não eleito 58280 conf: (0.95) lift: (1.1) lev: (0.01) [5515] conv:(2.77)

24. sexo = FEMININO grauinstrucao = ENSINO MÉDIO COMPLETO 51257 ==> resultado = não eleito 48576 conf: (0.95) lift: (1.1) lev: (0.01) [4520] conv:(2.69)

25. idade = (28.5-46.5] grauinstrucao = ENSINO MÉDIO COMPLETO resultado = não eleito 71667 ==> eleito_vezes = (-inf-0.5] 67772 conf: (0.95) lift: (1.07) lev: (0.01) [4707] conv:(2.21)

26. idade = (28.5-46.5] resultado = não eleito 188628 ==> eleito_vezes = (-inf-0.5] 178294 conf: (0.95) lift: (1.07) lev: (0.03) [12307] conv:(2.19)

27. votacao_comparecimentos = (114073-359375] 46642 ==> eleito_vezes = (-inf-0.5] 44080

conf: (0.95) lift: (1.07) lev: (0.01) [3036] conv:(2.18)
28. uf = SP eleito_vezes = (-inf-0.5] 67311 ==> resultado = não eleito 63404 conf: (0.94) lift: (1.1) lev: (0.01) [5549] conv:(2.42)
29. sexo = FEMININO 134655 ==> resultado = não eleito 126734 conf: (0.94) lift: (1.1) lev: (0.03) [10996] conv:(2.39)
30. grauinstrucao = ENSINO MÉDIO COMPLETO resultado = não eleito 131772 ==> eleito_vezes = (-inf-0.5] 123979 conf: (0.94) lift: (1.07) lev: (0.02) [8023] conv:(2.03)
31. idade = (50.5-57.5] eleito_vezes = (-inf-0.5] 50801 ==> resultado = não eleito 47796 conf: (0.94) lift: (1.09) lev: (0.01) [4132] conv:(2.37)
32. sexo = FEMININO estadocivil = CASADO(A) 65355 ==> eleito_vezes = (-inf-0.5] 61389 conf: (0.94) lift: (1.07) lev: (0.01) [3878] conv:(1.98)
33. ocupacao = OUTROS eleito_vezes = (-inf-0.5] 60181 ==> resultado = não eleito 56505 conf: (0.94) lift: (1.09) lev: (0.01) [4778] conv:(2.3)
34. despesa_maxima_campanha = 50000 resultado = não eleito 47115 ==> eleito_vezes = (-inf-0.5] 44139 conf: (0.94) lift: (1.06) lev: (0.01) [2679] conv:(1.9)
35. grauinstrucao = ENSINO MÉDIO COMPLETO estadocivil = SOLTEIRO(A) 53956 ==> eleito_vezes = (-inf-0.5] 50538 conf: (0.94) lift: (1.06) lev: (0.01) [3058] conv:(1.89)
36. resultado = não eleito 364686 ==> eleito_vezes = (-inf-0.5] 340969 conf: (0.93) lift: (1.06) lev: (0.05) [20056] conv:(1.85)
37. idade = (28.5-46.5] estadocivil = CASADO(A) resultado = não eleito 106364 ==> eleito_vezes = (-inf-0.5] 99369 conf: (0.93) lift: (1.06) lev: (0.01) [5772] conv:(1.82)
38. estadocivil = SOLTEIRO(A) 134768 ==> eleito_vezes = (-inf-0.5] 125774 conf: (0.93) lift: (1.06) lev: (0.02) [7182] conv:(1.8)
39. idade = (28.5-46.5] sexo = FEMININO 68694 ==> resultado = não eleito 64099 conf: (0.93) lift: (1.09) lev: (0.01) [5055] conv:(2.1)
40. votacao_comparecimentos = (114073-359375] 46642 ==> resultado = não eleito 43471 conf: (0.93) lift: (1.08) lev: (0.01) [3381] conv:(2.07)
41. uf = MG resultado = não eleito 58554 ==> eleito_vezes = (-inf-0.5] 54491 conf: (0.93) lift: (1.06) lev: (0.01) [2965] conv:(1.73)
42. grauinstrucao = ENSINO FUNDAMENTAL COMPLETO resultado = não eleito 53581 ==> eleito_vezes = (-inf-0.5] 49775 conf: (0.93) lift: (1.06) lev: (0.01) [2625] conv:(1.69)
43. grauinstrucao = ENSINO FUNDAMENTAL INCOMPLETO resultado = não eleito 65069 ==> eleito_vezes = (-inf-0.5] 60395 conf: (0.93) lift: (1.05) lev: (0.01) [3136] conv:(1.67)
44. idade = (28.5-46.5] sexo = MASCULINO resultado = não eleito 124529 ==> eleito_vezes = (-inf-0.5] 115544 conf: (0.93) lift: (1.05) lev: (0.01) [5962] conv:(1.66)

45. grauinstrucao = SUPERIOR COMPLETO resultado = não eleito 65172 ==> eleito_vezes = (-inf-0.5] 60409 conf: (0.93) lift: (1.05) lev: (0.01) [3059] conv:(1.64)
46. idade = (28.5-46.5] sexo = MASCULINO grauinstrucao = ENSINO MÉDIO COMPLETO resultado = não eleito 46522 ==> eleito_vezes = (-inf-0.5] 43118 conf: (0.93) lift: (1.05) lev: (0.01) [2180] conv:(1.64)
47. grauinstrucao = ENSINO FUNDAMENTAL INCOMPLETO eleito_vezes = (-inf-0.5] 65237 ==> resultado = não eleito 60395 conf: (0.93) lift: (1.08) lev: (0.01) [4323] conv:(1.89)
48. sexo = FEMININO grauinstrucao = ENSINO MÉDIO COMPLETO 51257 ==> eleito_vezes = (-inf-0.5] resultado = não eleito 47412 conf: (0.92) lift: (1.15) lev: (0.01) [6221] conv:(2.62)
49. grauinstrucao = ENSINO MÉDIO COMPLETO estadocivil = SOLTEIRO(A) eleito_vezes = (-inf-0.5] 50538 ==> resultado = não eleito 46686 conf: (0.92) lift: (1.07) lev: (0.01) [3248] conv:(1.84)
50. grauinstrucao = ENSINO FUNDAMENTAL COMPLETO eleito_vezes = (-inf-0.5] 53941 ==> resultado = não eleito 49775 conf: (0.92) lift: (1.07) lev: (0.01) [3412] conv:(1.82)
51. sexo = FEMININO estadocivil = CASADO(A) 65355 ==> resultado = não eleito 60297 conf: (0.92) lift: (1.07) lev: (0.01) [4123] conv:(1.81)
52. grauinstrucao = ENSINO MÉDIO COMPLETO estadocivil = CASADO(A) resultado = não eleito 70655 ==> eleito_vezes = (-inf-0.5] 65116 conf: (0.92) lift: (1.05) lev: (0.01) [2941] conv:(1.53)
53. estadocivil = SOLTEIRO(A) eleito_vezes = (-inf-0.5] 125774 ==> resultado = não eleito 115909 conf: (0.92) lift: (1.07) lev: (0.02) [7805] conv:(1.79)
54. ocupacao = OUTROS 62949 ==> resultado = não eleito 58011 conf: (0.92) lift: (1.07) lev: (0.01) [3905] conv:(1.79)
55. idade = (28.5-46.5] estadocivil = SOLTEIRO(A) 75454 ==> eleito_vezes = (-inf-0.5] 69520 conf: (0.92) lift: (1.05) lev: (0.01) [3122] conv:(1.53)
56. sexo = MASCULINO grauinstrucao = ENSINO MÉDIO COMPLETO resultado = não eleito 83196 ==> eleito_vezes = (-inf-0.5] 76567 conf: (0.92) lift: (1.05) lev: (0.01) [3357] conv:(1.51)
57. uf = MG eleito_vezes = (-inf-0.5] 59299 ==> resultado = não eleito 54491 conf: (0.92) lift: (1.07) lev: (0.01) [3523] conv:(1.73)
58. idade = (28.5-46.5] estadocivil = SOLTEIRO(A) eleito_vezes = (-inf-0.5] 69520 ==> resultado = não eleito 63866 conf: (0.92) lift: (1.07) lev: (0.01) [4112] conv:(1.73)
59. sexo = FEMININO 134655 ==> eleito_vezes = (-inf-0.5] resultado = não eleito 123544 conf: (0.92) lift: (1.14) lev: (0.04) [15333] conv:(2.38)
60. idade = (28.5-46.5] sexo = MASCULINO estadocivil = CASADO(A) resultado = não

eleito 73969 ==> eleito_vezes = (-inf-0.5] 67797 conf: (0.92) lift: (1.04) lev: (0.01) [2706] conv:(1.44)

61. estadocivil = CASADO(A) resultado = não eleito 204949 ==> eleito_vezes = (-inf-0.5] 187631 conf: (0.92) lift: (1.04) lev: (0.02) [7282] conv:(1.42)

62. grauinstrucao = ENSINO MÉDIO COMPLETO eleito_vezes = (-inf-0.5] 135679 ==> resultado = não eleito 123979 conf: (0.91) lift: (1.06) lev: (0.02) [7361] conv:(1.63)

63. sexo = MASCULINO resultado = não eleito 237952 ==> eleito_vezes = (-inf-0.5] 217425 conf: (0.91) lift: (1.04) lev: (0.02) [8034] conv:(1.39)

64. idade = (28.5-46.5] sexo = FEMININO 68694 ==> eleito_vezes = (-inf-0.5] resultado = não eleito 62750 conf: (0.91) lift: (1.14) lev: (0.02) [7546] conv:(2.27)

65. eleito_vezes = (-inf-0.5] 373367 ==> resultado = não eleito 340969 conf: (0.91) lift: (1.06) lev: (0.05) [20056] conv:(1.62)

66. despesa_maxima_campanha = 50000 eleito_vezes = (-inf-0.5] 48376 ==> resultado = não eleito 44139 conf: (0.91) lift: (1.06) lev: (0.01) [2559] conv:(1.6)

67. uf = SP 73876 ==> eleito_vezes = (-inf-0.5] 67311 conf: (0.91) lift: (1.04) lev: (0.01) [2302] conv:(1.35)

68. sexo = MASCULINO estadocivil = SOLTEIRO(A) 85505 ==> eleito_vezes = (-inf-0.5] 77681 conf: (0.91) lift: (1.03) lev: (0.01) [2439] conv:(1.31)

69. sexo = MASCULINO grauinstrucao = ENSINO FUNDAMENTAL INCOMPLETO resultado = não eleito 47686 ==> eleito_vezes = (-inf-0.5] 43274 conf: (0.91) lift: (1.03) lev: (0) [1311] conv:(1.3)

70. sexo = MASCULINO grauinstrucao = ENSINO FUNDAMENTAL INCOMPLETO eleito_vezes = (-inf-0.5] 47844 ==> resultado = não eleito 43274 conf: (0.9) lift: (1.05) lev: (0.01) [2151] conv:(1.47)

71. idade = (50.5-57.5] resultado = não eleito 52914 ==> eleito_vezes = (-inf-0.5] 47796 conf: (0.9) lift: (1.03) lev: (0) [1233] conv:(1.24)

72. uf = SP 73876 ==> resultado = não eleito 66686 conf: (0.9) lift: (1.05) lev: (0.01) [3188] conv:(1.44)

73. idade = (28.5-46.5] grauinstrucao = ENSINO MÉDIO COMPLETO eleito_vezes = (-inf-0.5] 75081 ==> resultado = não eleito 67772 conf: (0.9) lift: (1.05) lev: (0.01) [3239] conv:(1.44)

74. estadocivil = CASADO(A) eleito_vezes = (-inf-0.5] 207986 ==> resultado = não eleito 187631 conf: (0.9) lift: (1.05) lev: (0.02) [8865] conv:(1.44)

75. grauinstrucao = ENSINO MÉDIO COMPLETO estadocivil = CASADO(A) eleito_vezes = (-inf-0.5] 72256 ==> resultado = não eleito 65116 conf: (0.9) lift: (1.05) lev: (0.01) [3011]

conv:(1.42)
76. sexo = MASCULINO grauinstrucao = ENSINO MÉDIO COMPLETO estadocivil = CASADO(A) resultado = não eleito 48691 ==> eleito_vezes = (-inf-0.5] 43859 conf: (0.9) lift: (1.02) lev: (0) [1012] conv:(1.21)

B.1.2 TODOS ABRANGENTE

=== Run information ===

Scheme: weka.associations.Apriori
-N 100 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.3 -S -1.0 -c -1 Relation: QueryResult
weka.filters.unsupervised.attribute.Remove-R71
weka.filters.unsupervised.instance.NonSparseToSparse
weka.filters.unsupervised.attribute.RemoveUseless-M99.0
weka.filters.unsupervised.attribute.Remove-R2,4
weka.filters.unsupervised.attribute.Remove-R10
weka.filters.unsupervised.attribute.Remove-R10-11
weka.filters.unsupervised.attribute.NumericToNominal-R10
weka.filters.unsupervised.attribute.Remove-R55-58
weka.filters.unsupervised.attribute.Remove-R10-21
weka.filters.unsupervised.attribute.Remove-R31-42
weka.filters.unsupervised.attribute.Remove-R10-30
weka.filters.unsupervised.attribute.Remove-R2
weka.filters.supervised.attribute.Discretize-Rfirst-last
weka.filters.unsupervised.attribute.Remove-R9
weka.filters.unsupervised.attribute.Remove-R9
Instances: 424296
Attributes: 10 uf, partido_sigla, despesa_maxima_campanha, ocupacao, idade, sexo, grauinstrucao, estadocivil, eleito_vezes, resultado

=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.3 (127289 instances)

Minimum metric <confidence>: 0.7
Number of cycles performed: 14
Generated sets of large itemsets:
Size of set of large itemsets L(1): 8
Size of set of large itemsets L(2): 13
Size of set of large itemsets L(3): 6
Size of set of large itemsets L(4): 1
Best rules found:

1. sexo=FEMININO 134655 ==> eleito_vezes= (-inf-0.5] 128504 conf: (0.95) lift: (1.08) lev: (0.02) [10011] conv:(2.63)
2. idade= (28.5-46.5] resultado=não eleito 188628 ==> eleito_vezes= (-inf-0.5] 178294 conf: (0.95) lift: (1.07) lev: (0.03) [12307] conv:(2.19)
3. resultado=não eleito 364686 ==> eleito_vezes= (-inf-0.5] 340969 conf: (0.93) lift: (1.06) lev: (0.05) [20056] conv:(1.85)
4. estadocivil= CASADO(A) resultado=não eleito 204949 ==> eleito_vezes= (-inf-0.5] 187631 conf: (0.92) lift: (1.04) lev: (0.02) [7282] conv:(1.42)
5. sexo= MASCULINO resultado=não eleito 237952 ==> eleito_vezes= (-inf-0.5] 217425 conf: (0.91) lift: (1.04) lev: (0.02) [8034] conv:(1.39)
6. eleito_vezes= (-inf-0.5] 373367 ==> resultado=não eleito 340969 conf: (0.91) lift: (1.06) lev: (0.05) [20056] conv:(1.62)
7. estadocivil= CASADO(A) eleito_vezes= (-inf-0.5] 207986 ==> resultado=não eleito 187631 conf: (0.9) lift: (1.05) lev: (0.02) [8865] conv:(1.44)
8. idade= (28.5-46.5] eleito_vezes= (-inf-0.5] 198230 ==> resultado=não eleito 178294 conf: (0.9) lift: (1.05) lev: (0.02) [7913] conv:(1.4)
9. sexo= MASCULINO estadocivil= CASADO(A) resultado=não eleito 144652 ==> eleito_vezes= (-inf-0.5] 129351 conf: (0.89) lift: (1.02) lev: (0) [2061] conv:(1.13)
10. sexo= MASCULINO eleito_vezes= (-inf-0.5] 244863 ==> resultado=não eleito 217425 conf: (0.89) lift: (1.03) lev: (0.02) [6963] conv:(1.25)
11. idade= (28.5-46.5] 223466 ==> eleito_vezes= (-inf-0.5] 198230 conf: (0.89) lift: (1.01) lev: (0) [1587] conv:(1.06)
12. grauinstrucao=ENSINO MÉDIO COMPLETO 152968 ==> eleito_vezes= (-inf-0.5] 135679 conf: (0.89) lift: (1.01) lev: (0) [1072] conv:(1.06)
13. sexo= MASCULINO estadocivil= CASADO(A) eleito_vezes= (-inf-0.5] 146597 ==> resultado=não eleito 129351 conf: (0.88) lift: (1.03) lev: (0.01) [3349] conv:(1.19)

14. grauinstrucao=ENSINO MÉDIO COMPLETO 152968 ==> resultado=não eleito 131772 conf: (0.86) lift: (1) lev: (0) [294] conv:(1.01)
15. idade= (28.5-46.5] sexo= MASCULINO 154772 ==> eleito_vezes= (-inf-0.5] 132374 conf: (0.86) lift: (0.97) lev: (-0.01) [-3820] conv:(0.83)
16. estadocivil= CASADO(A) 245260 ==> eleito_vezes= (-inf-0.5] 207986 conf: (0.85) lift: (0.96) lev: (-0.02) [-7835] conv:(0.79)
17. sexo= MASCULINO 289641 ==> eleito_vezes= (-inf-0.5] 244863 conf: (0.85) lift: (0.96) lev: (-0.02) [-10011] conv:(0.78)
18. idade= (28.5-46.5] 223466 ==> resultado=não eleito 188628 conf: (0.84) lift: (0.98) lev: (-0.01) [-3442] conv:(0.9)
19. estadocivil= CASADO(A) 245260 ==> resultado=não eleito 204949 conf: (0.84) lift: (0.97) lev: (-0.01) [-5854] conv:(0.85)
20. sexo= MASCULINO 289641 ==> resultado=não eleito 237952 conf: (0.82) lift: (0.96) lev: (-0.03) [-10996] conv:(0.79)
21. sexo= MASCULINO estadocivil= CASADO(A) 179905 ==> eleito_vezes= (-inf-0.5] 146597 conf: (0.81) lift: (0.93) lev: (-0.03) [-11713] conv:(0.65)
22. sexo= MASCULINO estadocivil= CASADO(A) 179905 ==> resultado=não eleito 144652 conf: (0.8) lift: (0.94) lev: (-0.02) [-9977] conv:(0.72)
23. idade= (28.5-46.5] 223466 ==> eleito_vezes= (-inf-0.5] resultado=não eleito 178294 conf: (0.8) lift: (0.99) lev: (0) [-1285] conv:(0.97)
24. estadocivil= CASADO(A) 245260 ==> eleito_vezes= (-inf-0.5] resultado=não eleito 187631 conf: (0.77) lift: (0.95) lev: (-0.02) [-9462] conv:(0.84)
25. sexo= MASCULINO 289641 ==> eleito_vezes= (-inf-0.5] resultado=não eleito 217425 conf: (0.75) lift: (0.93) lev: (-0.04) [-15333] conv:(0.79)
26. estadocivil= CASADO(A) 245260 ==> sexo= MASCULINO 179905 conf: (0.73) lift: (1.07) lev: (0.03) [12480] conv:(1.19)
27. sexo= MASCULINO estadocivil= CASADO(A) 179905 ==> eleito_vezes= (-inf-0.5] resultado=não eleito 129351 conf: (0.72) lift: (0.89) lev: (-0.04) [-15222] conv:(0.7)
28. estadocivil= CASADO(A) resultado=não eleito 204949 ==> sexo= MASCULINO 144652 conf: (0.71) lift: (1.03) lev: (0.01) [4745] conv:(1.08)
29. estadocivil= CASADO(A) eleito_vezes= (-inf-0.5] 207986 ==> sexo= MASCULINO 146597 conf: (0.7) lift: (1.03) lev: (0.01) [4617] conv:(1.08)

B.1.3 ELEITOS RESTRITO

=== Run information ===

Scheme: weka.associations.Apriori
-N 100 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1 Relation: QueryResult
weka.filters.unsupervised.attribute.Remove-R71
weka.filters.unsupervised.instance.NonSparseToSparse
weka.filters.unsupervised.attribute.RemoveUseless-M99.0
weka.filters.unsupervised.attribute.Remove-R2,4
weka.filters.unsupervised.attribute.Remove-R10
weka.filters.unsupervised.attribute.Remove-R10-11
weka.filters.unsupervised.attribute.NumericToNominal-R10
weka.filters.unsupervised.attribute.Remove-R55-58
weka.filters.unsupervised.attribute.Remove-R10-21
weka.filters.unsupervised.attribute.Remove-R31-42
weka.filters.unsupervised.attribute.Remove-R10-30
weka.filters.unsupervised.attribute.Remove-R2
weka.filters.supervised.attribute.Discretize-Rfirst-last
weka.filters.unsupervised.attribute.Remove-R9
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L1
weka.filters.unsupervised.attribute.Remove-R11
weka.filters.unsupervised.attribute.Remove-R9
Instances: 59610

Attributes: 9

uf, partido_sigla, despesa_maxima_campanha, ocupacao, idade, sexo, grauinstrucao, estadocivil, eleito_vezes

=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.1 (5961 instances)

Minimum metric <confidence>: 0.9
Number of cycles performed: 18
Generated sets of large itemsets:
Size of set of large itemsets L(1): 23

Size of set of large itemsets L(2): 39
Size of set of large itemsets L(3): 22
Size of set of large itemsets L(4): 5
Best rules found:

1. grauinstrucao=ENSINO FUNDAMENTAL INCOMPLETO estadocivil= CASADO(A) 6930 ==> sexo= MASCULINO 6604 conf: (0.95) lift: (1.1) lev: (0.01) [594] conv:(2.82)
2. grauinstrucao=ENSINO FUNDAMENTAL INCOMPLETO 9446 ==> sexo= MASCULINO 8984 conf: (0.95) lift: (1.1) lev: (0.01) [793] conv:(2.71)
3. ocupacao=AGRICULTOR 6604 ==> sexo= MASCULINO 6209 conf: (0.94) lift: (1.08) lev: (0.01) [482] conv:(2.22)
4. grauinstrucao=ENSINO FUNDAMENTAL COMPLETO 8227 ==> sexo= MASCULINO 7650 conf: (0.93) lift: (1.07) lev: (0.01) [516] conv:(1.89)
5. eleito_vezes='(1.5-inf)' 6896 ==> sexo= MASCULINO 6265 conf: (0.91) lift: (1.05) lev: (0) [285] conv:(1.45)
6. ocupacao=VEREADOR idade='(28.5-46.5]' 6627 ==> sexo= MASCULINO 5971 conf: (0.9) lift: (1.04) lev: (0) [224] conv:(1.34)

B.1.4 ELEITOS ABRANGENTE

=== Run information ===

Scheme: weka.associations.Apriori
-N 100 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.3 -S -1.0 -c -1 Relation: QueryResult
weka.filters.unsupervised.attribute.Remove-R71
weka.filters.unsupervised.instance.NonSparseToSparse
weka.filters.unsupervised.attribute.RemoveUseless-M99.0
weka.filters.unsupervised.attribute.Remove-R2,4
weka.filters.unsupervised.attribute.Remove-R10
weka.filters.unsupervised.attribute.Remove-R10-11
weka.filters.unsupervised.attribute.NumericToNominal-R10
weka.filters.unsupervised.attribute.Remove-R55-58
weka.filters.unsupervised.attribute.Remove-R10-21
weka.filters.unsupervised.attribute.Remove-R31-42
weka.filters.unsupervised.attribute.Remove-R10-30
weka.filters.unsupervised.attribute.Remove-R2

weka.filters.supervised.attribute.Discretize-Rfirst-last
weka.filters.unsupervised.attribute.Remove-R9
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L1
weka.filters.unsupervised.attribute.Remove-R11
weka.filters.unsupervised.attribute.Remove-R9
Instances: 59610

Attributes: 9

uf, partido_sigla, despesa_maxima_campanha, ocupacao, idade, sexo, grauinstrucao, estadocivil, eleito_vezes

=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.3 (17883 instances)

Minimum metric <confidence>: 0.7
Number of cycles performed: 14
Generated sets of large itemsets:
Size of set of large itemsets L(1): 6
Size of set of large itemsets L(2): 8
Size of set of large itemsets L(3): 1
Best rules found:

1. eleito_vezes= (0.5-1.5] 20316 ==> sexo= MASCULINO 17986 conf: (0.89) lift: (1.02) lev: (0.01) [369] conv:(1.16)
2. estadocivil= CASADO(A) 40311 ==> sexo= MASCULINO 35253 conf: (0.87) lift: (1.01) lev: (0.01) [298] conv:(1.06)
3. grauinstrucao=ENSINO MÉDIO COMPLETO 21196 ==> sexo= MASCULINO 18515 conf: (0.87) lift: (1.01) lev: (0) [135] conv:(1.05)
4. idade= (28.5-46.5] estadocivil= CASADO(A) 23554 ==> sexo= MASCULINO 20510 conf: (0.87) lift: (1) lev: (0) [85] conv:(1.03)
5. idade= (28.5-46.5] 34838 ==> sexo= MASCULINO 30243 conf: (0.87) lift: (1) lev: (0) [34] conv:(1.01)

6. eleito_vezes= (-inf-0.5] 32398 ==> sexo= MASCULINO 27438 conf: (0.85) lift: (0.98) lev: (-0.01) [-654] conv:(0.87)

B.1.5 ELEITOS MULHERES

=== Run information ===

Scheme: weka.associations.Apriori
-N 100 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.3 -S -1.0 -c -1 Relation: QueryResult
weka.filters.unsupervised.attribute.Remove-R71
weka.filters.unsupervised.instance.NonSparseToSparse
weka.filters.unsupervised.attribute.RemoveUseless-M99.0
weka.filters.unsupervised.attribute.Remove-R2,4
weka.filters.unsupervised.attribute.Remove-R10
weka.filters.unsupervised.attribute.Remove-R10-11
weka.filters.unsupervised.attribute.NumericToNominal-R10
weka.filters.unsupervised.attribute.Remove-R55-58
weka.filters.unsupervised.attribute.Remove-R10-21
weka.filters.unsupervised.attribute.Remove-R31-42
weka.filters.unsupervised.attribute.Remove-R10-30
weka.filters.unsupervised.attribute.Remove-R2
weka.filters.supervised.attribute.Discretize-Rfirst-last
weka.filters.unsupervised.attribute.Remove-R9
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L1
weka.filters.unsupervised.attribute.Remove-R11
weka.filters.unsupervised.attribute.Remove-R9
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C6-L1
weka.filters.unsupervised.attribute.Remove-R6
Instances: 7921 Attributes: 8 uf, partido_sigla, despesa_maxima_campanha, ocupacao, idade, grauinstrucao, estadocivil, eleito_vezes === Associator model (full training set) ===

Apriori
=======

Minimum support: 0.3 (2376 instances)

Minimum metric <confidence>: 0.5
Number of cycles performed: 14
Generated sets of large itemsets:
Size of set of large itemsets L(1): 5
Size of set of large itemsets L(2): 3
Best rules found:

1. idade= (28.5-46.5] 4595 ==> eleito_vezes= (-inf-0.5] 3106 conf: (0.68) lift: (1.08) lev: (0.03) [228] conv:(1.15)
2. idade= (28.5-46.5] 4595 ==> estadocivil= CASADO(A) 3044 conf: (0.66) lift: (1.04) lev: (0.01) [109] conv:(1.07)
3. eleito_vezes= (-inf-0.5] 4960 ==> estadocivil= CASADO(A) 3109 conf: (0.63) lift: (0.98) lev: (-0.01) [-58] conv:(0.97)
4. eleito_vezes= (-inf-0.5] 4960 ==> idade= (28.5-46.5] 3106 conf: (0.63) lift: (1.08) lev: (0.03) [228] conv:(1.12)
5. estadocivil= CASADO(A) 5058 ==> eleito_vezes= (-inf-0.5] 3109 conf: (0.61) lift: (0.98) lev: (-0.01) [-58] conv:(0.97)
6. estadocivil= CASADO(A) 5058 ==> idade= (28.5-46.5] 3044 conf: (0.6) lift: (1.04) lev: (0.01) [109] conv:(1.05)

B.2 PREFEITOS

B.2.1 TODOS RESTRITO

=== Run information ===

Scheme: weka.associations.Apriori
-N 100 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1 Relation: QueryResult
weka.filters.unsupervised.attribute.Remove-R71
weka.filters.unsupervised.attribute.Remove-R1-2,4-5
weka.filters.unsupervised.attribute.Remove-R2-3,5,12-65
weka.filters.unsupervised.attribute.NumericToNominal-R10
weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-R5
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R9

Instances: 16010

Attributes: 11

uf, partido_sigla, despesa_maxima_campanha, ocupacao, idade, sexo, grauinstrucao, estadocivil, votacao_comparecimentos, eleito_vezes, resultado

=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.1 (1601 instances)

Minimum metric <confidence>: 0.9
Number of cycles performed: 18
Generated sets of large itemsets:
Size of set of large itemsets L(1): 25
Size of set of large itemsets L(2): 52
Size of set of large itemsets L(3): 45
Size of set of large itemsets L(4): 15
Size of set of large itemsets L(5): 1
Best rules found:

1. ocupacao=EMPRESÁRIO 1868 ==> sexo= MASCULINO 1730 conf: (0.93) lift: (1.07) lev: (0.01) [113] conv:(1.81)
2. estadocivil= CASADO(A) eleito_vezes=1 resultado=eleito 1923 ==> sexo= MASCULINO 1762 conf: (0.92) lift: (1.06) lev: (0.01) [97] conv:(1.6)
3. grauinstrucao=ENSINO MÉDIO COMPLETO estadocivil= CASADO(A) 3159 ==> sexo= MASCULINO 2871 conf: (0.91) lift: (1.05) lev: (0.01) [137] conv:(1.47)
4. idade= (53.4-62] estadocivil= CASADO(A) 2281 ==> sexo= MASCULINO 2069 conf: (0.91) lift: (1.05) lev: (0.01) [94] conv:(1.44)
5. eleito_vezes=1 resultado=eleito 2476 ==> sexo= MASCULINO 2242 conf: (0.91) lift: (1.05) lev: (0.01) [99] conv:(1.42)

B.2.2 TODOS ABRANGENTE

=== Run information ===

Scheme: weka.associations.Apriori -N 100 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.3 -S -1.0 -c -1
Relation: QueryResult
weka.filters.unsupervised.attribute.Remove-R71
weka.filters.unsupervised.attribute.Remove-R1-2,4-5
weka.filters.unsupervised.attribute.Remove-R2-3,5,12-65
weka.filters.unsupervised.attribute.NumericToNominal-R10
weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-R5
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R9
Instances: 16010 Attributes: 11 uf, partido_sigla, despesa_maxima_campanha, ocupacao, idade, sexo, grauinstrucao, estadocivil, votacao_comparecimentos, eleito_vezes, resultado === Associator model (full training set) ===

Apriori
=======

Minimum support: 0.3 (4803 instances)

Minimum metric <confidence>: 0.7
Number of cycles performed: 14
Generated sets of large itemsets:
Size of set of large itemsets L(1): 8
Size of set of large itemsets L(2): 11
Size of set of large itemsets L(3): 4
Best rules found:
1. eleito_vezes=1 5761 ==> sexo= MASCULINO 5105 conf: (0.89) lift: (1.02) lev: (0.01) [119] conv:(1.18)
2. resultado=eleito 5771 ==> sexo= MASCULINO 5086 conf: (0.88) lift: (1.02) lev: (0.01) [91] conv:(1.13)
3. estadocivil= CASADO(A) 11694 ==> sexo= MASCULINO 10298 conf: (0.88) lift: (1.02) lev: (0.01) [177] conv:(1.13)
4. estadocivil= CASADO(A) resultado=não eleito 7267 ==> sexo= MASCULINO 6334 conf: (0.87) lift: (1.01) lev: (0) [44] conv:(1.05)

5. estadocivil= CASADO(A) eleito_vezes=0 6442 ==> sexo= MASCULINO 5560 conf: (0.86) lift: (1) lev: (0) [-15] conv:(0.98)
6. resultado=não eleito 10239 ==> sexo= MASCULINO 8770 conf: (0.86) lift: (0.99) lev: (-0.01) [-91] conv:(0.94)
7. eleito_vezes=0 9113 ==> sexo= MASCULINO 7729 conf: (0.85) lift: (0.98) lev: (-0.01) [-157] conv:(0.89)
8. eleito_vezes=0 resultado=não eleito 6231 ==> sexo= MASCULINO 5267 conf: (0.85) lift: (0.98) lev: (-0.01) [-125] conv:(0.87)
9. grauinstrucao=SUPERIOR COMPLETO estadocivil= CASADO(A) 5743 ==> sexo= MASCULINO 4822 conf: (0.84) lift: (0.97) lev: (-0.01) [-148] conv:(0.84)
10. grauinstrucao=SUPERIOR COMPLETO 7928 ==> sexo= MASCULINO 6525 conf: (0.82) lift: (0.95) lev: (-0.02) [-336] conv:(0.76)
11. sexo= MASCULINO 13856 ==> estadocivil= CASADO(A) 10298 conf: (0.74) lift: (1.02) lev: (0.01) [177] conv:(1.05)
12. sexo= MASCULINO grauinstrucao=SUPERIOR COMPLETO 6525 ==> estadocivil= CASADO(A) 4822 conf: (0.74) lift: (1.01) lev: (0) [56] conv:(1.03)
13. grauinstrucao=SUPERIOR COMPLETO 7928 ==> estadocivil= CASADO(A) 5743 conf: (0.72) lift: (0.99) lev: (0) [-47] conv:(0.98)
14. sexo= MASCULINO resultado=não eleito 8770 ==> estadocivil= CASADO(A) 6334 conf: (0.72) lift: (0.99) lev: (0) [-71] conv:(0.97)
15. sexo= MASCULINO eleito_vezes=0 7729 ==> estadocivil= CASADO(A) 5560 conf: (0.72) lift: (0.98) lev: (-0.01) [-85] conv:(0.96)
16. resultado=não eleito 10239 ==> estadocivil= CASADO(A) 7267 conf: (0.71) lift: (0.97) lev: (-0.01) [-211] conv:(0.93)
17. eleito_vezes=0 9113 ==> estadocivil= CASADO(A) 6442 conf: (0.71) lift: (0.97) lev: (-0.01) [-214] conv:(0.92)

B.2.3 ELEITOS RESTRITO

=== Run information ===

Scheme: weka.associations.Apriori N 50 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.01 -S -1.0 -c -1

Relation: QueryResult
weka.filters.unsupervised.attribute.Remove-R71

weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L2
weka.filters.unsupervised.attribute.Remove-R72
weka.filters.unsupervised.attribute.Remove-R1-2,4
weka.filters.unsupervised.attribute.Remove-R2,4,6
weka.filters.unsupervised.attribute.Remove-R10
weka.filters.unsupervised.attribute.Remove-R11
weka.filters.unsupervised.attribute.Remove-R21-60
weka.filters.unsupervised.attribute.Remove-R10-20
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R10
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R11
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R11
weka.filters.unsupervised.attribute.Discretize-O-B10-M-1.0-R6
weka.filters.unsupervised.attribute.NumericToNominal-R12
weka.filters.unsupervised.attribute.Remove-R10

Instances: 5771

Attributes: 11

uf, ue, partido_sigla, despesa_maxima_campanha, ocupacao, idade, sexo, grauinstrucao, estadocivil, votacao_comparecimentos, eleito_vezes

=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.06 (346 instances)

Minimum metric <confidence>: 0.9
Number of cycles performed: 19
Generated sets of large itemsets:
Size of set of large itemsets L(1): 52
Size of set of large itemsets L(2): 147
Size of set of large itemsets L(3): 110
Size of set of large itemsets L(4): 22
Best rules found:

1. ocupacao=AGRICULTOR estadocivil= CASADO(A) 297 ==> sexo= MASCULINO 294 conf: (0.99) lift: (1.12) lev: (0.01) [32] conv:(8.81)
2. ocupacao=AGRICULTOR 377 ==> sexo= MASCULINO 370 conf: (0.98) lift: (1.11) lev: (0.01) [37] conv:(5.59)
3. grauinstrucao=ENSINO FUNDAMENTAL INCOMPLETO estadocivil= CASADO(A) 328 ==> sexo= MASCULINO 316 conf: (0.96) lift: (1.09) lev: (0) [26] conv:(2.99)
4. grauinstrucao=ENSINO FUNDAMENTAL INCOMPLETO 414 ==> sexo= MASCULINO 395 conf: (0.95) lift: (1.08) lev: (0.01) [30] conv:(2.46)
5. grauinstrucao=ENSINO MÉDIO COMPLETO estadocivil= CASADO(A) eleito_vezes=1 549 ==> sexo= MASCULINO 522 conf: (0.95) lift: (1.08) lev: (0.01) [38] conv:(2.33)
6. grauinstrucao=ENSINO MÉDIO COMPLETO eleito_vezes=1 695 ==> sexo= MASCULINO 657 conf: (0.95) lift: (1.07) lev: (0.01) [44] conv:(2.12)
7. uf=RS estadocivil= CASADO(A) 407 ==> sexo= MASCULINO 384 conf: (0.94) lift: (1.07) lev: (0) [25] conv:(2.01)
8. uf=MG estadocivil= CASADO(A) 682 ==> sexo= MASCULINO 641 conf: (0.94) lift: (1.07) lev: (0.01) [39] conv:(1.93)
9. idade= (59.8-66.6] estadocivil= CASADO(A) 310 ==> sexo= MASCULINO 291 conf: (0.94) lift: (1.07) lev: (0) [17] conv:(1.84)
10. uf=MG eleito_vezes=1 355 ==> sexo= MASCULINO 333 conf: (0.94) lift: (1.06) lev: (0) [20] conv:(1.83)
11. ocupacao=EMPRESÁRIO estadocivil= CASADO(A) 528 ==> sexo= MASCULINO 495 conf: (0.94) lift: (1.06) lev: (0.01) [29] conv:(1.84)
12. ocupacao=EMPRESÁRIO estadocivil= CASADO(A) eleito_vezes=0 373 ==> sexo= MASCULINO 349 conf: (0.94) lift: (1.06) lev: (0) [20] conv:(1.77)
13. uf=RS 519 ==> sexo= MASCULINO 484 conf: (0.93) lift: (1.06) lev: (0) [26] conv:(1.71)
14. idade= (46.2-53] grauinstrucao=ENSINO MÉDIO COMPLETO 394 ==> sexo= MASCULINO 367 conf: (0.93) lift: (1.06) lev: (0) [19] conv:(1.67)
15. ocupacao=EMPRESÁRIO 687 ==> sexo= MASCULINO 639 conf: (0.93) lift: (1.06) lev: (0.01) [33] conv:(1.66)
16. idade= (46.2-53] grauinstrucao=ENSINO MÉDIO COMPLETO estadocivil= CASADO(A) 324 ==> sexo= MASCULINO 301 conf: (0.93) lift: (1.05) lev: (0) [15] conv:(1.6)
17. ocupacao=COMERCIANTE 364 ==> sexo= MASCULINO 338 conf: (0.93) lift: (1.05) lev: (0) [17] conv:(1.6)
18. idade= (39.4-46.2] grauinstrucao=ENSINO MÉDIO COMPLETO estadocivil= CASADO(A) 418 ==> sexo= MASCULINO 388 conf: (0.93) lift: (1.05) lev: (0) [19] conv:(1.6)

19. grauinstrucao=ENSINO MÉDIO COMPLETO estadocivil= CASADO(A) 1251 ==> sexo= MASCULINO 1160 conf: (0.93) lift: (1.05) lev: (0.01) [57] conv:(1.61)
20. uf=MG estadocivil= CASADO(A) eleito_vezes=0 355 ==> sexo= MASCULINO 329 conf: (0.93) lift: (1.05) lev: (0) [16] conv:(1.56)
21. partido_sigla=PP estadocivil= CASADO(A) 380 ==> sexo= MASCULINO 352 conf: (0.93) lift: (1.05) lev: (0) [17] conv:(1.56)
22. estadocivil= CASADO(A) votacao_comparecimentos= (58923-inf) 459 ==> sexo= MASCULINO 425 conf: (0.93) lift: (1.05) lev: (0) [20] conv:(1.56)
23. estadocivil= CASADO(A) votacao_comparecimentos= (7173-9352.5] 457 ==> sexo= MASCULINO 423 conf: (0.93) lift: (1.05) lev: (0) [20] conv:
(1.55)
24. grauinstrucao=ENSINO FUNDAMENTAL COMPLETO 372 ==> sexo= MASCULINO 344 conf: (0.92) lift: (1.05) lev: (0) [16] conv:(1.52)
25. uf=SP eleito_vezes=1 314 ==> sexo= MASCULINO 290 conf: (0.92) lift: (1.05) lev: (0) [13] conv:(1.49)
26. idade= (39.4-46.2] grauinstrucao=ENSINO MÉDIO COMPLETO 511 ==> sexo= MASCULINO 471 conf: (0.92) lift: (1.05) lev: (0) [20] conv:(1.48)
27. grauinstrucao=ENSINO MÉDIO COMPLETO 1585 ==> sexo= MASCULINO 1460 conf: (0.92) lift: (1.05) lev: (0.01) [63] conv:(1.49)
28. partido_sigla=PDT 323 ==> sexo= MASCULINO 297 conf: (0.92) lift: (1.04) lev: (0) [12] conv:(1.42)
29. uf=MG 882 ==> sexo= MASCULINO 811 conf: (0.92) lift: (1.04) lev: (0.01) [33] conv:(1.45)
30. eleito_vezes=2 380 ==> sexo= MASCULINO 349 conf: (0.92) lift: (1.04) lev: (0) [14] conv:(1.41)
31. ocupacao=EMPRESÁRIO eleito_vezes=0 478 ==> sexo= MASCULINO 439 conf: (0.92) lift: (1.04) lev: (0) [17] conv:(1.42)
32. idade= (53-59.8] estadocivil= CASADO(A) 603 ==> sexo= MASCULINO 553 conf: (0.92) lift: (1.04) lev: (0) [21] conv:(1.4)
33. idade= (53-59.8] eleito_vezes=1 325 ==> sexo= MASCULINO 298 conf: (0.92) lift: (1.04) lev: (0) [11] conv:(1.38)
34. estadocivil= CASADO(A) eleito_vezes=1 1923 ==> sexo= MASCULINO 1762 conf: (0.92) lift: (1.04) lev: (0.01) [67] conv:(1.41)
35. idade= (46.2-53] estadocivil= CASADO(A) eleito_vezes=1 534 ==> sexo= MASCULINO 488 conf: (0.91) lift: (1.04) lev: (0) [17] conv:(1.35)

36. estadocivil= CASADO(A) votacao_comparecimentos= (31627-58923] 429 ==> sexo= MASCULINO 392 conf: (0.91) lift: (1.04) lev: (0) [13] conv:(1.34)
37. idade= (46.2-53] eleito_vezes=1 652 ==> sexo= MASCULINO 593 conf: (0.91) lift: (1.03) lev: (0) [18] conv:(1.29)
38. idade= (59.8-66.6] 394 ==> sexo= MASCULINO 358 conf: (0.91) lift: (1.03) lev: (0) [10] conv:(1.26)
39. votacao_comparecimentos= (7173-9352.5] 577 ==> sexo= MASCULINO 524 conf: (0.91) lift: (1.03) lev: (0) [15] conv:(1.27)
40. despesa_maxima_campanha=300000 estadocivil= CASADO(A) 478 ==> sexo= MASCULINO 434 conf: (0.91) lift: (1.03) lev: (0) [12] conv:(1.26)
41. idade= (39.4-46.2] estadocivil= CASADO(A) eleito_vezes=1 572 ==> sexo= MASCULINO 519 conf: (0.91) lift: (1.03) lev: (0) [14] conv:(1.26)
42. grauinstrucao=SUPERIOR COMPLETO estadocivil= CASADO(A) votacao_comparecimentos= (58923-inf) 345 ==> sexo= MASCULINO 313 conf: (0.91) lift: (1.03) lev: (0) [8] conv:(1.24)
43. uf=MG eleito_vezes=0 468 ==> sexo= MASCULINO 424 conf: (0.91) lift: (1.03) lev: (0) [11] conv:(1.23)
44. eleito_vezes=1 2476 ==> sexo= MASCULINO 2242 conf: (0.91) lift: (1.03) lev: (0.01) [59] conv:(1.25)
45. idade= (53-59.8] 735 ==> sexo= MASCULINO 665 conf: (0.9) lift: (1.03) lev: (0) [17] conv:(1.23)
46. votacao_comparecimentos= (58923-inf) 577 ==> sexo= MASCULINO 522 conf: (0.9) lift: (1.03) lev: (0) [13] conv:(1.22)
47. partido_sigla=PMDB estadocivil= CASADO(A) eleito_vezes=1 386 ==> sexo= MASCULINO 349 conf: (0.9) lift: (1.03) lev: (0) [8] conv:(1.21)
48. grauinstrucao=ENSINO MÉDIO COMPLETO estadocivil= CASADO(A) eleito_vezes=0 600 ==> sexo= MASCULINO 542 conf: (0.9) lift: (1.02) lev: (0) [13] conv:(1.21)
49. uf=SP estadocivil= CASADO(A) 537 ==> sexo= MASCULINO 485 conf: (0.9) lift: (1.02) lev: (0) [11] conv:(1.2)
50. partido_sigla=PT estadocivil= CASADO(A) 492 ==> sexo= MASCULINO 444 conf: (0.9) lift: (1.02) lev: (0) [10] conv:(1.19)

B.2.4 ELEITOS ABRANGENTE

=== Run information ===

Scheme: weka.associations.Apriori -N 100 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.3 -S -1.0 -c 7

Relation: QueryResult
weka.filters.unsupervised.attribute.Remove-R71
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L2
weka.filters.unsupervised.attribute.Remove-R72
weka.filters.unsupervised.attribute.Remove-R1-2,4
weka.filters.unsupervised.attribute.Remove-R2,4,6
weka.filters.unsupervised.attribute.Remove-R10
weka.filters.unsupervised.attribute.Remove-R11
weka.filters.unsupervised.attribute.Remove-R21-60
weka.filters.unsupervised.attribute.Remove-R10-20
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R10
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R11
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R11
weka.filters.unsupervised.attribute.Discretize-O-B10-M-1.0-R6
weka.filters.unsupervised.attribute.NumericToNominal-R12
weka.filters.unsupervised.attribute.Remove-R10
Instances: 5771

Attributes: 11

uf, ue, partido_sigla, despesa_maxima_campanha, ocupacao, idade, sexo, grauinstrucao,vestadocivil, votacao_comparecimentos,veleito_vezes

=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.3 (1731 instances)

Minimum metric <confidence>: 0.7 Number of cycles performed: 14 Generated sets of large itemsets:
Size of set of large itemsets L(1): 5
Size of set of large itemsets L(2): 7
Size of set of large itemsets L(3): 3
Best rules found:
1. estadocivil= CASADO(A) eleito_vezes=1 1923 ==> sexo= MASCULINO 1762 conf: (0.92)

lift: (1.04) lev: (0.01) [67] conv:(1.41)
2. eleito_vezes=1 2476 ==> sexo= MASCULINO 2242 conf: (0.91) lift: (1.03) lev: (0.01) [59] conv:(1.25)
3. estadocivil= CASADO(A) 4427 ==> sexo= MASCULINO 3964 conf: (0.9) lift: (1.02) lev: (0.01) [62] conv:(1.13)
4. estadocivil= CASADO(A) eleito_vezes=0 2176 ==> sexo= MASCULINO 1891 conf: (0.87) lift: (0.99) lev: (0) [-26] conv:(0.9)
5. eleito_vezes=0 2882 ==> sexo= MASCULINO 2462 conf: (0.85) lift: (0.97) lev: (-0.01) [-77] conv:(0.81)
6. grauinstrucao=SUPERIOR COMPLETO estadocivil= CASADO(A) 2139 ==> sexo= MASCULINO 1827 conf: (0.85) lift: (0.97) lev: (-0.01) [-58] conv:(0.81)
7. grauinstrucao=SUPERIOR COMPLETO 2837 ==> sexo= MASCULINO 2377 conf: (0.84) lift: (0.95) lev: (-0.02) [-123] conv:(0.73)
8. sexo= MASCULINO eleito_vezes=1 2242 ==> estadocivil= CASADO(A) 1762 conf: (0.79) lift: (1.02) lev: (0.01) [42] conv:(1.09)
9. sexo= MASCULINO 5086 ==> estadocivil= CASADO(A) 3964 conf: (0.78) lift: (1.02) lev: (0.01) [62] conv:(1.05)
10. eleito_vezes=1 2476 ==> estadocivil= CASADO(A) 1923 conf: (0.78) lift: (1.01) lev: (0) [23] conv:(1.04)
11. sexo= MASCULINO grauinstrucao=SUPERIOR COMPLETO 2377 ==> estadocivil= CASADO(A) 1827 conf: (0.77) lift: (1) lev: (0) [3] conv:(1)
12. sexo= MASCULINO eleito_vezes=0 2462 ==> estadocivil= CASADO(A) 1891 conf: (0.77) lift: (1) lev: (0) [2] conv:(1)
13. eleito_vezes=0 2882 ==> estadocivil= CASADO(A) 2176 conf: (0.76) lift: (0.98) lev: (-0.01) [-34] conv:(0.95)
14. grauinstrucao=SUPERIOR COMPLETO 2837 ==> estadocivil= CASADO(A) 2139 conf: (0.75) lift: (0.98) lev: (-0.01) [-37] conv:(0.95)
15. eleito_vezes=1 2476 ==> sexo= MASCULINO estadocivil= CASADO(A) 1762 conf: (0.71) lift: (1.04) lev: (0.01) [61] conv:(1.08)

B.2.5 ELEITOS MULHERES

=== Run information ===

Scheme: weka.associations.Apriori -N 100 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c 7
Relation: QueryResult

weka.filters.unsupervised.attribute.Remove-R71
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L2
weka.filters.unsupervised.attribute.Remove-R72
weka.filters.unsupervised.attribute.Remove-R1-2,4
weka.filters.unsupervised.attribute.Remove-R2,4,6
weka.filters.unsupervised.attribute.Remove-R10
weka.filters.unsupervised.attribute.Remove-R11
weka.filters.unsupervised.attribute.Remove-R21-60
weka.filters.unsupervised.attribute.Remove-R10-20
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R10
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R11
weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-R11
weka.filters.unsupervised.attribute.Discretize-O-B10-M-1.0-R6
weka.filters.unsupervised.attribute.NumericToNominal-R12
weka.filters.unsupervised.attribute.Remove-R10
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C7-L2-V
weka.filters.unsupervised.attribute.Remove-R7
Instances: 685

Attributes: 10

uf, ue, partido_sigla, despesa_maxima_campanha, ocupacao, idade, grauinstrucao, estadocivil, votacao_comparecimentos, eleito_vezes

=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.1 (69 instances)

Minimum metric <confidence>: 0.7
Number of cycles performed: 18
Generated sets of large itemsets:
Size of set of large itemsets L(1): 23
Size of set of large itemsets L(2): 25
Size of set of large itemsets L(3): 8

Best rules found:

1. ocupacao=PREFEITO estadocivil= CASADO(A) 86 ==> eleito_vezes=1 74 conf: (0.86) lift: (2.52) lev: (0.07) [44] conv:(4.36)

2. ocupacao=PREFEITO 121 ==> eleito_vezes=1 101 conf: (0.83) lift: (2.44) lev: (0.09) [59] conv:(3.79)

3. idade= (39.4-46.2] eleito_vezes=0 104 ==> estadocivil= CASADO(A) 79 conf: (0.76) lift: (1.12) lev: (0.01) [8] conv:(1.3)

4. idade= (46.2-53] eleito_vezes=0 106 ==> estadocivil= CASADO(A) 80 conf: (0.75) lift: (1.12) lev: (0.01) [8] conv:(1.27)

5. idade= (46.2-53] 171 ==> estadocivil= CASADO(A) 129 conf: (0.75) lift: (1.12) lev: (0.02) [13] conv:(1.29)

6. idade= (46.2-53] grauinstrucao=SUPERIOR COMPLETO 122 ==> estadocivil= CASADO(A) 90 conf: (0.74) lift: (1.09) lev: (0.01) [7] conv:(1.2)

7. ocupacao=PREFEITO eleito_vezes=1 101 ==> estadocivil= CASADO(A) 74 conf: (0.73) lift: (1.08) lev: (0.01) [5] conv:(1.17)

8. partido_sigla=PMDB 130 ==> estadocivil= CASADO(A) 95 conf: (0.73) lift: (1.08) lev: (0.01) [7] conv:(1.17)

9. grauinstrucao=ENSINO MÉDIO COMPLETO 125 ==> estadocivil= CASADO(A) 91 conf: (0.73) lift: (1.08) lev: (0.01) [6] conv:(1.16)

10. idade= (46.2-53] 171 ==> grauinstrucao=SUPERIOR COMPLETO 122 conf: (0.71) lift: (1.06) lev: (0.01) [7] conv:(1.12)

11. ocupacao=PREFEITO 121 ==> estadocivil= CASADO(A) 86 conf: (0.71) lift: (1.05) lev: (0.01) [4] conv:(1.09)

12. idade= (39.4-46.2] 190 ==> estadocivil= CASADO(A) 135 conf: (0.71) lift: (1.05) lev: (0.01) [6] conv:(1.1)

13. idade= (39.4-46.2] grauinstrucao=SUPERIOR COMPLETO 126 ==> estadocivil= CASADO(A) 89 conf: (0.71) lift: (1.05) lev: (0.01) [3] conv:(1.07)