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

Aplicação do Algoritmo Apriori para Detectar Relacionamentos entre Empresas nos Processos Licitatórios do Governo Federal

Rebeca Andrade Baldomir

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

Orientadora
Prof.ª Dr.ª Célia Ghedini Ralha

Brasília
2017

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

Coordenador: Prof. Dr. Rodrigo Bonifácio

Banca examinadora composta por:

Prof.ª Dr.ª Célia Ghedini Ralha (Orientadora) — CIC/UnB
Prof. MSc. Gustavo Cordeiro Galvão Van Erven — CIC/UnB
MSc. Carlos Vinícius Sarmento Silva — CIC/UnB

CIP — Catalogação Internacional na Publicação

Baldomir, Rebeca Andrade.

Aplicação do Algoritmo Apriori para Detectar Relacionamentos entre Empresas nos Processos Licitatórios do Governo Federal / Rebeca Andrade Baldomir. Brasília : UnB, 2017.
49 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2017.

1. Licitações, 2. Fraude, 3. Mineração de dados

CDU 004

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

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

Aplicação do Algoritmo Apriori para Detectar Relacionamentos entre Empresas nos Processos Licitatórios do Governo Federal

Rebeca Andrade Baldomir

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

Prof.ª Dr.ª Célia Ghedini Ralha (Orientadora)
CIC/UnB

Prof. MSc. Gustavo Cordeiro Galvão Van Erven
CIC/UnB

MSc. Carlos Vinícius Sarmento Silva
CIC/UnB

Prof. Dr. Rodrigo Bonifácio
Coordenador do Bacharelado em Ciência da Computação

Brasília, 14 de dezembro de 2017

Dedicatória

À Rute,
que sempre apoiou e esteve presente nos momentos de dificuldades
e nunca deixou de comemorar cada vitória comigo.
Obrigada mãe.

Agradecimentos

Agradeço a Deus que me proporcionou todos os momentos vividos até agora. Agradeço a toda minha família e amigos que me acompanharam durante essa jornada e a tornaram mais leve e à professora Célia e ao Gustavo pela orientação, atenção e paciência ao desenvolvermos este trabalho. Sobretudo, agradeço à minha mãe e irmão por todas as palavras de apoio e incentivo não só agora como durante toda a vida. Por último e especialmente importante, agradeço ao Uriel que não só me incentivou como me acompanhou durante essa graduação e me proporcionou momentos inesquecíveis durante esses anos.

Resumo

A mineração de dados tem sido uma área de alta visibilidade nos últimos anos e de muitas pesquisas que mostraram boa eficiência dessa área para encontrar informações em grandes bases de dados. Esse trabalho propõe usar a mineração de dados nas bases digitais de licitações públicas do Governo Federal Brasileiro. O objetivo é encontrar indícios de fraudes, tais como conluios e cartéis. Essa tarefa é complexa para os auditores dado que a quantidade de dados disponíveis é muito grande e dada a dificuldade de correlacionar esses dados. Como resultado desse trabalho espera-se que os auditores possam ter um auxílio na tarefa de auditoria de licitações na Controladoria Geral da União (CGU).

Palavras-chave: Licitações, Fraude, Mineração de dados

Abstract

Data mining has been an area of high visibility in recent years and many researches have shown good efficiency in this area to find information in large databases. This project proposes to use data mining in the digital databases of public bidding of the Brazilian Federal Government. The aim is to find evidence of fraud, such as stunts and cartels. This task is complex for auditors since the amount of data available is very large and given the difficulty of correlating this data. As a result of this work it is expected that the auditors can have an aid in the task of auditing bids in the Controladoria Geral da União (CGU).

Keywords: Biddings, Fraud, Data Mining

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

As licitações públicas são o meio de compra do Governo Federal Brasileiro. As licitações públicas do Poder Executivo Federal podem ser auditadas pela Controladoria-Geral da União (CGU)[1]. As licitações são, frequentemente, alvos de fraudes e para detectá-las é necessário um trabalho complexo de auditoria por se tratar de um grande volume de informações disponíveis nas bases de dados públicas, dificultando o correlacionamento dos mesmos.

A mineração de dados tem se mostrado de grande valia na obtenção de informações e no processo de descoberta de conhecimento (Witten et al., 2011). Dessa maneira, a utilização da mineração de dados é bastante útil na busca de fraudes nas grandes bases de dados de licitação.

Esse trabalho propõe o uso de técnicas de mineração de dados para auxiliar o trabalho de auditoria na CGU, mais especificamente, aplicando regras de associação e, assim, identificando indícios de cartéis e conluio em licitações públicas. Como fonte de dados para a mineração será utilizado o Portal de Compras (ComprasNET)[2]. O ComprasNET é um site Web instituído pelo Ministério do Planejamento, Orçamento e Gestão (MPOG)[3] para disponibilizar à sociedade informações referentes às licitações e contratações promovidas pelo Governo Federal.

A preocupação tratada nesse trabalho surgiu dos auditores e já vem sendo abordada em outros trabalhos. Ralha and Silva (2012) já afirmou que a maior dificuldade se encontra no correlacionamento das informações disponíveis para geração de conhecimento útil para os auditores. Nesse trabalho a verificação da existência de correlacionamento das empresas será através da aplicação das regras de associação.

Conforme exposto, a questão de pesquisa sendo investigada nesse trabalho é: será que o uso de técnicas de mineração de dados no portal ComprasNet pode auxiliar os auditores da CGU na detecção de fraudes, como conluios ou cartéis, utilizando correlação de empresas participantes em processos licitatórios do Governo Federal Brasileiro?

[1] http://www.cgu.gov.br/
[2] http://www.comprasgovernamentais.gov.br/
[3] http://www.planejamento.gov.br/

1.1 Objetivos

Esse trabalho tem como objetivo definir um modelo para correlacionar dados de licitações públicas utilizando o Portal ComprasNET. O modelo deve encontrar relações entre as empresas participantes de licitações usando regras de associação.

Para que seja possível alcançar o objetivo geral descrito, será necessário o cumprimento dos seguintes objetivos secundários:

- O desenvolvimento de uma ferramenta que implemente o modelo de mineração de dados definido para ser utilizado pelos auditores da CGU;
- Validação da ferramenta desenvolvida no ambiente real da CGU, com avaliação de uso pelos auditores.

1.2 Metodologia

A metodologia desse trabalho está dividida em seis fases consecutivas e complementares, as quais envolvem desde o estudo até a análise dos resultados, a saber:

1. Estudo de conceitos, algoritmos e ferramentas de mineração de dados.
2. Estudo dos conceitos de licitação, auditoria e do Portal de Compras do Governo Federal ComprasNET.
3. Estudo para escolha de algoritmos e linguagens mais adequadas para implementar um protótipo com mineração de dados para auxiliar no processo de detecção de fraudes em licitações públicas.
4. Desenvolvimento de um protótipo com interface Web.
5. Desenvolvimento do estudo de caso para ilustrar a utilização da ferramenta.
6. Análise dos resultados e conclusões.

1.3 Estrutura do Documento

Esse trabalho possui uma fundamentação teórica apresentada no Capítulo 2, incluindo métodos de aprendizagem de máquina, técnicas de mineração de dados englobando o algoritmo Apriori que foi o utilizado nessa monografia, e uma breve apresentação de conceitos relacionados ao processo licitatório. Nesse capítulo também é abordado os trabalhos que já foram desenvolvidos e também utilizam mineração de dados em dados públicos em busca de fraudes.

No Capítulo 3 será apresentada a solução proposta, os trabalho desenvolvido em cada etapa da mineração de dados, a técnica de mineração utilizada, as tecnologias utilizadas e todo trabalho realizado para resolver o problema. Nesse capítulo também serão incluídos os resultados obtidos através da aplicação da solução proposta e utilização da ferramenta desenvolvida pelos auditores da CGU.

Por último, no Capítulo 4, serão apresentadas as conclusões do trabalho e os trabalhos futuros que podem ser realizados em relação ao que já foi desenvolvido.

Capítulo 2

Fundamentação Teórica

Nesse capítulo são apresentados os conceitos relacionados ao trabalho, incluindo aprendizado de máquina com foco em técnicas de mineração de dados e o domínio de aplicação no contexto de processo licitatório do governo federal brasileiro. Nesse capítulo também serão abordados alguns trabalhos que já foram desenvolvidos, os quais utilizam técnicas inteligentes, tais como mineração de dados, para descoberta de fraudes nos processos licitatórios.

2.1 Aprendizado de Máquina

De acordo com Han and Kamber (2005), o aprendizado de máquina pode ser definido como uma área que estuda a maneira com que os computadores podem aprender, ou melhorar seu desempenho de forma automática. Uma área de pesquisa principal é fazer programas de computador aprenderem a reconhecer padrões complexos automaticamente e tomarem decisões inteligentes com base em dados. Existem várias abordagens para o aprendizado de máquina, entre elas o aprendizado por reforço, o aprendizado supervisionado, o aprendizado não-supervisionado e o aprendizado semi-supervisionado.

- Aprendizado por reforço

  Segundo Russell and Norvig (2010), no aprendizado por reforço, o agente aprende com uma série de reforços - recompensas ou punições. Dessa forma, a partir do *feedback* recebido, a aplicação pode priorizar determinadas ações em detrimento de outras. Um exemplo que ilustra essa abordagem é o de um restaurante, onde o garçom pode ou não receber uma gorjeta, dependendo de seu serviço. Se serviu bem os clientes, pode receber uma recompensa (no caso a gorjeta), então naturalmente vai tentar tratar os clientes seguintes da mesma forma que tratou aqueles que lhes deram a gorjeta.

- Aprendizado de máquina supervisionado

  Esse tipo de aprendizado pode ser visto como um sinônimo para classificação (Han and Kamber (2005)). Procura-se generalizar novas entradas em um programa de acordo com entradas-saídas padrões (rótulos dos dados de treino) para obter uma classificação desses dados. Há diversos tipos de aprendizados supervisionadas:

1. Árvores de Decisão - é uma técnica que apresenta o problema a ser resolvido conforme um estrutura de dados em árvore, onde em cada nó é apresentada uma entrada e os dados são analisados e divididos. Esse processo leva a formação de um caminho. O objetivo do algoritmo é que esse caminho seja o menor possível. Para encontrar o menor caminho consistente, usa-se a abordagem dividir para conquistar procurando o atributo que melhor divide os dados.

   Na Figura 2.1 está ilustrada uma árvore de decisão para atendimento em um restaurante.

2. Redes Neurais Artificiais - é um modelo baseado em neurônios biológicos, os quais são ativados quando atingem um determinado valor ou um valor de *threshold*. Os nós da rede são interligados e cada uma das ligações assume um peso que é usado para calcular a função de perda para cada um dos dados de entrada. A função de perda estima o quanto se perde ao escolher o caminho errado para a entrada em questão. Essa função de perda também deve levar em consideração a estrutura da rede, sendo que ela pode apresentar em sua estrutura uma camada única, todas entradas ligadas à saídas, ou várias camadas, tendo camadas não ligadas diretamente à saídas, conforme apresentado na Figura 2.2.

3. Máquina de Vetor de Suporte - é uma técnica recomendada quando se tem pouco ou nenhum conhecimento sobre o domínio de aplicação. Ela consiste em classificar os dados conforme um separador ou de acordo com a distância que os dados mantém do separador, procurando manter essa distância a maior possível. Esse separador é criado para categorizar os dados dado um conjunto de treinamento como pode ser visto na Figura 2.3.

4. K-Vizinhos mais Próximos - essa técnica consiste em procurar os vizinhos mais próximos de uma entrada, dado um cálculo de distância, e classificar essa entrada de acordo com a classificação de seus vizinhos. A escolha do conjunto de treinamento é essencial nessa técnica para evitar os ruídos.

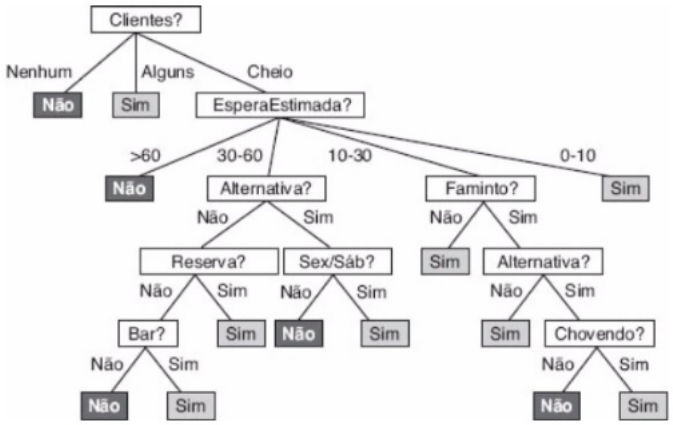

Figura 2.1: Ilustração de uma árvore de decisão (traduzida de Russell and Norvig (2010)).

- Aprendizado de máquina não-supervisionado

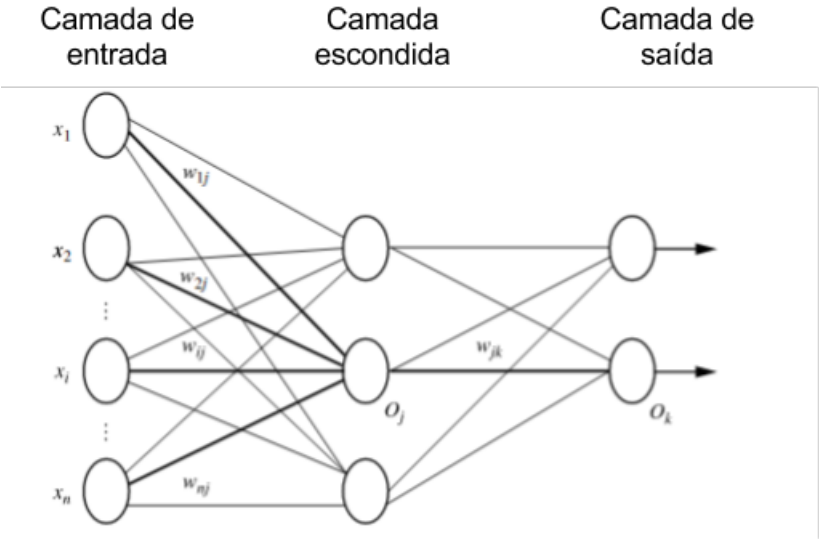

Figura 2.2: Ilustração de uma rede neural de várias camadas (traduzida de Han and Kamber (2005)).

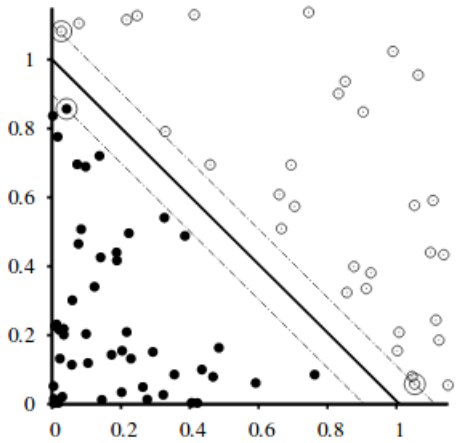

Figura 2.3: Ilustração de uma máquina vetor de suporte Han and Kamber (2005).

O aprendizado não-supervisionada procura semelhança entre os próprios dados de entrada, sem nenhum dado de treino. Para encontrar essas semelhanças, a clusterização é a técnica mais usada. A clusterização consiste em agrupar dados de acordo com as semelhanças existentes nos dados. Geralmente calcula-se a distância entre os valores depois de transformar os dados em valores numéricos. O aprendizado não-supervisionada utiliza entradas não rotuladas, então esse modelo de aprendizagem automática não fornece o significado semântico dos *clusters* encontrados (Han and Kamber, 2005).

Os métodos mais conhecidos dessa abordagem são:

1. K-means - esse método consiste em escolher como centróide de um *cluster* o valor médio dos pontos dentro desse *cluster*. Para isso o algoritmo primeira-

mente escolhe de forma aleatória um número k de agrupamentos e seus centros. Com os agrupamentos definidos, os dados são separados nos agrupamentos de acordo com a distância entre pontos. Isso é feito até que os centros fiquem estáveis como está ilustrado na Figura 2.4.

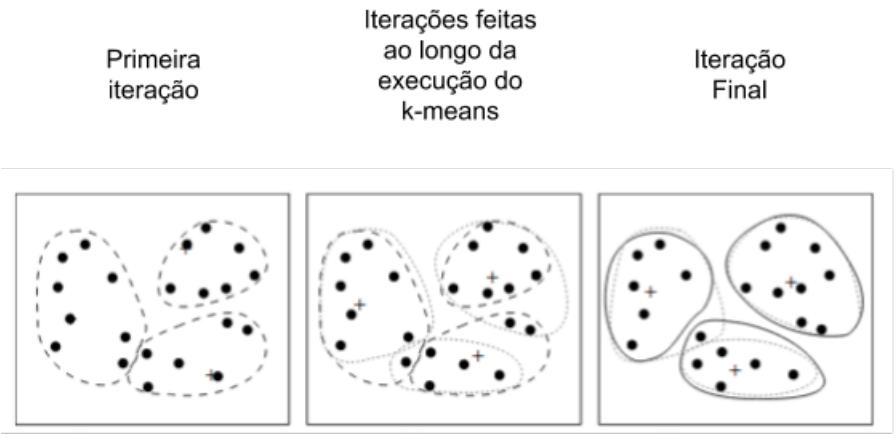

Figura 2.4: Ilustração da aplicação do algoritmo k-means (traduzido de Han and Kamber (2005)).

2. Método Hierárquico - é baseado na união dos dados por similaridades, sendo que os dados vão formar grupos que vão se unir hieraquicamente e assim até que o usuário decida quando essa aglomeração deve ter fim.

- Aprendizado semi-supervisionado

  É uma abordagem que além de possuir dados de entrada-saída padrão (rotulados), possui também dados que não são padrões (não-rotulados). O aprendizado semi-supervisionado procura usar ambos os tipos de dados para classificar as entradas, enquanto as entradas rotuladas são utilizadas para aprender as categorias de classificação, as não-rotulados ajudam a refinar as fronteiras entre as categorias.

2.2 Técnicas de Mineração

De acordo com Han and Kamber (2005), a mineração de dados, tratada por muitas pessoas como *Knowledge-Discovery in Databases - KDD*, é um processo para o descobrir conhecimento a partir de uma grande quantidade de dados. Para sair dos dados e alcançar o conhecimento é necessário uma divisão em fases pelo fato do processo ser muito extenso. Conforme ilustrado na Figura 2.5 as fases de KDD são:

- Limpeza dos dados: remoção de impurezas e dados inconsistentes.
- Integração dos dados: é a fase onde diversos *data sources* podem se combinar.
- Seleção dos dados: busca de dados relevantes na base de dados.
- Transformação de dados: preparação dos dados para a fase de mineração utilizando sumarização, agregação ou técnicas semelhantes.

- Mineração dos dados: aplicação de métodos inteligentes que tem o objetivo de extrair padrões dos dados.
- Avaliação de padrões: identificar os padrões que melhor se adaptam ao objetivo baseado em algumas medidas.
- Apresentação do conhecimento: utilização de técnicas de visualização para apresentar os resultados obtidos ao usuário.

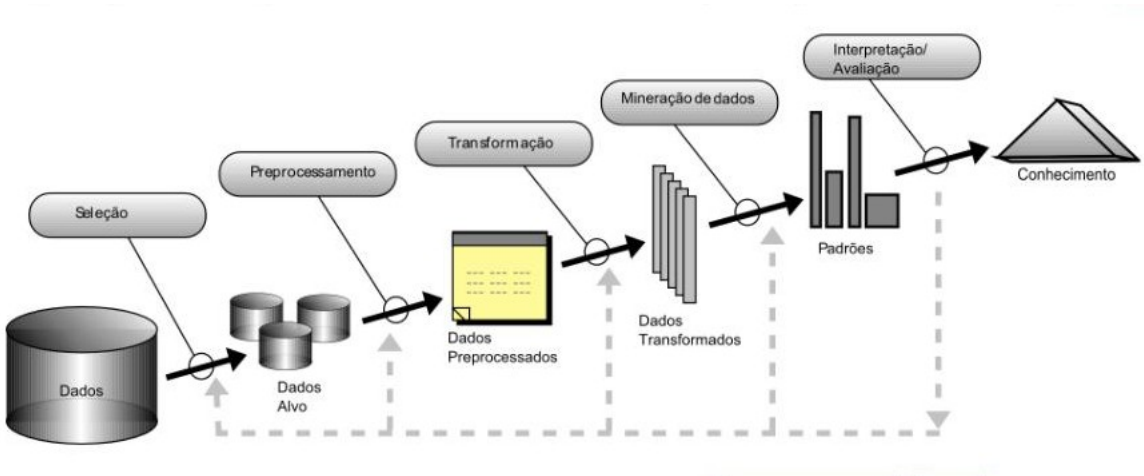

Figura 2.5: Fases do Processo de Mineração de Dados (adaptado de Fayyad et al. (1996))

Para realizar a mineração de dados podem ser utilizadas diversas técnicas, segundo David J. Hand and Smyth (2001):

- Análise de dados exploratória: busca sem um objetivo definido.
- Modelagem descritiva: o objetivo é descrever os dados, podendo ser utilizada técnicas para descobrir distribuição de probabilidade, agrupamento dos dados, entre outras.
- Modelagem preditiva: a função dessa técnica é descobrir o valor de uma variável com base no valor das outras variáveis.
- Descoberta de padrões e regras: o objetivo é encontrar comportamentos incomuns nos dados.
- Recuperação por conteúdo: deve encontrar dados na base de dados com padrões anteriormente definidos.

O Processo CRISP-DM

O *CRoss-Industry Standard Process for Data Mining* (CRISP-DM), é um modelo processual que procurava atender as necessidades da realidade empresarial. Esse modelo é divido em seis fases que estão em constante iteração representadas pela Figura 2.6. Segundo Chapman et al. (2000), as fases do CRISP-DM podem ser descritas como:

- Compreensão do Negócio - é a fase em que é entendido o domínio de aplicação em que as técnicas de mineração de dados serão utilizadas. A partir desse entendimento são traçadas as metas do projeto.
- Entendimento de Dados - o objetivo dessa fase é familiarização com os dados e identificação de possíveis falhas nos dados. É nessa fase que os dados podem sugerir um conjunto ou agrupamento relevante para alcançar as metas traçadas no passo anterior.
- Preparação dos Dados - essa fase contém várias tarefas que procuram construir o conjunto final dos dados. Essas tarefas podem incluir desde limpeza dos dados até integração de diferentes tipos de dados.
- Modelagem - nessa fase as técnicas de modelagem são aplicadas de fato e, para isso, pode ser necessário voltar ao passo de preparação dos dados para que a entrada esteja ajustada a técnica desejada. Os testes para validação da aplicação da modelagem também são definidos nessa fase.
- Avaliação - essa fase tem a função de verificar se o resultado obtido até o momento atende ao objetivo do negócio considerado na fase de compreensão do negócio.
- Implantação - essa é a última fase do CRISP-DM e diz respeito a apresentação dos resultados ao usuário para que dê suporte a tomada de decisão.

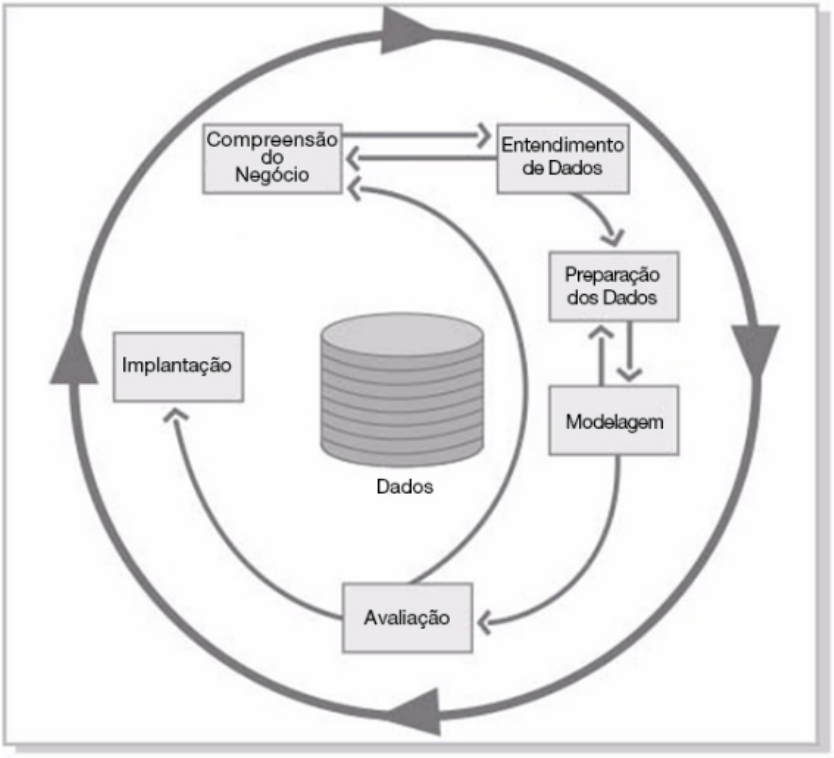

Figura 2.6: Fases do modelo CRISP-DM (adapatado de Chapman et al. (2000))

As Técnicas

- Clusterização

  Segundo Jain and Dubes (1988), clusterização é uma tarefa descritiva onde se procura identificar um conjunto finito de categorias ou *clusters* para descrever uma informação. Existem diversas técnicas para a separação dos dados em agrupamentos ou *clusters*. A clusterização é definida como classificação não supervisionada, pelo fato de novos dados serem classificados de acordo com a classificação dos dados anteriores.

  A clusterização geralmente se dá por (Jain and Dubes, 1988):

  1. Representação de um padrão.
  2. Definição de uma aproximação entre os dados.
  3. Agrupamento dos dados.
  4. Abstração de dados.
  5. Avaliação dos resultados.

- Classificação

  De acordo com Han and Kamber (2005), a classificação é uma forma de análise de dados que extrai modelos que descrevem classes importantes de dados. Esses modelos são chamados classificadores cujo objetivo é categorizar os rótulo de uma classe. A maioria dos algoritmos é residente em memória, geralmente assumindo um pequeno tamanho de dados. A classificação possui inúmeras aplicações, incluindo detecção de fraude, marketing alvo, previsão de desempenho, fabricação e diagnóstico médico.

  Após o uso dos dados de treinamento para obter um modelo classificado é possível avaliar esse modelo com as métricas:

  – verdadeiro positivo (TP): classificação correta dos dados em relação às classes que pertencem.
  – falso positivo (FP): classificação dos dados à classe que não pertencem verdadeiramente.
  – verdadeiro negativo (TN): classificação correta em relação a classes que não pertencem.
  – falso negativo (FN): classificação incorreta como não pertencentes de classes que, verdadeiramente, pertencem.

  Os valores TP, FP, TN, FN aparecem nas chamadas **matrizes de confusão**. Essas matrizes é uma ferramenta útil para analisar o quão bem o seu classificador pode reconhece as tuplas de diferentes classes. Os valores TP e TN nos dizem quando o classificador está classificando corretamente, enquanto os valores FP e FN nos dizem quando o classificador está classificando de erroneamente. É possível ver a matriz na Figura 2.7, onde P é o número de tuplas que foram rotuladas como positivas e N é o número de tuplas que foram rotulados como negativos.

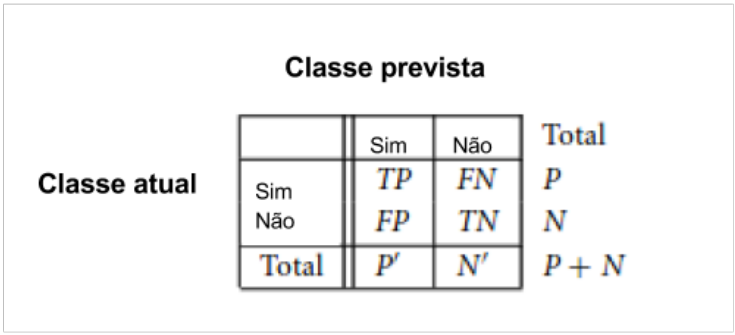

Figura 2.7: Matriz de confusão (traduzida de Han and Kamber (2005))

Uma das maneiras de calcular o desempenho do classificador é através da curva ROC (*Receiver Operating Characteristic*) que expõe o *trade-off* entre capacidade de detectar casos que devem ser detectados e falsos alarmes. Quanto mais brando for o critério do classificador para a detecção de um dado como pertencente à uma categoria, maior a chance de aparecer um falso alarme. Uma forma de analisar o desempenho de um classificador é analisar a Área sob a Curva ROC (Figura 2.8) pois esse valor leva em consideração os diferentes pontos de *trade-off* da curva. Quanto maior esse valor, mais o gráfico consegue atingir uma taxa alta de verdadeiros positivos sem aumentar tanto a taxa de falsos positivos.

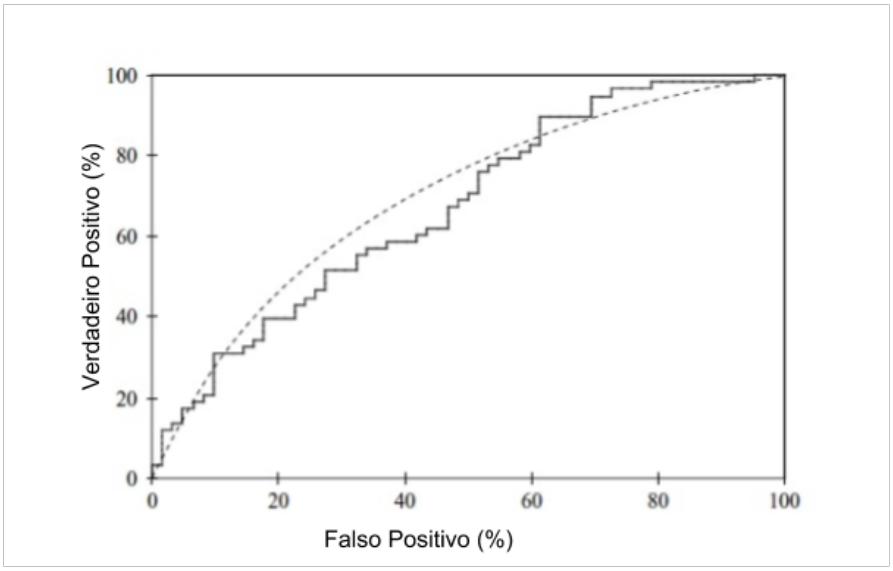

Figura 2.8: Exemplo de curva ROC (traduzida de Witten et al. (2011))

- Regras de Associação

  São técnicas de mineração de dados que tem por objetivo descobrir relações fortes entre os dados, relacionando padrões aos dados que estão associados. Existem duas

grandezas importantes para que a a regra de associação encontre bons relacionamentos entre os dados:

- Suporte: representa a chance da regra aparecer na base de dados analisada.
- Confiança: representa o percentual de vezes que a regra apareceu corretamente, quanto maior a confiança, melhor é a qualidade da regra.

A Tabela 2.1 apresenta uma transação de exemplo para calcular o valor do suporte e da confiança. O cálculo é feito para o conjunto {Café, Pão} que aparecem nas transações. Para o cálculo do suporte, foi verificado que os itens Café e Pão apareceram 3 vezes juntos nas transações, também foi verificado que há 7 transações no total, levando a um suporte de 0,4. O cálculo da confiança foi baseado no fato dos itens Café e Pão apareceram 3 vezes juntos nas transações e do item Café aparecer 3 vezes nas transações, levando a uma confiança de 1. A partir desses cálculos é possível extrair das transações a regra {Café} ⇒ {Pão} com suporte de 40% e confiança de 100%.

Tabela 2.1: Exemplo para o cálculo de suporte e confiança

	Café	Pão
Transação 1	**1**	**1**
Transação 2	0	1
Transação 3	**1**	**1**
Transação 4	**1**	**1**
Transação 5	0	0
Transação 6	0	0
Transação 7	0	1

$$\{\text{Café, Pão}\}$$
$$\text{Suporte} = 3 \div 7 = 0.4$$
$$\text{Confiança} = 3 \div 3 = 1$$
$$\{\text{Café}\} \Rightarrow \{\text{Pão}\}$$

Uma outra variável muito importante nas regras de associação é o *lift* que é uma medida de correlação simples que é dada da seguinte forma. A ocorrência do *itemset* A é independente da ocorrência do *itemset* B se P $(A \cup B)$ = P (A) P (B). Caso contrário, os *itemsets* A e B são dependentes e correlacionados como eventos. Esta definição pode ser facilmente estendida para mais de dois *itemsets*. O levantamento entre a ocorrência de A e B pode ser medido pela fórmula:

$$lift(A, B) = \frac{P(A \cup B)}{P(A).P(B)} \tag{2.1}$$

Segundo Han and Kamber (2005), se o valor da equação for inferior a 1, os itemsets A e B são negativamente correlacionados, ou seja, a ocorrência de um provavelmente leva a ausência do outro. Já se o valor for maior que 1, eles são positivamente correlacionados, o que significa que a ocorrência de um provavelmente leva a ocorrência

do outro. Se o valor for exatamente 1, os valores são independentes e então não há correlação entre eles.

De acordo com Han and Kamber (2005), regra de associação pode ser definida conforme descrita na Definição 1.

Definição 1: *Seja $I = \{I_1, I_2, ..., I_M\}$ um conjunto de itens e D os dados da base, onde T é o conjunto de transações de D, tal que $T \subseteq I$. Sejam também A e B conjuntos de itens. Uma regra de associação é uma implicação da forma $A \Rightarrow B$, onde $A \subset I$, $B \subset I$ e $A \cap B = \emptyset$. A regra $A \Rightarrow B$ se aplica no conjunto de transações D com suporte s, onde s é o percentual de transações em D, que contém $A \cup B$, isto é, a probabilidade $P(A \cup B)$. A regra $A \Rightarrow B$ tem confiança c no conjunto de transações D, onde c é o percentual de transações em D contendo A, que também contém B, isto é, a probabilidade condicional $P(A|B)$.*

Para encontrar fortes associações entre os dados, Agrawal and Srikant (1994) propuseram o algoritmo Apriori.

- Apriori - é um algoritmo que extrai de regras de alta confiança, assim, encontrando relações entre os dados. Esse algoritmo utiliza uma abordagem iterativa que consiste dos $k-itemsets$ serem usados para explorar os $(k+1)-itemsets$. Primeiramente, o conjunto $1-itemsets$ de itens frequentes é encontrado buscando no banco de dados a presença de cada item e fazendo sua contagem e coletando os itens que satisfazem o suporte mínimo. O conjunto resultante é indicado por L1. Em seguida, L1 é usado para encontrar L2, o conjunto $2-itemsets$ de itens frequentes, que é usado para encontrar L3, e assim por diante, até que não se encontrem mais $k-itemsets$ frequentes. A descoberta de cada L_k requer uma verificação completa do banco de dados. O Algoritmo 2.1 mostra a implementação do algoritmo Apriori, o qual é baseado nos Teoremas 1 e 2 (Han and Kamber, 2005), a seguir:
  - Teorema 1: Se em uma regra $X \rightarrow Y$, X não satisfaz a confiança, em qualquer regra $X' \rightarrow Y$, X', sendo um subconjunto de X, não irá satisfazer a confiança também.
  - Teorema 2: Se um subconjunto é frequente, todos seus subconjuntos também serão.

Algoritmo Apriori 2.1

```
L1 = find_frequent_1-itemsets(D);
for(k = 2; L_{k-1} = ∅; k++){
        C_k = apriori_gen(L_{k-1});
        for each transaction t ∈ D{ //scan D for counts
            C_t = subset(C_k, t);
            //get the subsets of t that are candidate c ∈ C_t
                for each candidate c ∈ C_t {
                        c.cout++;
                }
```

```
        }
        Lk = {c ∈ Ck | c.count ≥ min_sup}
}
return L = ∪ kLk;

procedure apriori_gen(Lk-1:frequent(k−1)−itemsets)
        for each itemset l1 ∈ Lk-1
                for each itemset l2 ∈ Lk-1
                        if(l1[1] = l2[1]) ∧ (l1[2] = l2[2])
                        ∧ ... ∧
                        (l1[k−2] = l2[k−2]) ∧
                        (l1[k−1] < l2[k−1]) then {
                        c = l1 ⋈ l2
                        //join step:generate candidates
                        if has_infrequent_subset(c, Lk-1) then
                                delete c;
                                //prune step: remove
                                //unfruitful candidate
                        else add c to Ck;
                        }
        return Ck;

procedure has_infrequent_subset(c: candidate k−itemset;
Lk-1: frequent(k−1)−itemsets);
// use prior knowledge
        for each (k−1)−subset s of c
                if s ∉ Lk-1 then
                        return TRUE;
        return FALSE;
```

Na Tabela 2.2 estão 10 transações que serão utilizadas para exemplificar o funcionamento do algoritmo Apriori. Inicialmente, são criados conjuntos com cada um dos itens da transação e calculado seu suporte (Tabela 2.3). O próximo conjunto será com 2 itens das transações, serão combinados os conjuntos com cada um dos itens com suporte maior que o desejado. Usando um suporte de 20%, todos os conjunto serão combinados gerando novos conjuntos (Tabela 2.4). Esse processo é repetido (Tabela 2.5) até que não haja mais o que combinar.

Na Seção 2.2 serão apresentados os conceitos principais do domínio de aplicação desse trabalho.

2.3 Licitações Públicas Brasileiras

De acordo com Brasil (1993), licitação é um procedimento pelo qual o poder público realiza compra de produtos ou serviços. É a forma que assegura a plena concorrência entre os participantes procurando garantir a observância do princípio constitucional da isonomia, onde a seleção da proposta mais vantajosa para a administração é assegurada.

Tabela 2.2:

	Item 1	Item 2	Item 3	Item 4	Item 5
Transação 1	1	1	0	0	1
Transação 2	0	1	0	1	0
Transação 3	0	1	1	0	0
Transação 4	1	1	0	1	0
Transação 5	1	0	1	0	0
Transação 6	0	1	1	0	0
Transação 7	1	0	1	0	0
Transação 8	1	1	1	0	1
Transação 9	1	1	1	0	0
Transação 10	0	0	0	0	0

Tabela 2.3:

Conjunto	Suporte
1	0.6
2	0.7
3	0.6
4	0.2
5	0.2

Tabela 2.4:

Conjunto	Suporte
1,2	0.4
1,3	0.4
1,4	0.1
1,5	0.2
2,3	0.4
2,4	0.2
2,5	0.2
3,4	0
3,5	0.1
4,5	0

Tabela 2.5:

Conjunto	Suporte
1,2,3	0.2
1,2,5	0.2
2,3,4	0
2,4,5	0

- Modalidades
  - Concorrência: é a modalidade com o maior número de participantes, pois a seleção se dá entre todos interessados que se mostraram aptos a participar.

- Tomada de Preços: a seleção é feita em um cadastro prévio ou entre participantes que atenderem a todas as condições exigidas para cadastramento até o terceiro dia anterior à data do recebimento das propostas.
- Convite: nessa modalidade, a licitação é feita entre, no mínimo, três interessados do ramo pertinente. O interesse deve ser demonstrado através do cadastro, pelo menos 24 horas antes da apresentação da proposta.
- Concurso: a seleção para licitação dessa modalidade é feita entre todos os aptos para escolha de trabalho técnico, científico ou artístico para concessão de prêmios ou remuneração aos participantes vencedores.
- Leilão: é para venda de bens móveis inservíveis para a administração ou de produtos legalmente apreendidos ou penhorados para todos os interessados que oferecerem lance igual ou superior a avaliação.
- Pregão: é a modalidade de licitação para compra de bens e serviços comuns através de propostas em sessão pública, independente do preço.

Irregularidades

A auditoria tem sido um processo importante na busca e prevenção das irregularidades nas licitações públicas brasileiras. As licitações públicas são grande alvo de corrupção por estar ligada ao sistema financeiro, além de seu processo conter falhas. Entre as irregularidades, pode-se citar:

- Vínculo entre licitante e servidores: a participação de empresa com sócio com vínculo familiar com algum licitante vai de encontro ao art. 9º, inciso III, da Lei 8.666/1993. Essa situação pode constituir indício de simulação e fraude à licitação
- Fracionamento de despesas: consiste basicamente em dividir a despesa para utilizar a modalidade de licitação inferior à determinada pela Lei, ou mesmo para realização da dispensa de licitação em razão do valor, pois apenas é necessário realizar licitação para compras acima de R$8.000,00 reais.
- Rodízio em licitações: é o acordo entre os participantes da licitação para ocorrer uma alternância de vitória em licitações visando o superfaturamento de preços.
- Conluios e cartéis: é um acordo entre concorrentes para principalmente fixação de preços de produção, divisão de clientes e de mercados de atuação. Dessa maneira é possível eliminar a concorrência, com o consequente aumento de preços e redução de bem-estar para o consumidor.

2.4 Trabalhos Correlatos

Dada a situação econômica e financeira atual do Brasil e do mundo, a corrupção tem sido alvo de grande preocupação. No relatório OECD (2016) da *The Organisation for Economic Co-operation and Development (OECD)* consta que as atividades governamentais são altamente propensas a corrupção devido ao grande volume de transações, ao interesse financeiro e à complexidade dos processos públicos. Pode-se ver na Figura 2.9 a quantidade de suborno que é utilizada em atividades governamentais, como compras

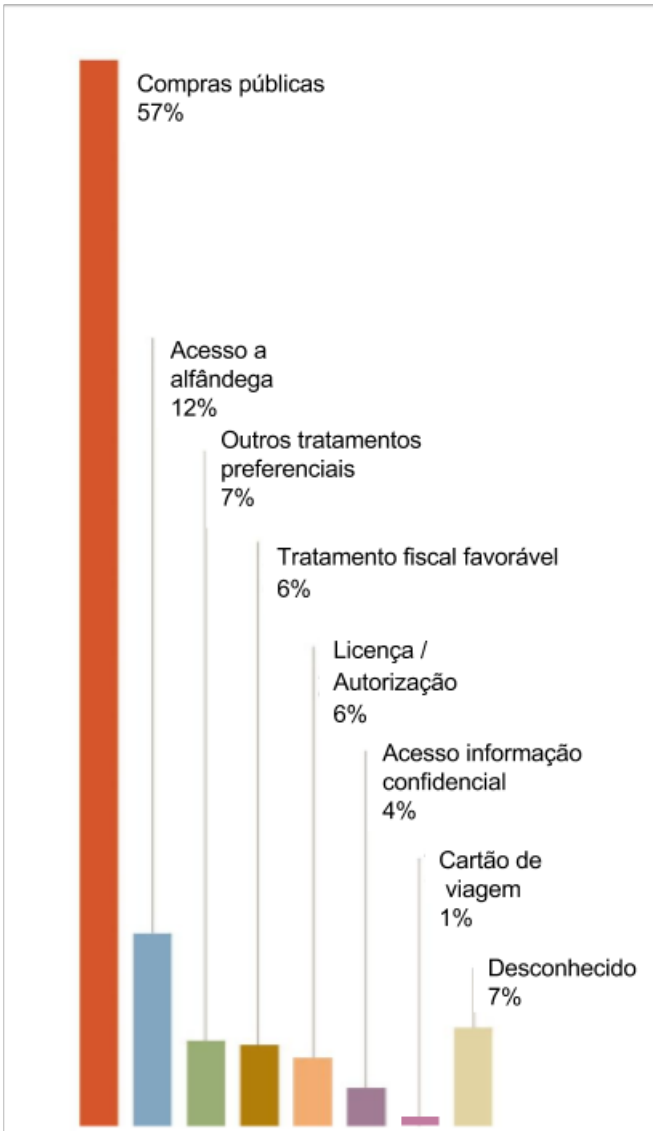

Figura 2.9: Tipos de operações publicas com uso de suborno (traduzido de OECD (2016))

públicas, acesso a alfândega, outros tratamentos preferenciais, tratamento fiscal, licença, por exemplo. Tem-se procurado detectá-las de diversas formas e isso pode ser visto pelos trabalhos desenvolvidos em diversas partes do mundo.

Utilização de Agentes de Mineração nos Dado de Licitações Públicas para Detectar Fraudes

No âmbito da CGU, Ralha and Silva (2012) propõem uma solução útil para detecção de cartéis em licitações públicas utilizando o portal ComprasNet. Nesse trabalho foram utilizadas duas áreas de pesquisa interessantes: mineração de dados em ambiente distribuído e sistemas multiagentes. Os agentes do sistema denominado AGMI (*AGent-MIning tool*) são autônomos e podem deliberar sobre técnicas adequadas de mineração de dados conforme os dados de licitações públicas brasileiras constantes do ComprasNet. Foi de-

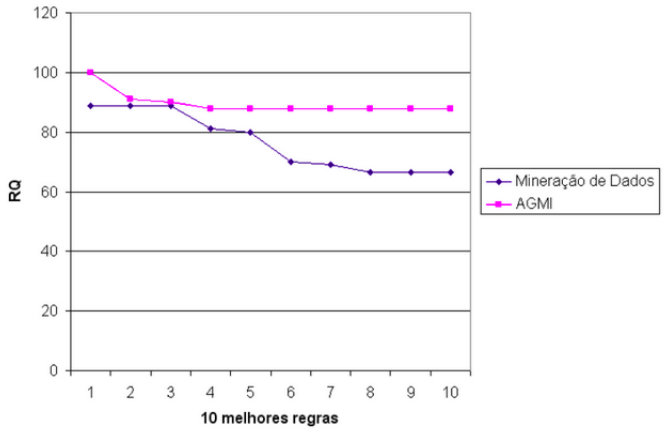

Figura 2.10: Comparação de qualidade da regra (RQ) entre as 10 melhores regras de AGMI e DM (retirado de Ralha and Silva (2012)).

finido uma arquitetura em três camadas (estratégica, tática e operacional) com diversos tipos de agentes que foi definida para aplicar diferentes técnicas de mineração de dados com o objetivo de melhorar o conhecimento descoberto. O uso da AGMI e dos agentes de mineração de dados criaram uma ferramenta flexível que consegue utilizar bases de dados de diferentes tamanhos independente do recurso disponível. Essa abordagem levou a ótimos resultados com um bom desempenho em termos de performance e produção de regras de qualidade. As regras de associação descobertas apresentaram indícios de irregularidades em licitações. Como resultado desse trabalho foi possível verificar que a AGMI obteve um bom desempenho em termos de performance e produção de regras de qualidade. É possível observar na Figura 2.10 que as dez melhores tiveram uma média de melhoria de 58,48% na sua qualidade com o uso da AGMI em relação ao uso apenas da mineração de dados.

Uso de Tecnologias Semâticas no Processo Licitatório na Sérvia

Minović et al. (2014) procurou introduzir tecnologias semânticas no processo licitatório na Sérvia para permitir a manipulação de dados por máquinas. Com o modelo criado nesse trabalho seria possível que um especialista estabelecesse regras relativas a condições específicas, a fim de ser alertado sobre possíveis aquisições irregulares. A aplicação da solução proposta deve permitir o reconhecimento prévio de aquisições potencialmente irregulares, o que pode ser prevenido antes da realização ou sancionado antes da obsolescência.

Modelo de Banco de Dados NoSQL com Objetivo de Detectar Fraudes em Processos Licitatórios

Erven (2015) propôs um modelo para um banco de dados NoSQL baseado em grafos utilizando Neo4j [1]. Esse modelo foi validado utilizando a implementação de um banco de dados de vínculos societários de empresas, combinados com os relacionamentos dessas

[1] https://neo4j.com/

pessoas jurídicas com os processos de licitação pública brasileira. O objetivo do modelo desenvolvido foi auxiliar na detecção de fraudes em processos licitatórios. Esse vínculo pode ser visto na Figura 2.11 que representa uma pesquisa pelo menor caminho entre duas empresas.

Nesse trabalho foi possível verificar a vantagem do banco de dados em grafos para buscas que envolvem a navegação entre os relacionamentos. Outra verificação apresentada foi na busca de vínculos foi uma obtida uma vantagem nas consultas do Neo4j dada a estrutura de dados orientada a grafos.

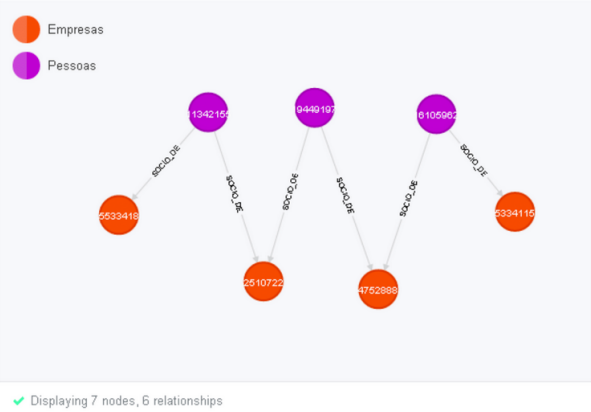

Figura 2.11: Menor caminho entre duas empresas utilizando Neo4j (retirado de Erven (2015)).

A mineração de dados também vem sendo aplicada em outros domínios como pode ser visto em Nelson Hein (2014) que utiliza análise fatorial como técnica de mineração de dados para selecionar indicadores de desempenho social de empresas do setor de consumo cíclico de maior representatividade na economia. Com os resultados dessa pesquisa e com a análise de desempenho é possível gerar as organizações informações acerca da sua situação e tendências futura das empresas. Nguyen et al. (2017) mostra o estado da arte do uso da mineração de dados para gerenciamento de cadeias de suprimentos e discute quais são as técnicas de mineração de dados utilizadas nesse domínio, em que áreas está presente a mineração de dados, entre outras questões.

No Capítulo 3 será apresentada a proposta de solução deste trabalho, a qual visa auxiliar os auditores na tarefa de encontrar relações entre empresas participantes de licitações públicas do governo federal.

Capítulo 3

Proposta de Solução

Nesse capítulo é apresentada a solução proposta com o objetivo desenvolver uma ferramenta que possa auxiliar os auditores na tarefa de encontrar relações entre as empresas participantes de licitações públicas e possíveis fraudes. Nesse Ccpítulo também serão incluídos os resultados obtidos através da aplicação da solução proposta e utilização da ferramenta desenvolvida para os auditores da CGU.

3.1 Modelo Conceitual

O modelo apresentado na Figura 3.1 mostra como as fases da mineração de dados, apresentadas na Figura 2.5 foram utilizadas nesse trabalho. Cada uma das fases utilizadas serão abordadas abaixo.

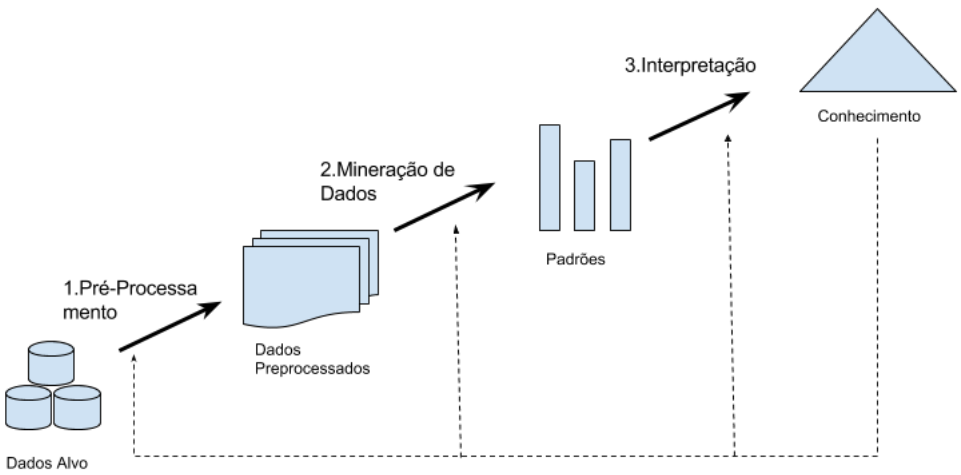

Figura 3.1: Fases do modelo de mineração de dados

Pré-processamento

O pré-processamento foi um processo de escolha dos dados entre todos disponíveis, ou seja, entre as 43 colunas de dados disponíveis apresentadas na Tabela 3.1. Dessas colunas,

Tabela 3.1: Colunas de dados disponibilizadas no portal ComprasNet

Ambiente de execução	
Identificação_Compra	Cod_OrgaoSup
Modalidade_Licitação	Nome_OrgaoSup
Tipo_Licitação	**Cpf_Cnpj_Fornecedor**
Justif_Dispensa_Inexig	Nome_Fornecedor
Inciso_Dispos_Legal	Municipio_Fornecedor
Data_Referencia_Compra	UF_Fornecedor
Data_Resultado_Compra	Cpf_Cnpj_Fornecedor_Representante
Forma_de_Compra	Nome_Fornecedor_Representante
Identificação_ItemCompra	Municipio_Fornecedor_Representante
Codigo_MaterialServico	UF_Fornecedor_Representante
Descricao_MaterialServico	**Valor_Total_Homologado**
TipoMaterialServico	QT_OFERTADA
ClasseMaterialServico	VL_ACEITO
GrupoMaterial	VL_ULT_RENEG_ATA_SRP
GrupoServivo	VL_PRECO_UNIT_HOMOL
Cod_Unidade	VL_ULT_LANCE
Nome_Unidade	Mes_referencia_Compra
UF_Unidade	Ano_Referencia_Compra
Regiao_Unidade	Mes_Resultado_Compra
Cod_Orgao	Ano_Resultado_Compra
Nome_Orgao	**Poder_Unidade**
Esfera_Unidade	

apenas 6 foram consideradas relevantes para esse projeto: o identificador da compra, o CNPJ da empresa que participou da licitação, a data de referência, o valor da compra e a modalidade da licitação que estão marcadas em negrito na Tabela 3.1. Essas colunas foram escolhidas pois a partir delas seria possível identificar o relacionamento entre as empresas.

De posse dos dados de todas essas colunas, foi necessário desenvolver um programa em Python para obter outro arquivo que contivesse apenas as colunas desejadas. Os dados obtidos foram utilizado para popular o banco de dados.

Mineração de Dados

A mineração dos dados foi feita utilizando os dados pré-processados já disponíveis no banco de dados. O algoritmo utilizado nesse trabalho foi o Apriori. Para aplicar o algoritmo nos dados foi utilizada uma biblioteca do Python e sua execução será detalhada na Seção 3.2.

Interpretação e Avaliação

Descoberto os padrões de relacionamento entre as empresas através das regras de associação, as mesmas serão apresentadas aos auditores da CGU através de uma interface

web. Sendo assim, a interpretação e avaliação dessa informação será feita pelos próprios auditores da CGU.

3.2 Modelo Implementacional

O protótipo desenvolvido visa auxiliar os auditores da CGU a encontrarem fraudes nos relacionamento entre as empresas participantes das licitações públicas. Para isso, foi idealizada uma interface *web* que apresentasse informações sobre esses relacionamentos, de forma que essas informações facilitem a busca dos auditores pelas possíveis fraudes.

Para a construção do protótipo, foi adotada a utilização de duas linguagens de programação, Python[1] e R[2]. A escolha dessas linguagens se deve ao desejo que integrar essa ferramenta com outras já implantadas na CGU e desenvolvidas em Python. Facilitaria a integração se esse protótipo também fosse desenvolvido em Python. O R foi o utilizado para a aplicação do algoritmo, mas não dificulta a integração, pois foi utilizada uma biblioteca que permite executar código em R no código Python. O protótipo foi subdividido em três componentes principais: Web, Dados e Apriori de acordo com a Figura 3.2 que serão detalhados abaixo.

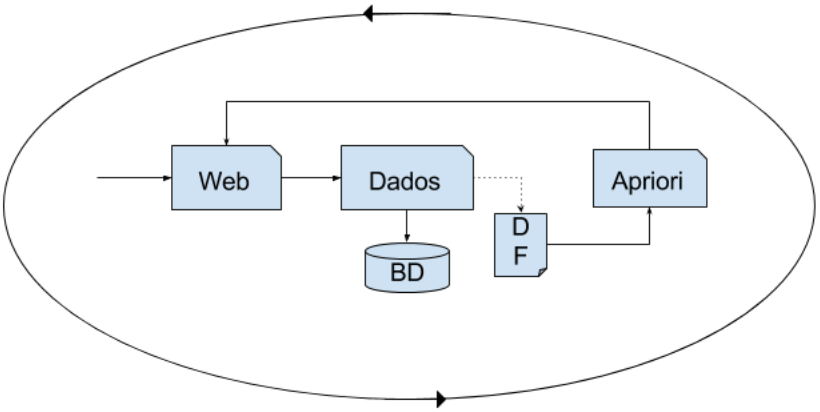

Figura 3.2: Componentes do protótipo

Web

O componente Web é responsável por receber um determinado CNPJ por meio de requisições *The Hypertext Transfer Protocol* (HTTP), realizadas, por exemplo, por um navegador, ao enviar este CNPJ para ser feito o processamento e retornar uma resposta no formato *Hypertext Markup Language* (HTML).

Foi desenvolvido utilizando a linguagem Python e o código implementado pode ser visto no Apêndice A.1. Utilizado para construção da interface *web* foi escolhido o CherryPy[3], por ser um *framework* que possibilita a criação de aplicações *web* da mesma forma

[1]https://www.python.org/
[2]https://www.r-project.org/
[3]http://cherrypy.org/

que se criaria qualquer outro programa Python orientado a objetos. Isso resulta em código-fonte menor desenvolvido em menos tempo.

Dados

Esse componente é o responsável por acessar o banco de dados e procurar todas as compras em que o CNPJ informado no componente Web estEJA presente. Na sequência, o componente de Dados retorna a lista de CNPJs que também participaram de cada uma das compras. O componente disponibiliza esse conjunto de CNPJs para o componente Apriori através de uma estrutura de dados denominada *dataframe*.

O componente Dados também foi desenvolvido em Python como pode ser visto no Apêndice A.2. O banco de dados acessado por esse componente foi modelado utilizando a ferramenta *MySQL Workbench Community*[4] na versão 6.3.9. Foi construída uma tabela que possui uma coluna correspondente a cada uma que foi extraída no pré-processamento.

Apriori

Visando identificar as possíveis correlações entre as empresas foram aplicadas regras de associação nos dados das licitações disponíveis no ComprasNet.

O algoritmo utilizado foi o Apriori, o qual minera itens frequentes descrito pelo algoritmo **??**. Para um conjunto ser considerado frequente, ele deve aparecer nas transações em, pelo menos, uma quantidade igual ao suporte mínimo. O conjunto de itens analisado pelo algoritmo serão os CNPJs retornados pelo componente Banco. O fluxo de execução da fase de mineração do protótipo está detalhado na Figura 3.3.

Para exemplificar a aplicação do Apriori nos dados de licitação foi utilizado o CNPJ 007XXXXXXXXXX. Uma das regras resultantes da aplicação foi a seguinte:

```
                [1]
1           039XXXXXXXXXX
2                   →
3           676XXXXXXXXXX
4             0.3333333
5                   1
```

Observe nessa regra, especificamente nas Linhas 1 e 3 que estão apresentados os CNPJs relacionados pela regra. Na Linha 4 está apresentado o suporte e na Linha 5 a confiança utilizados na aplicação da regra. Pode-se concluir a partir dessa regra que em 33% das compras realizadas em conjunto entre os CNPJs 007XXXXXXXXXX e 039XXXXXXXXXX, o CNPJ 676XXXXXXXXXX também estava contido no conjunto dos compradores.

A partir da aplicação do Apriori, temos as regras resultantes usando o suporte e confiança determinados anteriormente. Usando um conjunto de CNPJs como exemplo, podemos encontrar algumas regras de associação entre eles. Essas regras estão descritas

[4]https://www.mysql.com/products/workbench/

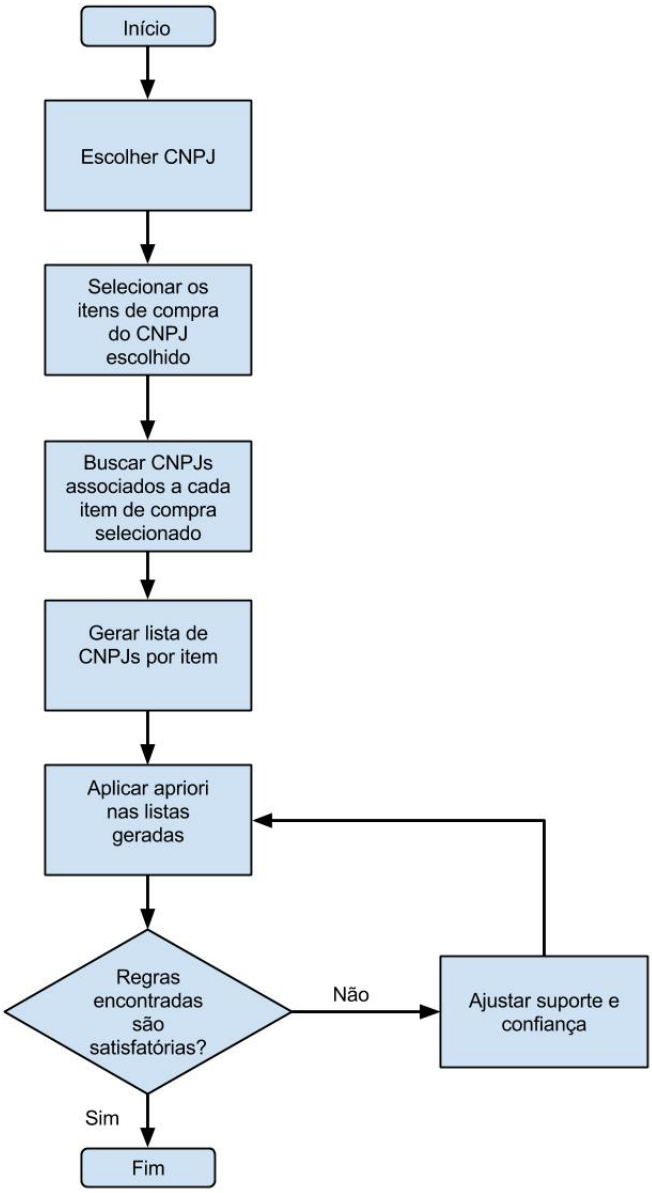

Figura 3.3: Fluxograma da execução da mineração de dados com o Algoritmo Apriori.

na Tabela 3.2 e podem ser visualizadas de acordo com o suporte e confiança de cada uma na Figura 3.4 gerada pela linguagem R.

Segundo Ralha and Silva (2012), valores altos de suporte mínimo para execução do algoritmo nessa aplicação específica não nos garante boas regras, pois uma regra que associa alguns fornecedores e que tem suporte alto provavelmente indica a presença de grandes fornecedores participando de várias licitações. Desta forma, a configuração de um suporte mínimo alto para execução do algoritmo pode suprimir a aparição de diversas regras boas. Valores altos de confiança, por sua vez, garantem a seleção de regras boas. No caso específico do problema de Rodízio de Licitações, o valor alto de confiança garante

Tabela 3.2: Regras que resultante do Apriori

Empresa 1	Empresa 2	Suporte	Confiança	Lift
13303039000120	16500873000101	42.8%	1	1.75
16500873000101	13303039000120	42.8%	0.75	1.75
16500873000101	19915068000129	42.8%	0.75	1.75
19915068000129	16500873000101	42.8%	0.75	1.75

que a frequência de ocorrência de todos os fornecedores que aparecem na regra gerada pelo algoritmo seja aproximada. Desta forma, todos os fornecedores que aparecem na regra como um todo, podem ser considerados um grupo para fins de identificação de cartéis.

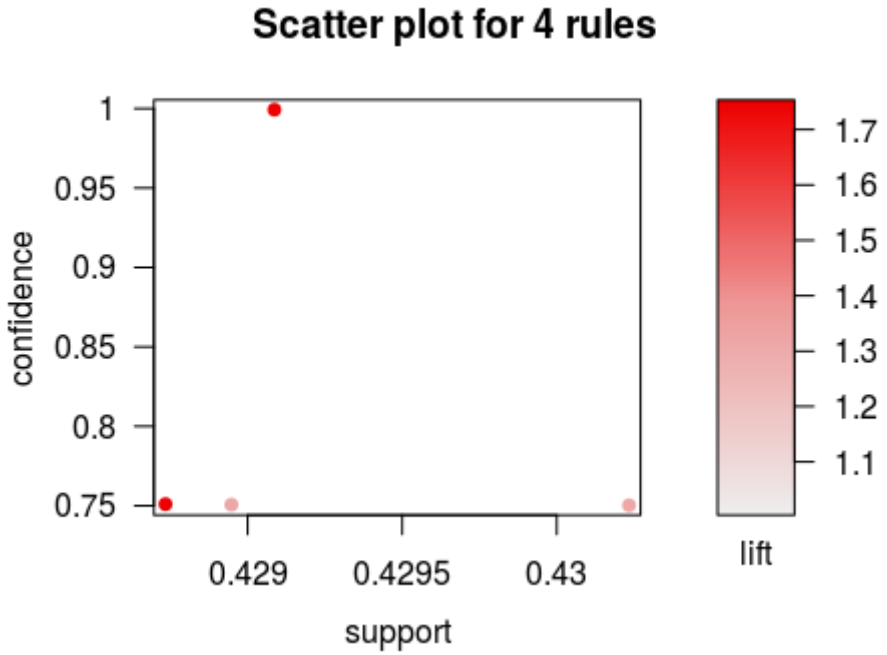

Figura 3.4: Regras encontradas com aplicação do Apriori

No protótipo, a aplicação do algoritmo Apriori foi feita utilizando uma biblioteca do Python que permite acessar os recursos da linguagem R com o Python, a rpy2 [5]. Como uso dessa biblioteca foi possível utilizar o Apriori implemetado pelo R, como pode ser visto no Apêndice A.3.

3.3 Discussão dos Resultados

O protótipo desenvolvido tem a interface apresentada na Figura 3.5. Note que nela estão presentes o CNPJ das empresas participantes da regra, o número de vitórias das empresas participantes da licitação, o *lift*, a confiança e o suporte utilizados no algoritmo Apriori. O campo de busca é para incluir o CNPJ que estará presente do lado esquerdo

[5] https://rpy2.readthedocs.io/en/version_2.8.x/

das regras geradas pelo algoritmo. A coluna *Empresa* e a coluna *Participou com* são as empresas que participaram das licitações junto com CNPJ buscado e aparecem do lado direito das regras (não havendo limites de número de empresas). A coluna *Suporte* é a probabilidade da regra se repetir no conjunto de dados. A coluna $Confianca$ é a quantidade de instâncias preditas corretamente pela regra. A coluna $Lift$ representa o desempenho da regra de associação, em outras palavras, é a resposta alvo dividida pela resposta média.

Figura 3.5: Interface do protótipo desenvolvido.

Inicialmente, o protótipo foi executado localmente em um computador Intel i7, com CPU de 1.80GHz e 8GB de memória RAM e levava cerca de 8 segundos para fazer as consultas no banco e aplicar o algoritmo (tempo aceitável para uso da ferramenta pelos usuários). Depois dos testes locais, foram iniciados os primeiros testes na CGU para captar alguns *feedbacks* da execução do protótipo. Hoje, o protótipo desenvolvido está executando na CGU com acesso a uma base de 123.940.403 registros (Tabela 3.3) que correspondem a 20 anos de dados de licitações públicas, de 1997 a 2017.

Tabela 3.3: Quantidade de registros em relação ao ambiente de execução

Ambiente de execução	Número de Registros
Local	4.482.006
Primeiros testes na CGU	37.522.993
Ambiente real na CGU	123.940.403

O protótipo apresentado mostra as empresas que podem ter algum tipo de correlação. Dado os CNPJs indicados pelo protótipo é possível fazer uma busca em outra ferramenta utilizada pela CGU que utiliza grafos para identificar vínculos entre empresas. Sendo assim, foi utilizada uma das pesquisas realizadas no protótipo desenvolvido nesse trabalho para realizar uma busca no sistema da CGU e verificar a existência de vínculos.

A Figura 3.7 mostra a busca feita no sistema da CGU utilizando os CNPJs que mostram fortes relações de acordo com o protótipo desenvolvido neste trabalho (3.6). O centro em vermelho do grafo são CNPJs, os círculos em laranja são informações de pessoas ou empresas ligadas ao CNPJ em questão. É possível visualizar que há ligação entre dois CNPJs que dividem uma aresta no grafo. Essa ligação aparece no grafo, pois o empregado

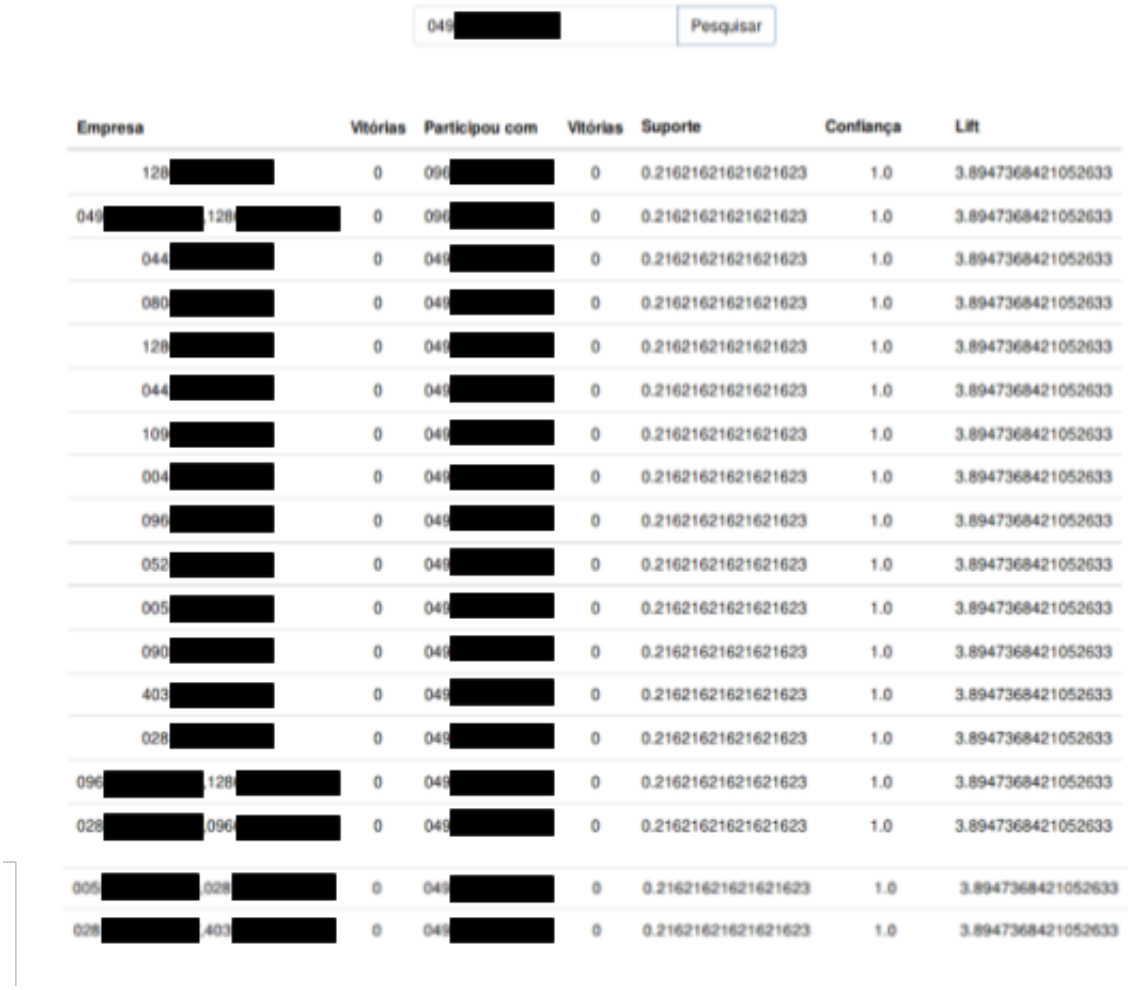

Figura 3.6: Regras utilizadas para buscar vínculos em sistema da CGU.

de uma das empresas tem o mesmo sobrenome de um dos sócios da outra empresa, o que pode configurar em fraude.

O sistema de regras por si próprio permite ampliar o escopo de análise em relação a um CNPJ alvo, mas em conjunto com a análise de vínculos ambos podem se complementar. Buscar todos os vínculos, ou menores caminhos, par a par em uma base de mais de 500 milhões de relacionamentos seria computacionalmente custoso e apenas algumas relações realmente seriam relevantes. Entretanto, em conjunto com o sistema de regras, o mesmo pode indicar quais conjuntos de empresas são mais interessantes de se avaliar com relação aos seus relacionamentos.

A Figura 3.7 foi cedida pela CGU e a baixa resolução é devido ao sigilo dos dados das empresas envolvidas. São representadas na Figura 3.7 três empresas sendo que duas dessas empresas estão conectadas por um vértice e a outra, que é o CNPJ alvo no sistema de regras, não apresenta relação com as demais.

No Capítulo 4 serão discutidos os resultados alcançados neste trabalho.

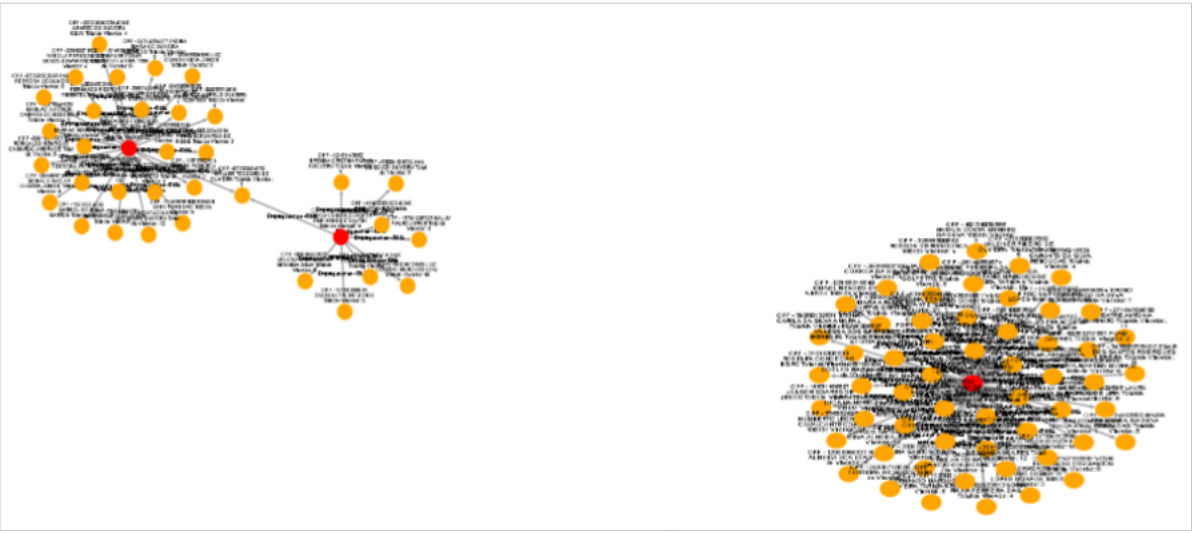

Figura 3.7: Vínculo encontrado entre as empresas utilizando sistema da CGU.

Capítulo 4

Conclusões

Uma das responsabilidades da CGU é combater a corrupção e para isso, é necessário considerar fraudes como conluios e cartéis. Esse trabalho propõe a utilização da mineração de dados nos dados das licitações públicas a fim de encontrar relacionamento entre as empresas participantes.

Para descobrir os relacionamentos entre as empresas no processos licitatórios do Governo Federal, os auditores da CGU teriam que analisar manualmente todos os dados de licitações disponíveis no Portal ComprasNet, o que torna essa tarefa inviável. Desta forma, para auxiliar o trabalho dos auditores na descoberta de relacionamentos das empresas em processos licitatórios esse trabalho foi desenvolvido. O auditor ao tomar ciência desse relacionamento é possível verificar se há outros indícios de um relacionamento fraudulento utilizando outro sistema da CGU que faz a verificação de vínculos entre empresas.

Nesse momento é possível responder a pergunta de pesquisa que é: será que o uso de técnicas de mineração de dados no portal ComprasNet pode auxiliar os auditores da CGU na detecção de fraudes, como conluios ou cartéis, utilizando correlação de empresas participantes em processos licitatórios do Governo Federal Brasileiro? Com esse trabalho foi possível concluir que a mineração de dados pode trazer informações relevantes sobre o relacionamento de empresas em licitações públicas, informação que pode ser utilizada pelos auditores para o combate à corrupção.

Como evolução do trabalho já desenvolvido, propõe-se a utilização de outros algoritmos de mineração de dados para buscar melhores resultados, como a utilização de clusterização para procurar associação de empresas dentro das regiões do país. Otimizar o algoritmo e as buscas no banco para diminuir o tempo de espera do auditor também seria uma evolução plausível. Outro trabalho futuro seria a evolução na apresentação das regras resultantes, como por exemplo a ordenação por relevância.

Referências

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 12

Brasil (1993). Lei nº 8.666, de 21 de junho de 1993. Disponível em: http://www.planalto.gov.br/ccivil_03/leis/L8666cons.html. Acessado em: 07/11/2017. 13

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R. (2000). Crisp-dm 1.0 step-by-step data mining guide. Disponível em: https://www.the-modeling-agency.com/crisp-dm.pdf. Acessado em: 07/11/2017. ix, 7, 8

David J. Hand, H. M. and Smyth, P. (2001). *Principles of Data Mining*. The MIT Press. 7

Erven, G. C. G. V. (2015). MDG-NoSQL: Modelo de Dados para Bancos NoSQL Baseados em Grafos. Master's thesis, Universidade de Brasília, Departamento de Ciência da Coputação. ix, 17, 18

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining*. American Association for Artifcial Intelligence. ix, 7

Han, J. and Kamber, M. (2005). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, Inc. ix, 3, 5, 6, 9, 10, 11, 12

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc. 9

Minović, M., Milovanović, M., Štavljanin, V., Drašković, B., and Lazić, (2014). Semantic technologies on the mission: Preventing corruption in public procurement. 17

Nelson Hein, Fernanda Kreuzber, L. D. e. M. V. (2014). Aplicação da análise fatorial como ferramenta de data mining no desempenho social das empresas do setor de consumo cíclico listadas na bm&fbovespa. 18

Nguyen, T., Zhou, L., Spiegler, V., Ieromonachou, P., and Lin, Y. (2017). Big data analytics in supply chain management: A state-of-the-art literature review. *Computers & Operations Research*. 18

OECD (2016). Preventing corruption in public procurement. Disponível em: http://www.oecd.org/gov/ethics/Corruption-in-Public-Procuremet-Brochure.pdf. Acessado em: 07/11/2017. ix, 15, 16

Ralha, C. G. and Silva, C. V. S. (2012). A multi-agent data mining system for cartel detection in brazilian government procurement. *Expert Syst. Appl.*, 39(14):11642–11656. ix, 1, 16, 17, 23

Russell, S. J. and Norvig, P. (2010). *Artificial Intelligence: a modern approach*. Prentice Hall, 3nd edition. ix, 3, 4

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition. ix, 1, 10

Apêndice A

A.1 Código Python

```
from Apriori import *
import os, os.path
import cherrypy
import rpy2.robjects as ro
import rpy2.robjects.packages as rpackages
from rpy2.robjects.packages import importr
import mysql.connector
from Banco import *


class ApprioriApp(object):
        @cherrypy.expose
        def index(self):
                return open('view.html')

        def compras(self):
                return open('verifica_vencendor.html')

@cherrypy.expose
class AprioriCNPJ(object):
        @cherrypy.tools.json_out()
        def GET(self, cnpj):
                cnpjs = Banco().searchCNPJS(cnpj)
                regras = Apriori().extractRules(cnpjs)
                cnpjs = Banco().extractCNPJs(regras)
                return Banco().formatCNPJS(regras, cnpj)

@cherrypy.expose
class CompraPorCNPJ(object):
        @cherrypy.tools.json_out()
        def GET(self, idcompra):
                compras_cnpj = Banco().searchCompras(idcompra)
```

```
                return idcompra


if __name__ == '__main__':
        conf = {
                '/': {
                        'tools.sessions.on': True,
                        'tools.staticdir.debug': True,
                        'tools.staticdir.root': os.path.dirname(
                        os.path.abspath(__file__))
                },
                '/cnpj': {
                        'request.dispatch':
                        cherrypy.dispatch.MethodDispatcher(),
                        'tools.response_headers.on':
                        True,
                        'tools.response_headers.headers':
                        [('Content-Type', 'application/json')],
                },
                '/compras': {
                        'request.dispatch':
                        cherrypy.dispatch.MethodDispatcher(),
                        'tools.response_headers.on':
                        True,
                        'tools.response_headers.headers':
                        [('Content-Type', 'application/json')],
                },
                '/css': {
          'tools.staticdir.on':
          True,
          'tools.staticdir.dir': os.path.join(os.path.dirname(
          os.path.abspath(__file__)), 'css/')
        }
        }
        webapp = AprioriApp()
        webapp.cnpj = AprioriCNPJ()
        webapp.compras = CompraPorCNPJ()
        cherrypy.quickstart(webapp, '/', conf)
```

A.2 Componente de Banco

```
import mysql.connector
from pandas import *
import rpy2
```

```
class Banco(object):
        def searchCNPJS(self,cnpj):
                try:
                     conn = mysql.connector.connect(host="localhost",
                                                    user="root",
                                                    passwd="root",
                                                    db="mydb")
                except:
                    print("I am unable to connect to the database")

                dfsql = read_sql('select iditemcompra, cnpj from'+
                'licitacoes where iditemcompra in'+
                '(select iditemcompra from licitacoes where cnpj = '
                + cnpj + ')  limit 50;', conn)
                conn.close()
                return dfsql

        def formatCNPJS(self, regras, cnpj):
                        dados = []
                        antes_da_seta = []
                        depois_da_seta = []
                        suporte = []
                        confianca = []
                        lift = []
                        if(type(regras) == rpy2.rinterface.RNULLType):
                                return dados
                        cnpjs = self.extractCNPJs(regras)
                        for label in regras[3]:
                                suporte.append(label)

                        for label in regras[4]:
                                confianca.append(label)

                        for label in regras[5]:
                                lift.append(label)

                        for i in range(0, len(cnpjs)):
                                cnpjs[i].append(cnpj)
                                vitorias = Banco().contaVitorias(cnpjs[i])
                                print(vitorias)
                                dados.append({
                                        'cnpj_1': cnpjs[i][0],
                                        'cnpj_2': cnpjs[i][1],
                                        'vitorias_1':
                                        vitorias[cnpjs[i][0]],
                                        'vitorias_2':
```

```
                                        vitorias[cnpjs[i][1]],
                                        'suporte': str(suporte[1]),
                                        'confianca': str(confianca[1]),
                                        'lift': str(lift[1])
                                    })
                    return dados

    def extractCNPJs(self, regras):
            dados = []
            antes_da_seta = []
            depois_da_seta = []
            for label in regras[0].iter_labels():
                            antes_da_seta.append(label)

            for label in regras[2].iter_labels():
                    depois_da_seta.append(label)

            for i in range(0, len(antes_da_seta)):
                            value1 = antes_da_seta[i].
                            replace("{", "").replace("}","")
                            value2 = depois_da_seta[i].
                            replace("{", "").replace("}","")
                            dados.append([value1, value2])
            return dados

    def contaVitorias(self, cnpjs):
            try:
                 conn = mysql.connector.connect(host="localhost",
                                                user="root",
                                                passwd="root",
                                                db="mydb")
            except:
                print("I am unable to connect to the database")

            df = read_sql('select A.cnpj as cnpj1,'+
            'A.valorpreco as preco1, B.cnpj as cnpj2,'+
            'B.valorpreco as preco2, C.cnpj as cnpj3, '+
            'C.valorpreco as preco3 from licitacoes '+
            'as A  INNER JOIN licitacoes as B ON '+
            'A.iditemcompra = B.iditemcompra INNER JOIN '+
            'licitacoes as C ON B.iditemcompra = C.iditemcompra ' +
            'WHERE A.cnpj ='+cnpjs[0]+' and B.cnpj ='
            +cnpjs[1]+' and C.cnpj = '+cnpjs[2]+' ' +
            'and (A.valorpreco != \',0000\''+
            ' or B.valorpreco != \',0000\' or '+
            'C.valorpreco != \',0000\')', conn)
```

```
        conn.close()

        vitorias = {}
        vitorias[cnpjs[0]] = len(df[df['preco1'] != ',0000'])
        vitorias[cnpjs[1]] = len(df[df['preco2'] != ',0000'])
        vitorias[cnpjs[2]] = len(df[df['preco3'] != ',0000'])

        return vitorias
```

A.3 Componente Apriori

```
from numpy import *
import scipy as sp
from pandas import *
import rpy2
from rpy2.robjects.packages import importr
import rpy2.robjects as robjects
from rpy2.robjects import r, pandas2ri
import rpy2.robjects.packages as rpackages
from rpy2.robjects.packages import importr
import psycopg2
pandas2ri.activate()

class Apriori(object):

    def extractRules(self, cnpjs):
        robjects.globalenv['itens'] = cnpjs
        cnpjs.head()

        # import R's "packages
        base = importr('base')
        utils = importr('utils')
        arules = importr('arules')

        # using R
        r_split = robjects.r['split']
        r_list = robjects.r['list']
        r_as = robjects.r['as']
        r_apriori = robjects.r['apriori']
        r_inspect = robjects.r['inspect']
        splitdf = r_split(cnpjs.cnpj, cnpjs.iditemcompra)

        # apply apriori
        trans = r_as(splitdf, "transactions")
        rules = r_apriori(trans, parameter = r_list
```

```
(minlen = 2,supp = 0.2, conf = 0.98, target = "rules"))
regras = r_inspect(robjects.r["head"]
(robjects.r["sort"](rules, by="lift"),3));
return regras
```

A.4 Visualização

```
<!DOCTYPE html>
<html>
<head>
        <link href="css/bootstrap.css" rel="stylesheet">
        <link href="css/style.css" rel="stylesheet">
        <meta charset="UTF-8">
        <title></title>
</head>
<style type="text/css">
.ajax-loader {
  visibility: hidden;
  background-color: rgba(255,255,255,0.7);
  position: absolute;
  z-index: +100 !important;
  width: 100%;
  height:100%;
}
.ajax-loader img {
  position: relative;
  top:15%;
  left:50%;
}
#texto {
  position: relative;
  top:15%;
  left:35%;
  font-weight: bold;
}
#coluna-img{
        display: block;
}
</style>
<body >
        <div class="container">
            <div class="row">
                <div class="input-group col-md-4  col-md-offset-4">
                        <input type="text" name="cnpj"
                        class="form-control" />
```

```
                        <span class="input-group-btn">
                            <button id="get-cnpj"
                            class="btn btn-primary">
                            Pesquisar</button>
                        </span>
    </div>
<div class="col-lg-4 col-lg-offset-3">
      <table id="result" class="table">
            <thead><tr>
                                <th>
                                        Empresa
                                </th>
                                <th>
                                        Vitorias
                                </th>
                                <th>
                                <div style="width: 110px">
                                        Participou com
                                </div>
                                </th>
                                <th>
                                        Vitorias
                                </th>
                                <th class="coluna-img">
                                <div style="width: 100px;">
                                <span>Suporte</span>
                                <img src="css/idea.png"
                                height="20" width="20"
                                data-toggle="tooltip"
                                title="Chance da regra aparecer "+
                                "novamente na base de dados "+
                                "analisada."/    >
                                </div>
                                </th>
                                <th >
                                <div style="width: 100px;">
                                Confianca <img src="css/idea.png"
                                height="20"width="20"
                                data-toggle="tooltip"
                                title="Numero de  vezes que a"+
                                "regra apareceu corretamente , "+
                                "quanto maior a confianca , "+
                                "melhor e a qualidade da regra.">
                                </div>
                                </th>
                                <th >
```

```
                                        <div style="width: 100px;">
                                        Lift <img src="css/idea.png"
                                        height="20" width="20"
                                        data-toggle="tooltip"
                                        title="Medida de desempenho"+
                                        "da regra.">
                                        </div>
                                        </th>
                                    </tr>
                                </thead>
                                <tbody>
                                </tbody>
                            </table>
                            <div id="texto">
                            </div>
                    </div>
                        </div>
                    </div>
                    <div class="ajax-loader">
                      <img src="css/ajax-loader.gif" class="img-responsive" />
                    </div>
                    <script src="https://code.jquery.com/jquery-3.2.1.min.js"
                    integrity="sha256-hwg4gsxgFZhOsEEamdOYGBf13FyQuiTwlAQgxVSNgt4="
                    crossorigin="anonymous">
                    $(document).ready(function(){
                    $('[data-toggle="tooltip"]').tooltip();
                    });
                    </script>
                <script type="text/javascript">
                    $(document).ready(function() {
                            $("#get-cnpj").click(function(e) {
                                    var cnpj = $("input[name=cnpj]").val()
                                    $.ajax({
                            type: "GET",
                            beforeSend: function(){
                                            $('.ajax-loader').
                                            css("visibility", "visible");
                                            },
                            url: '/cnpj?cnpj=' + cnpj,
                            dataType: 'json',
                            complete: function(){
                                                $('.ajax-loader').
                                                css("visibility", "hidden");
                                            },
                            success: function(data) {
                                            console.log(data);
```

```
                $('#result tbody').empty();
                if(data == null||
                        data == undefined ||
                        data.length == 0){
                        $('#texto').
                        append("<br/> Nao ha regras!")
            } else{
            for(var i =0; i<data.length; i++){
                $('#result tbody')
                        .append(
                            "<tr>"+
                                "<td>" +
                                data[i].cnpj_1 +
                                "</td>" +
                            "<td>" +
                            data[i].vitorias_1 +
                            "</td> " +
                            "<td>" +
                            data[i].cnpj_2 +
                            " </td>" +
                            "<td>" +
                            data[i].vitorias_2 +
                            " </td>" +
                            "<td>" +
                            data[i].suporte +
                            "</td>" +
                            "<td>" +
                            data[i].confianca +
                            "</td>" +
                            "<td>" +
                            data[i].lift +
                            "</td></tr>")}
                }
                    }
                })
        e.preventDefault();
        });
        })
    </script>
</body>
</html>
```