

UNIVERSIDADE FEEVALE

CLEBER RODRIGO DE MORAES

ANÁLISE DE INCIDENTES DE UMA CENTRAL DE SERVIÇOS DE TI,
BASEADA EM ITIL, APLICANDO TÉCNICAS DE DATA MINING PARA
TOMADA DE DECISÃO ESTRATÉGICA

Novo Hamburgo

2014

CLEBER RODRIGO DE MORAES

ANÁLISE DE INCIDENTES DE UMA CENTRAL DE SERVIÇOS DE
TI, BASEADA EM ITIL, APLICANDO TÉCNICAS DE DATA MINING
PARA TOMADA DE DECISÃO ESTRATÉGICA

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do grau de Bacharel em
Sistemas de Informação pela
Universidade Feevale

Orientador: Juliano Varella de Carvalho

Novo Hamburgo

2014

AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, à minha família, em especial à minha mãe pelas lições de vida e por nos ensinar a lutar pelos sonhos.

À minha esposa Bruna pelo amor, carinho, dedicação e apoio incondicional em todos os momentos.

Aos meus sogros, que são a família que escolhi, pelas orientações e pela torcida para que nossos sonhos se realizassem.

Aos meus amigos que, nas jantãs e esportes me animavam após longas horas de estudo.

Ao meu gestor Stewart Unger pelo incentivo, aposta e orientações!

RESUMO

Com o crescimento exponencial das informações ao longo do tempo, tem sido cada vez mais difícil analisá-las de modo que se possa extrair conhecimento relevante para tomada de decisões. Este crescimento no volume de informações torna a análise dos registros através de planilhas no Microsoft Excel inviável, o que faz com que o time de gerenciamento de incidentes do *IT Global Service Desk* (ou Central de Serviços Global de TI) foque principalmente nos problemas mais recorrentes e que possuem maior impacto no negócio. Esta Central de Serviços foi implantada baseando-se nas melhores práticas da ITIL (*Information Technology Infrastructure Library*) para o gerenciamento de uma infraestrutura de TI. O presente trabalho analisa os incidentes do *Global Service Desk* de uma companhia multinacional. Utilizando técnicas de *Data Mining* sobre os registros de incidentes, foram extraídas oportunidades de resolução de incidentes no primeiro nível de atendimento, os quais o *Service Desk* seria capaz de resolver e desta forma garantir fatores críticos de sucesso para a organização. As oportunidades encontradas representaram até 25% de redução de custo no suporte de TI.

Palavras-chave: Mineração de dados; Gerenciamento de Incidentes; ITIL; Global Service Desk; Central de Serviços de TI.

ABSTRACT

With the exponential growth of information along time, it has been increasingly difficult to analyse this large amount of information in order to be possible to extract useful knowledge for decision-making. As the amount of data is too large, it becomes unviable to analyse it through Microsoft Excel spreadsheets, which leads the Incident Management team of the Global IT Service Desk to keep the focus only in the problems that are recurrent and have a higher impact on the business. This study has analysed the incidents of the Global Service Desk of a multinational company. This Service Desk was deployed based on the ITIL best practices for managing IT infrastructure. Using data mining techniques on incident records were extracted opportunities from which Service Desk analyst could have been resolved at first level, enabling to ensure critical success factors to the business. These opportunities founds up to 25% of opportunities for cost reduction in the incident management.

Key words: Data mining, Incident Management, ITIL, Global Service Desk, IT Service Desk.

LISTA DE FIGURAS

Figura 1.1 - Cadeia de Valor dos Serviços	13
Figura 1.2 – Processo de atendimento de um incidente	15
Figura 2.1 - Interação do <i>Service Desk</i> com as demais áreas da TI	19
Figura 2.2 – Arquitetura local de uma Central de Serviços.....	23
Figura 3.1 – Evolução de um erro até a implementação da solução	26
Figura 4.1 - Visão geral dos processos que envolvem a descoberta de conhecimento	32
Figura 4.2 - Classificando a prioridade de um incidente.....	35
Figura 4.3 - Algoritmo de Hunt para induzir árvores de decisão	38
Figura 4.4 - Agrupamento de objetos utilizando o algoritmo <i>k-means</i>	42
Figura 4.5 - Processo típico de um projeto de mineração de dados suportado pelo Rattle	48
Figura 4.6 – Tela inicial do “RGui (64-bit).....	49
Figura 4.7 – Tela inicial do Rattle	49
Figura 4.8 – Rattle – Tela de seleção de variáveis	50
Figura 4.9 – Tela de exploração de dados	50
Figura 4.10 – Tela de transformação dos dados	51
Figura 4.11 – Tela de construção de modelos	51
Figura 4.12 – Tela de avaliação de modelos	52
Figura 5.1 – Captura de tela da aplicação do <i>Service Desk</i>	54
Figura 5.2 - Níveis de serviço e o valor (estratégia <i>shift right-to-left</i>).....	57
Figura 5.3 – Correlação de valores ausentes utilizando o algoritmo de Pearson	64
Figura 5.4 - Árvore de decisão Ctree da variável Total Resolved Time(hrs).....	67
Figura 5.5 - Distribuição de frequências antes e depois do ajuste dos limites de TTTR_Cat..	69
Figura 5.6 – Comparação da distribuição de frequências do atributo TTTR_Cat antes e depois do ajuste.....	69
Figura 5.7 – Distribuição de frequências dos atributos discretizados	70
Figura 5.8 – Grupos gerados com o algoritmo K-Means exibidos como coordenadas discriminantes.....	74
Figura 5.9 – Proporção de TTTR para os incidentes FTF, FCR, NO e All.....	76
Figura 5.10 – Árvore de inferência condicional: TTTR para FCR.FTF.....	77
Figura 5.11 – Árvore de inferência condicional: CallTT_Cat, Priority e TTTR para FTF.FCR	77

Figura 5.12 – Árvore de inferência condicional – RG.SD, CallTT e TTTR para FCR	78
Figura 5.13 – Ilustração da estratégia <i>shift right-to-left</i>	87

LISTA DE TABELAS

Tabela 4.1- Conjunto de treino para prever tomadores de empréstimos que ficarão inadimplentes.....	37
Tabela 5.1 – Valores das variáveis excluídas	61
Tabela 5.2 – Resumo de valores ausentes obtidos através da função Mice	63
Tabela 5.3 – Valores utilizados para discretização dos atributos	65
Tabela 5.4 – Condições utilizadas para discretização do atributo FCR.FTF	66
Tabela 5.5 – Valores de <i>Total Resolved Time(hrs)</i> discretizados para TTTR_Cat	68
Tabela 5.6 – Descrição dos atributos categóricos.....	72
Tabela 5.7 - Descrição dos atributos numéricos.....	72
Tabela 5.8 – Características de cobertura e confiança das regras de associação.....	80
Tabela 5.9 - Regras de associação selecionadas.....	81

LISTA DE QUADROS

Quadro 2.1 - Tipos de arquitetura de Centrais de Serviço.....	22
Quadro 2.2 – Indicadores Chave de Desempenho de um <i>Service Desk</i>	24
Quadro 3.1 - Indicadores chave de desempenho de gerenciamento de incidentes.....	29
Quadro 4.1 - Algoritmo K-means básico	42
Quadro 5.1 - Atributos selecionadas para o processamento inicial.....	58
Quadro 5.2 – Resumo do modelo de classificação árvore de decisão Ctree para o atributo <i>Total Resolved Time(hrs)</i>	68
Quadro 5.3 – Combinação de atributos e seus valores para potenciais oportunidades de FTF	80
Quadro 5.4 – Resumo de tipo de oportunidades de melhoria por regra	85

LISTA DE ABREVIATURAS E SIGLAS

ANS	Acordo de Nível de Serviço
BI	Business Intelligence
CallTT	Atributo: <i>Call Total Transfers</i> (Número de transferências entre equipes)
CallTT_Cat	Atributo categórico: <i>Call Total Transfers Category</i>
FTF	Atributo e métrica de ITIL: <i>First Time Fix</i>
FCR	Atributo e métrica de ITIL: <i>First Call Resolution</i> (conhecido também como <i>First Level Resolution - FLR</i>)
FTF.FCR	Atributo: Indica se o incidente é FTF, FCR ou NO (nenhum dos anteriores)
GUI	<i>Graphics User Interface</i> (Interface Gráfica do Usuário)
ID3	<i>Iterative Dichotomiser</i>
IMF	<i>Incident Management Feedback</i> (Índice de Satisfação do Usuário)
ITIL	<i>Information Technology Infrastructure Library</i>
JUSE	União dos Engenheiros e Cientistas Japoneses
KDD	<i>Knowledge Discovery Database</i>
KPI	<i>Key Performance Indicator</i> (Indicador de desempenho chave)
RG	<i>Resolver Group</i>
SLA	<i>Service Level Agreement</i>
TI	Tecnologia da Informação
TTTR	Métrica de ITIL = <i>Total Time to Resolve</i> (hrs) = Atributo <i>Total Resolved Time</i> (hrs) em horas
TTTR_Cat	Atributo = <i>Total Time to Resolve</i> (hrs) <i>Category</i>

SUMÁRIO

1	INTRODUÇÃO	13
2	CENTRAL DE SERVIÇOS DE TI (SERVICE DESK)	18
2.1	Qualidade e Melhoria dos Serviços de TI	20
2.2	Tipos de Centrais de Serviço	20
2.3	Arquitetura da Central de Serviços	22
2.4	Indicadores-chave de desempenho	23
3	GERENCIAMENTO DE INCIDENTES BASEADO EM ITIL	25
3.1	processo de Gerenciamento de Incidentes	25
3.1.1	Ciclo de vida de um incidente no <i>Service Desk</i> baseado em ITIL	26
3.1.2	Indicadores Chave de Desempenho (KPIs)	29
4	MINERAÇÃO DE DADOS	31
4.1	KDD (Knowledge Discovery in Databases)	32
4.2	Técnicas	33
4.2.1	Classificação	34
4.2.1.1	Árvores de Decisão	35
4.2.2	Agrupamento	38
4.2.2.1	Diferentes tipos de agrupamentos	40
4.2.2.2	Algoritmo K-Means	41
4.2.3	Regras de associação	42
4.2.3.1	Gerando regras com eficiência	43
4.2.3.2	Representando as regras	44
4.2.4	WEKA	45
4.2.5	SPSS	45
4.2.6	Linguagem R	46
4.2.6.1	Rattle	47
5	ANÁLISE DE INCIDENTES DE UMA CENTRAL DE SERVIÇOS DE TI	54
5.1	Compreensão do escopo	55
5.2	Etapas de seleção	57
5.3	Pré-processamento	59
5.3.1	Exclusões	59
5.3.2	Valores ausentes	62

5.4	Transformação de atributos	64
5.4.1	Transformação da variável <i>Total Resolved Time(hrs)</i>	66
5.4.2	Descrição dos dados	70
5.5	Aplicando técnicas de mineração de dados	72
5.5.1	Escolha das tarefas e algoritmos	73
5.5.2	Árvores de decisão	74
5.5.3	Regras de associação	79
5.5.3.1	Análise das regras de associação obtidas	81
5.6	Avaliação dos resultados	85
5.6.1	Redução de custos	86
6	CONCLUSÃO	90
	REFERÊNCIAS BIBLIOGRÁFICAS	92
	APÊNDICE A - DESCRIÇÃO DOS CAMPOS DISPONÍVEIS DA BASE DE DADOS DE INCIDENTES	95
	APÊNDICE B – REGRAS DE ASSOCIAÇÃO GERADAS PELO ALGORITMO APRIORI	98

1 INTRODUÇÃO

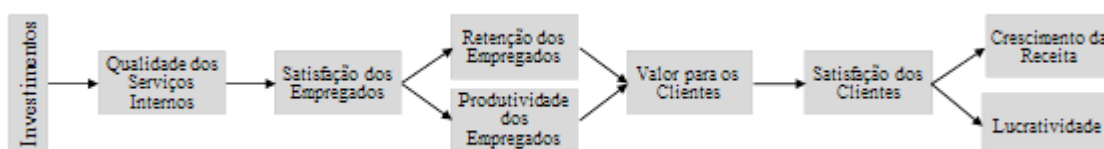
A presente pesquisa foi desenvolvida tomando-se como base o registro de incidentes de uma companhia multinacional do ramo alimentício que possui cerca de 150 unidades de operação no Brasil, sendo a sede localizada no estado do Rio de Janeiro. É integrante de um grupo de companhias multinacionais do mesmo ramo, com mais de 100 anos de existência e atuação em mais de 180 países.

Do ponto de vista da infraestrutura de TI, somente no Brasil são aproximadamente 100 aplicações e mais de 4000 estações de trabalho, além de telefones, celulares e *smartphones*. Os sistemas de comunicação e a plataforma de servidores estão distribuídos em mais de 100 unidades de operação e negócios, como também interconectadas através de *links* de internet internacionais a *data centers* situados estrategicamente em Cachoeirinha (Brasil), Frankfurt (Alemanha) e Kuala Lumpur (Malásia). Nas Américas, são mais de 10 mil computadores espalhados em 26 dos 36 países nos quais a companhia possui filiais. Entre os principais sistemas de informação está o sistema integrado de gestão SAP, sistemas de venda, faturamento, comunicação e colaboração.

Diante deste complexo ambiente, verificou-se a necessidade de implantação de uma central de serviços de TI (Tecnologia da Informação), e há aproximadamente 2 anos foi implantado o *Service Desk*, em um ambiente virtualizado e centralizado na Costa Rica, baseado nos guias de melhores práticas de ITIL (*Information Technology Infrastructure Library*).

O objetivo do *Service Desk* é elevar a qualidade dos serviços internos. Desta forma, a eficiência, a eficácia, a efetividade e a economicidade no suporte aos serviços de TI e aos seus usuários são fatores críticos para o sucesso no alcance dos objetivos estratégicos traçados pela organização. A Cadeia de Valor dos Serviços é ilustrada na Figura 1.1, conforme (MAGALHÃES, 2007, p. 109).

Figura 1.1 - Cadeia de Valor dos Serviços



Fonte: Magalhães (2007)

Segundo Magalhães (2007, p. 111), o *Service Desk*

[...]estende a gama de serviços e oferece uma abordagem ao mesmo tempo global, por ser a única porta de entrada, focada, pela especialização nos diferentes tipos de atendimento, permitindo que os processos de negócio sejam integrados aos processos que compõem o gerenciamento dos Serviços de TI. Não trata apenas de atender aos incidentes, problemas e consultas, mas também de prover uma interface para outras atividades relacionadas com as demais necessidades dos usuários e clientes dos serviços de TI, como requisições de mudança, contratos de manutenção, licenciamento de produtos de software, solicitações de serviços, reclamações sobre divergências nas faturas de serviços, cronograma de manutenções preventivas e mudanças a serem realizadas na infraestrutura de TI, orientações em caso de desastres etc.

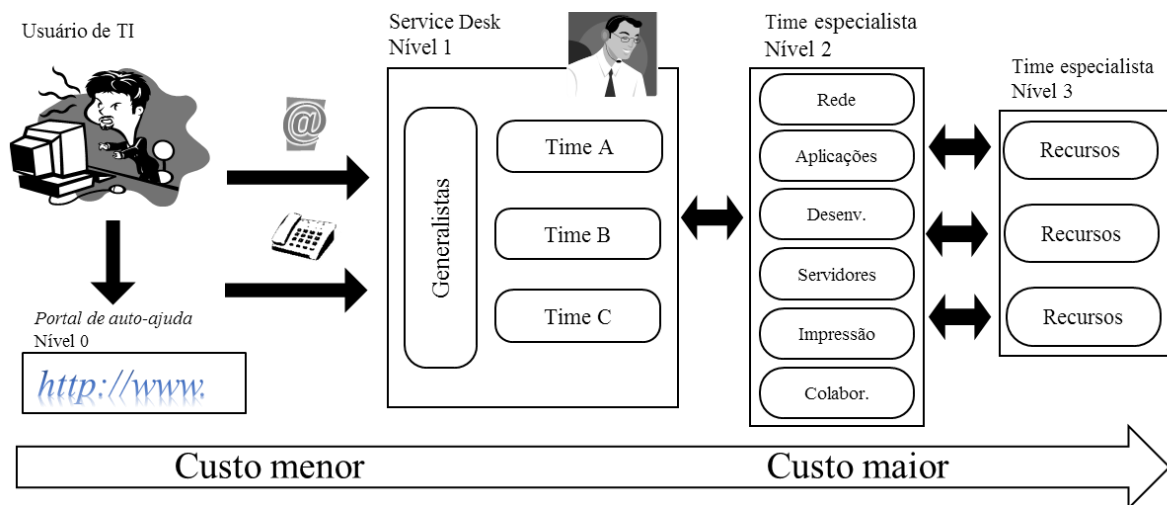
As tarefas normalmente atribuídas ao *Service Desk* são:

- Receber e registrar todas as chamadas dos usuários dos serviços de TI;
- Lidar diretamente com pedidos e reclamações simples;
- Prover uma avaliação inicial de todos os incidentes comunicados;
- Encaminhar para o suporte técnico de segundo nível os incidentes não-solucionados conforme os níveis de serviço estabelecidos;
- Manter os usuários dos serviços de TI informados sobre o estado atual dos incidentes comunicados e da evolução do atendimento;
- Produzir informações gerenciais, coletando dados e calculando indicadores de desempenho.

Tendo em vista a complexidade do gerenciamento de um ambiente global, para que se possa garantir um padrão no fornecimento de serviços, a organização empregou as melhores práticas citadas em ITIL. ITIL fornece um guia de melhores práticas para o gerenciamento de serviços de TI como um negócio que envolve pessoas, processos e tecnologia (MAGALHÃES, 2007).

Como a Central de Serviços oferece serviços menos especializados do que os níveis subsequentes, seu custo por contato é menor se comparado ao segundo e terceiro nível de atendimento, conforme detalhado na Figura 1.2.

Figura 1.2 – Processo de atendimento de um incidente



Fonte: Elaborado pelo autor

Destá forma, a transferência de incidentes e procedimentos de suporte para o primeiro nível é um dos desafios constantes para a equipe de gerenciamento de incidentes. Assim, “garantir o encerramento do maior número de incidentes e consultas dentro do seu nível de atendimento (nível 1), evita a sobrecarga das demais equipes que atuam no processo...” (MAGALHÃES, 2007. p. 112).

A definição de incidente, segundo ITIL, “é qualquer evento que não faz parte do funcionamento padrão de um serviço de TI e que causa, ou pode causar, uma interrupção do serviço ou uma redução do seu nível de desempenho. Ou ainda, simplesmente, qualquer evento que reduz a qualidade de um serviço” (ABBOTT e FISHER, 2009, p. 134 – tradução nossa)

Gerenciamento de incidentes é o processo de restaurar um serviço o mais rápido possível quando existe uma interrupção ou redução na qualidade deste. O objetivo do processo é minimizar qualquer impacto negativo no negócio que pode ser causado por um incidente. O gerenciamento de incidentes também é responsável por garantir que a qualidade e disponibilidade do serviço sejam mantidas de acordo com o nível de serviço contratado (SLA). (OGC, 2007, tradução nossa)

A partir deste complexo ambiente, muitas informações gerenciais são geradas, demandando cada vez mais tempo da equipe de gestores na aquisição de informações para tomada de decisões. O *Service Desk* que, por exemplo, no ano de 2013, registrou para os usuários do Brasil, 53135 incidentes, demonstra ser um ambiente crítico caso um ambiente padronizado não seja adotado. Em 2014, de janeiro a julho, foram mais de 55 mil incidentes registrados, aumento este, em virtude de que novos mercados migraram para este modelo de atendimento centralizado. O grande problema diante disto é analisar o grande volume de dados

sobre os incidentes para garantir que as métricas e indicadores de eficiência do *Service Desk* e satisfação dos usuários sejam mantidas e melhoradas.

Todos estes incidentes, também conhecidos como *tickets*, são registrados em um software – fabricado pela *Computer Associates* – específico para gerenciamento de *Service Desk* e armazenados em uma base de dados centralizada. A partir desta base de dados, podem ser executados e extraídos alguns relatórios básicos para análise gerencial como, por exemplo, incidentes por região, equipe, prioridade, *status*, etc.

Frequentemente são necessárias análises de incidentes dos últimos 12 meses, por exemplo, para investigar algum problema específico. Um grande volume de dados como este, permitiria somente análises mensais ou divididas em regiões. Nem sempre exportar e trabalhar com planilhas no *Microsoft Excel* é algo viável, pois demanda muito tempo de análise e processamento. Isto tem como consequência uma análise superficial dos problemas, levando a investigações de problemas com maior reincidência ou impacto sobre o negócio, excluindo, muitas vezes, análises proativas ou mais abrangentes a respeito do ambiente.

Buscando-se alternativas para uma investigação mais acurada, observou-se que a mineração de dados pode ser utilizada para descobrir informações relevantes e inesperadas em grandes volumes de dados utilizando algoritmos que aplicam técnicas como associação, classificação, agrupamento e previsão. (PICCOLI, 2012). A partir disto, fez-se uma pesquisa a respeito de linguagens de programação aplicadas à mineração de dados, que fossem dinâmicas e ágeis e chegou-se à linguagem R. Esta linguagem é de uso livre e de código aberto, e está disponível para *download* gratuitamente. Suas funcionalidades e o conjunto diversificado de pacotes adicionais disponíveis fazem desta ferramenta uma excelente (e barata) alternativa diante de várias (e caras) ferramentas de mineração de dados para computação estatística e gráfica existentes (TORGO, 2010, p. 3).

A partir disto, foram executados todos os passos para a mineração de dados utilizando-se das funcionalidades já conhecidas e implementadas da linguagem R, a fim de produzir e encorajar pesquisadores a produzir novas experiências através desta ferramenta. Além do R, algumas técnicas foram aplicadas utilizando o Weka, para comparação de resultados e testes.

Uma das questões mais importantes que distinguem a mineração de dados é o tamanho das bases de dados analisadas. Isto coloca desafios difíceis para as disciplinas de análise de dados padrão: deve-se considerar questões como a eficiência computacional, os recursos de memória limitada, interfaces com bancos de dados, etc. Aproveitando-se das interfaces de banco de dados disponíveis em R, é possível realizar mineração de dados em grandes volumes de dados (TORGO, 2010, p. 10).

Pensando nisto é que se fundamentou o objetivo desta pesquisa. Utilizar a facilidade de manipulação de grandes volumes de dados da Linguagem R, e as técnicas de análise abordadas pela mineração de dados, a fim de realizar uma análise mais aprofundada de alguns dos principais problemas levantados pela Gerência de Incidentes e também mencionados pelas melhores práticas de ITIL. Desta forma, foi possível fornecer informações, além das métricas-padrão já adotadas, para que o time de gerenciamento de incidentes tenha mais assertividade nas tomadas de decisão.

Este trabalho está organizado em 5 capítulos onde, além desta introdução, apresenta no capítulo 1 uma Central de Serviços de TI, seus conceitos e como é organizada, ambiente no qual a pesquisa está inserida. No capítulo 2 são discutidos os padrões de ITIL nos quais a Central de Serviços é baseada. Já no capítulo 3, são apresentadas as técnicas de mineração de dados que foram estudadas e serão aplicadas para o cumprimento dos objetivos da pesquisa. No capítulo 4, são discutidos os processos para que seja possível chegar aos objetivos estabelecidos, além da apresentação dos dados disponíveis para análise e a forma como eles estão organizados. Finalmente, no capítulo 5, estão descritas as etapas de mineração de dados que foram executadas para que o objetivo fosse cumprido, tais como seleção, pré-processamento, transformação, aplicação das técnicas e avaliação dos resultados. Neste capítulo também se encontra uma estimativa de redução de custos, onde se comparou o ambiente atual com o ambiente baseado nas oportunidades de melhoria que foram encontradas no presente trabalho.

2 CENTRAL DE SERVIÇOS DE TI (SERVICE DESK)

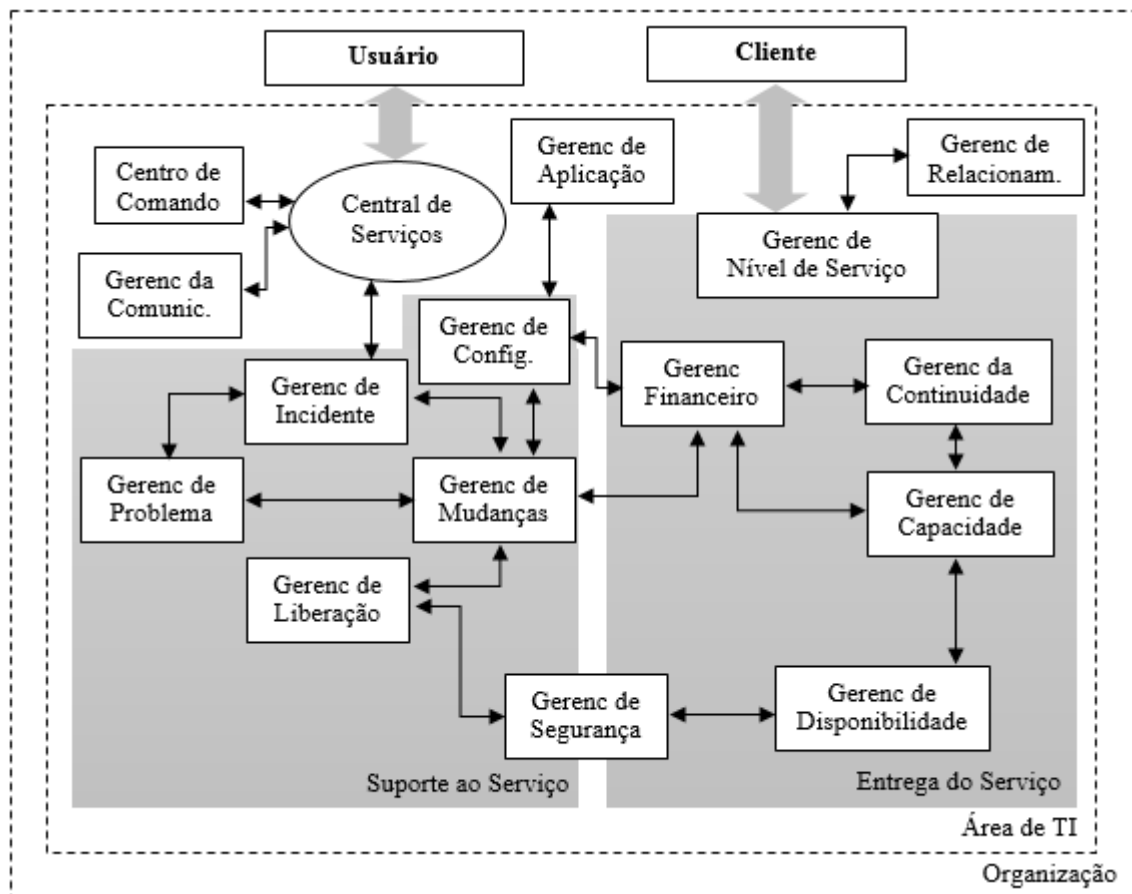
Empresas com milhares de funcionários precisam se preocupar com a manutenção da sua capacidade produtiva. Sendo assim, o potencial dos serviços prestados a partir de uma central passou a ser observado como oportunidade de otimização de processos e dos esforços de gestão para o público interno da organização. Diante deste cenário, a organização precisa se equipar com dispositivos que permitam que processos críticos para a continuidade do negócio estejam alinhados. Isto pode significar o sucesso ou o fracasso de uma organização, ou ainda, passível de ser considerado uma vantagem competitiva para a sua maioria. Com a globalização das organizações e o acirramento da concorrência decorrente disto, tornou-se necessária a entrega de serviços em classe mundial.

Assim surgiu o conceito de Central de Serviços ou, em inglês, *Service Desk*, que visa concentrar o suporte de colaboradores e o público interno da organização em um mesmo lugar, no que diz respeito à TI, e colabora diretamente para a continuidade das atividades.

O *Service Desk* é uma função essencial para a implantação do Gerenciamento dos Serviços de TI, constituindo-se em um ponto de suporte aos usuários dos serviços de TI (*Help Desk*), e sim, uma interface operacional entre a área de TI e os usuários dos seus serviços. Sendo assim, ela também é responsável pela primeira impressão que a área de TI dará aos seus usuários quando houver necessidade de interação.

Como podemos observar na Figura 2.1, o *Service Desk* atua como interface entre a área de TI e os usuários de forma bidirecional. Além do atendimento a chamadas, ela também é responsável pelo envio de notificações de eventos enviadas pela equipe de monitoração de toda a infraestrutura de TI, como também envia informações sobre os incidentes comunicados pelos usuários. No caso da companhia, ela atua como único ponto de contato da organização com a área de TI. Desta forma, em processos padrão, durante o tratamento de incidentes ou requisições feitas pelo usuário, ele não é contactado por fornecedores ou membros da área de TI, a não ser pelo próprio *Service Desk*.

Figura 2.1 - Interação do *Service Desk* com as demais áreas da TI



Fonte: Adaptado de Magalhães (2007, p. 108)

A função do *Service Desk* interage principalmente com o gerenciamento de incidentes, que, executa algumas das atividades deste processo através do recebimento de chamadas originadas de erros enfrentados pelos usuários que utilizam os serviços. Estes, após serem analisados, poderão se constituir incidentes, classificados e atendidos no primeiro nível ou não.

Como benefícios de uma Central de Serviços, pode-se observar:

- A melhoria do suporte técnico aos usuários de TI;
- Melhoria do índice de satisfação dos usuários pelo incremento da percepção de qualidade e profissionalismo;
- Aumento da acessibilidade para os usuários dos serviços de TI, pelo estabelecimento de um único ponto de contato;
- Melhorias no tempo de atendimento;
- Uso eficiente de recursos de suporte técnico;
- Melhor gerenciamento da informação para a tomada de decisão relativa aos serviços de suporte aos usuários de TI.

Neste trabalho será utilizado o termo *Service Desk* para identificar a Central de Serviços de TI, a fim de padronizar a identificação do termo e facilitando seu entendimento.

2.1 QUALIDADE E MELHORIA DOS SERVIÇOS DE TI

Um dos maiores desafios da gestão contemporânea é garantir a qualidade dos produtos e serviços e isso só é possível a partir da experiência em executar processos de produção por diversas vezes. Em um processo como o *Service Desk*, que possui muitas variáveis, esta tarefa pode se tornar ainda mais complexa, o que exige foco e maior esforço na melhoria do resultado das atividades. Os principais objetivos do processo de melhoria continuada exploram a eficiência, efetividade dos processos, pessoas, tecnologias e infraestrutura que compõem cada um dos ativos de serviços entregues aos usuários/clientes (MELENDEZ, 2011).

Tal princípio sobre as principais ações para o aumento do nível de serviço em projetos de *Service Desk*, pode ser complementado como segue, segundo (MADRUGA, 2006):

- Integração entre sistemas e telecomunicações;
- Automatização de parte das chamadas;
- Adequação da infraestrutura às exigências de produtividade e prazer no trabalho;
- Aprimoramento da gestão por parte dos supervisores e gerente;
- Treinamento do pessoal em técnicas e mudança de comportamento;
- Revisão de processos e indicadores recalibrados;
- Melhoria na previsão de chamadas;
- Adequação do quadro de pessoal;
- Informações ágeis e seguras para os agentes;
- Estratégias mercadológicas coerentes.

Com isso, é possível concluir que, cada vez mais, estratégias de melhoria de serviço por constatações práticas ou alinhamento estratégico, são imprescindíveis para que o *Service Desk* consiga cumprir seus objetivos de entrega de serviço com qualidade, eficiência e eficácia.

2.2 TIPOS DE CENTRAIS DE SERVIÇO

Há ainda alguns tipos de Centrais de Serviço encontrados nas bibliografias, cada uma com suas peculiaridades, conforme listadas abaixo, juntamente com um breve resumo dos propósitos, segundo Magalhães (2007):

- Central de Suporte (*Help Desk*): gerenciar, coordenar e dissolver incidentes o mais rápido possível e assegurar que nenhuma requisição de ajuda ficará perdida;
- Centro de Contatos (*Contact Center*): atender grandes volumes de transações baseadas em ligações telefônicas de serviços relacionados aos setores como TI, financeiros, seguros, mídia, etc.
- Central de Serviços (*Service Desk*): visa centralizar a comunicação dos erros, dúvidas e solicitações relacionadas aos serviços de TI disponibilizados pela organização, desta forma, colaborando com a redução do tempo de atendimento e de reparação dos serviços.

Muitas operações de Centrais de Suporte e Centro de Contatos naturalmente evoluem para Centrais de Serviços, ampliando seu leque de serviços prestados aos usuários e, conseqüentemente, ao negócio. Algumas ainda evoluem para Centrais de Serviço Compartilhadas, que aumentam as áreas atendidas onde, além da TI, prestam serviços de Recursos Humanos, Financeiros, Marketing, entre outros. Mas todas elas possuem alguns pontos em comum:

- Representam a TI para o cliente e para o usuário, sejam internos ou externos;
- Operam sob o princípio de satisfação do cliente e do usuário, portanto, precisam zelar pela impressão causada;
- Dependem de pessoas dinâmicas e multifuncionais, processos e tecnologia para entregar serviços à organização.

Ainda, segundo Magalhães (2007), os agentes, como são conhecidos os operadores do *Service Desk*, comunicam-se com usuários de diversos meios, predominando principalmente a comunicação partindo dos usuários, os quais podem estar distribuídos em qualquer lugar do mundo. Considerando este perfil de recepção de chamadas, o esforço é grande para que se garanta um atendimento dentro dos padrões estabelecidos, pois não é possível ter controle sobre o número de chamadas recebidas. Diante deste cenário, outro grande desafio é lidar com as diferentes culturas e línguas de cada local, as quais podem afetar diretamente o índice de satisfação do usuário caso o atendimento seja não cuidadosamente planejado.

Assim, iniciaram-se alguns trabalhos de automatização ou autoatendimento, nos quais se procura substituir o agente humano por um sistema de resposta audível ou reconhecimento de voz. Ou, ainda, sistemas baseados em teclado, com opções pré-definidas nas quais o usuário escolhe as opções disponíveis para a resolução do seu problema (MAGALHÃES, 2007).

2.3 ARQUITETURA DA CENTRAL DE SERVIÇOS

Entre os tipos de Centrais de Serviços, algumas arquiteturas podem ser definidas de acordo com o tamanho da organização, sendo elas: Local, Centralizada ou Virtualizada, conforme o Quadro 2.1.

Quadro 2.1 - Tipos de arquitetura de Centrais de Serviço

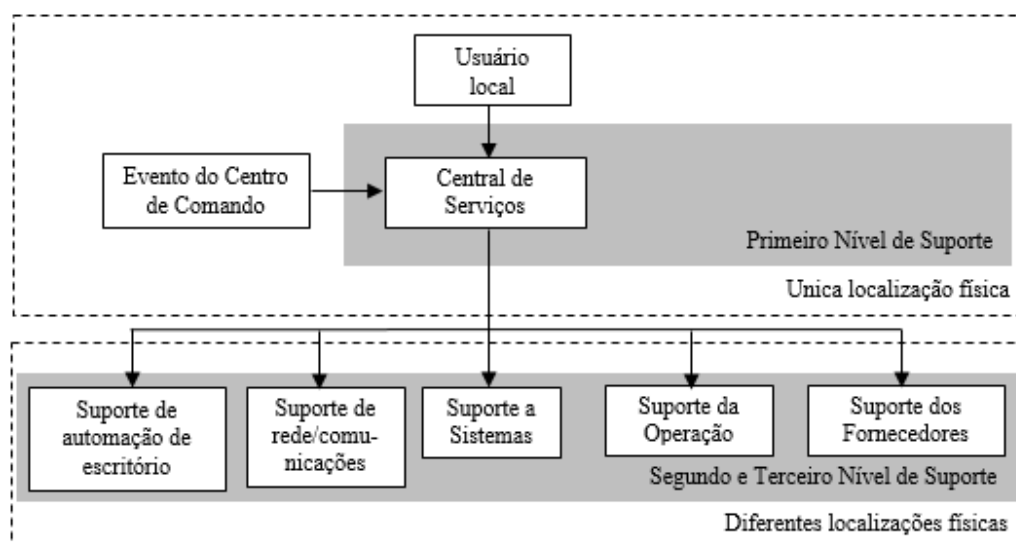
Tamanho da organização	Estrutura organizacional	Heterogeneidade da Infraestrutura de TI	Arquitetura da Central de Serviços
Pequena	Centralizada	Baixa	Local
Pequena	Centralizada	Elevada	Local
Pequena	Distribuída	Baixa	Centralizada
Pequena	Distribuída	Elevada	Centralizada
Grande	Centralizada	Baixa	Local
Grande	Centralizada	Elevada	Centralizada
Grande	Distribuída	Baixa	Centralizada/Virtualizada
Grande	Distribuída	Elevada	Virtualizada

Fonte: Magalhães (2007)

- **Central de Serviços Local**

Uma Central de Serviços seguirá a arquitetura denominada Local quando toda a sua infraestrutura estiver instalada e operando na mesma localização física dos usuários dos serviços de TI, como se pode observar na Figura 2.2. Apesar de a Central de Serviços localizar-se junto aos usuários dos serviços de TI, o suporte técnico que não for do nível 1 poderá ser realizado por equipes situadas em diferentes lugares, concentradas de acordo com a sua especialidade, obtendo-se maior capacidade de atendimento e o acesso mais rápido ao conhecimento específico e a experiência da equipe (MAGALHÃES, 2007, P. 119).

Figura 2.2 – Arquitetura local de uma Central de Serviços



Fonte: Magalhães (2007, p. 120)

- **Central de Serviços Centralizada**

Uma Central de Serviços centralizada é identificada se toda a sua infraestrutura estiver em um local físico diferente da localização dos usuários de TI. (MAGALHÃES, 2007) É a arquitetura mais comumente encontrada nas organizações e a encontrada no ambiente estudado, na qual o *Service Desk* localiza-se fisicamente na Costa Rica e os usuários e demais equipes de suporte (segundo e terceiro nível) localizam-se em outros países.

- **Central de Serviços Virtualizada**

Quando a infraestrutura da Central de Serviços estiver distribuída por diferentes locais físicos, seja em âmbito nacional ou internacional, ela é considerada uma Central de Serviços Virtualizada.

2.4 INDICADORES-CHAVE DE DESEMPENHO

Para que o *Service Desk* possa prover pontos de controle, é necessário avaliar sua eficiência, eficácia, efetividade e economicidade. Estes pontos de controle são conhecidos como Indicadores-Chave de Desempenho (*Key Performance Indicator – KPI*).

No Quadro 2.2 são exibidas as propostas de indicadores-chave modelados a partir da metodologia *Strategic Activity System (SAS)*, para o gerenciamento de *Service Desk* (MAGALHÃES, 2007, p. 130).

Quadro 2.2 – Indicadores Chave de Desempenho de um *Service Desk*

Perspectiva	Indicador
Eficiência	- Índice de chamadas atendidas. - Índice de incidentes encerrados no primeiro nível de suporte.
Eficácia	- Índice de incidentes encerrados dentro do prazo estabelecido. - Índice de disponibilidade da infraestrutura do <i>Service Desk</i> .
Efetividade	- Índice de incidentes encerrados na primeira chamada. - Índice de satisfação dos usuários com o atendimento.
Economicidade	- Índice de incidentes resolvidos de forma remota. - Índice de evolução do custo médio por chamada atendida.

Fonte: Magalhães (2007)

Estes e outros indicadores mencionados no próximo capítulo são monitorados pelo time de operações de TI e, com frequência, são estabelecidas e revisadas as metas para as equipes de suporte. Estas metas são acompanhadas em painéis conhecidos como *dashboards* (“painel de instrumentos” - na tradução direta), espalhados nas áreas comuns do *Service Desk* e operações de TI.

Estas métricas são estabelecidas pela companhia e acompanhadas mensalmente completando um ciclo anual. Ao final deste ciclo, os funcionários das áreas (TI, finanças, produção, centro de pesquisa) que atingem as metas estabelecidas são premiados com um bônus no salário, podendo este ser de até 2,5 vezes o valor do seu salário.

3 GERENCIAMENTO DE INCIDENTES BASEADO EM ITIL

ITIL (*Information Technology Infrastructure Library*), ou Biblioteca de Infraestrutura de Tecnologia da Informação, na tradução direta, é composta por um conjunto das melhores práticas para a definição dos processos necessários ao funcionamento de uma área de TI. Tem como objetivo permitir o máximo alinhamento entre a área de TI e as demais áreas do negócio, garantindo-se assim a geração de valor para a organização.

ITIL fornece orientações e não impõe regras de implementação, pois entende que os processos devem se adequar à realidade de cada organização, cabendo a ela decidir quais serão implementados e como irá seguir as orientações (MAGALHÃES, 2007, p. 65).

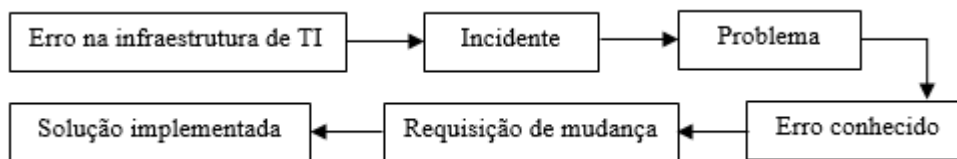
3.1 PROCESSO DE GERENCIAMENTO DE INCIDENTES

O processo de gerenciamento de incidentes é responsável pelo tratamento e pela resolução de todos os incidentes observados nos serviços de TI, visando o reestabelecimento dos serviços afetados no menor prazo possível. É a área responsável pelo atendimento dos usuários e registros dos incidentes, passando a zelar por eles durante todo o seu ciclo de vida (MAGALHÃES, 2007). Desta forma, pretende-se minimizar o impacto decorrente da redução do nível de serviço ou até mesmo a sua indisponibilidade total.

Incidente é todo e qualquer comportamento anormal de um serviço e que causa ou poderá causar indisponibilidade ou a redução do nível de desempenho. Em 99% dos casos, um incidente é registrado reativamente devido a estes fatores. Para registro de incidentes proativamente, implementa-se um Centro de Comando (*Command Center*), responsável por monitorar e sinalizar qualquer evento na infraestrutura de TI, o qual afeta ou possa afetar os serviços disponibilizados. (MAGALHÃES, 2007, p. 134)

Para buscar eficiência no processo de gerenciamento de incidentes, é necessária a implementação de um procedimento robusto de escalonamento e interação com os demais processos de gerenciamento de serviços, especialmente com Gerenciamento de Problema e Gerenciamento de Mudança. O objetivo maior é a implementação de uma solução definitiva para o problema observado conforme ilustrado na Figura 3.1.

Figura 3.1 – Evolução de um erro até a implementação da solução



Fonte: Magalhães (2007, p. 132)

A manutenção de dados históricos sobre os incidentes comunicados e resolvidos é de grande importância para o processo de Gerenciamento de Nível de Serviço. Estes dados poderão auxiliar na questão de avaliação e decisão pelo cancelamento de contratos com fornecedores externos, assim como na comparação entre níveis de serviço obtidos e os acordados com as áreas-cliente dos serviços de TI.

O processo de gerenciamento de incidentes também permite a melhora no atendimento aos usuários dos serviços de TI, o aumento da produtividade da equipe de analistas da Central de Serviços e a redução do tempo de indisponibilidade dos serviços. Com isto, se pretende preservar a imagem da TI criada pelas áreas-cliente, aumento da satisfação dos clientes e usuários, bem como o uso mais eficiente dos recursos.

3.1.1 Ciclo de vida de um incidente no *Service Desk* baseado em ITIL

O ciclo de vida de um incidente inicia-se assim que o usuário percebe a queda do nível de desempenho de um determinado serviço de TI ou até mesmo a indisponibilidade do mesmo. Desta forma, o *Service Desk* em questão dispõe de 3 formas básicas para comunicação de um incidente: telefone, e-mail e portal de autoajuda. Todos os meios estão disponíveis 24 horas por dia, 7 dias por semana.

Para que um usuário consiga registrar um incidente, é necessário previamente que já tenha sua identificação junto ao *Service Desk*, a qual também é utilizada para direcionar corretamente um analista que fale a mesma língua que o usuário. O registro do incidente é feito por um analista de *Service Desk*, principal figura deste processo, em uma ferramenta própria para isto, integrada com as demais abordagens citadas pela ITIL. Esta ferramenta é utilizada globalmente, em todos os países onde o *Service Desk* global foi implantado.

Através da classificação e outros parâmetros descritos pela ITIL, como impacto e urgência, é estabelecida uma prioridade para a solução do incidente. Para cada prioridade, existe um nível de serviço (SLA) em horas acordado por contrato com o fornecedor, que deve

solucionar o problema dentro do prazo estabelecido. Caso este não resolva dentro do prazo acordado, estará sujeito a penalidades estabelecidas no contrato.

Durante o registro do incidente por telefone, o analista de *Service Desk* direciona, através de *scripts*, algumas perguntas fundamentais para que se possa classificar, tentar identificar o problema e resolver ainda por telefone, caso já existam registros na base de conhecimento sobre o problema reclamado. Caso o problema não possa ser solucionado pelo nível 1 de atendimento, o mesmo é direcionado para a segunda equipe responsável pela linha de serviço afetada, de acordo com a classificação que o analista designou para aquele incidente.

No ciclo de um incidente, existem basicamente 3 níveis de atendimento. O guia de implementação da ITIL, no qual o ambiente se baseia, fala sobre a estrutura de responsabilidades de acordo com cada nível (MAGALHÃES, 2007):

1º nível

- Registro do incidente;
- Suporte inicial e classificação;
- Propriedade do incidente até seu encerramento;
- Monitoração do progresso de atendimento;
- Acompanhamento do Acordo de Nível de Serviço de atendimento;
- Comunicação de qualquer fato relacionado com o atendimento do incidente para a área usuária;
- Encaminhamento dos incidentes aos demais níveis quando o incidente não está encerrado;
- Resolução e recuperação de incidentes não atribuídos ao segundo nível;
- Encerramento dos incidentes.

2º nível

- Atendimento dos incidentes encaminhados pelo primeiro nível;
- Monitoramento dos detalhes do incidente, incluindo os Itens de Configuração afetados;
- Pesquisa da causa e diagnóstico para solução do incidente;
- Detecção de possíveis problemas e respectivo encaminhamento para a equipe responsável pelo processo de gerenciamento de problema;
- Resolução e recuperação de incidentes encaminhados.

3º nível

- Atendimento dos incidentes encaminhados pelo segundo nível;

- Pesquisa da causa e diagnóstico da solução para resolução do incidente;
- Detecção de possíveis problemas e respectivo encaminhamento para a equipe responsável pelo processo de gerenciamento de problema;
- Resolução e solução de incidentes encaminhados.

Os incidentes resolvidos no primeiro nível, ainda com o usuário na linha, são conhecidos, de acordo com ITIL, como FTF (*First Time Fix* – “solução na primeira vez”, na tradução direta). FTF é uma métrica que indica a capacidade do *Service Desk* resolver incidentes no nível 1, sem a necessidade de escalar para outros níveis de suporte (segundo e terceiro nível: suporte local, de campo, ou profissionais de TI focados em aplicações, *data center*, etc.). A meta do *Service Desk* é resolver pelo menos 80% dos incidentes classificados como tal. Para a identificação de novos incidentes candidatos a FTF, estes devem se encaixar em uma governança pré-estabelecida, conforme descrita abaixo e, obviamente, aprovada pela companhia e fornecedor:

- Sob circunstâncias normais, a solução de um FTF pode ser atingida em 100% dos casos;
- As etapas necessárias para concluir um FTF estão completamente sob controle do *Global Service Desk* e não dependem de ações de outras partes, incluindo o usuário;
- Deve ser possível medir a taxa de sucesso nas resoluções;
- Não deve causar insatisfação para o usuário final, como por exemplo, levar muito tempo para ser resolvido ou necessitar de altos níveis de intrusão no equipamento do usuário.

Existe ainda uma segunda métrica, muito parecida com FTF, que é o FCR (*First Call Resolution* – “Resolução da primeira chamada” na tradução direta). FCR e FTF são resolvidos pelo *Service Desk*, porém o FCR é resolvido sem que o usuário esteja no telefone, através de uma segunda ligação ou contato por *e-mail*. No caso de FTF, o usuário entra em contato por telefone e o incidente é resolvido até aproximadamente 10 minutos, ainda com a ligação em andamento. Por outro lado, quando o incidente necessita alguma ação de outro grupo especialista ou uma rápida investigação sobre o problema, será um FCR. Em ambos os casos, devem existir processos mapeados e condições de serem medidos.

O incidente é transferido consecutivamente para uma equipe responsável por cada área afetada até que o incidente seja resolvido. Após o incidente ser marcado como “Resolvido” pela equipe responsável, o incidente volta aos cuidados do *Service Desk* que então faz a validação com o usuário. Caso o usuário confirme a solução do incidente, o incidente é fechado, se não, volta para a fila de atendimento da equipe especialista (também conhecido como *Resolver Group*) responsável novamente.

Caso um incidente ocorra repetidamente, o processo de gerenciamento de problemas deverá atuar para que seja tratada a causa raiz do incidente e para que desta forma este não ocorra novamente.

3.1.2 Indicadores Chave de Desempenho (KPIs)

Os indicadores de desempenho padrão (*KPI – Key Performance Indicator*) tem por objetivo prover pontos de controle que permitam avaliar a eficiência, eficácia, efetividade e economicidade do serviço.

No ambiente estudado, os KPIs são disponibilizados por demanda através de uma ferramenta especializada em *Business Intelligence (BI)* online, adquirida pela companhia. Esta ferramenta faz a captura dos dados no *software* de gerenciamento de *Service Desk* e, após isto, consolida os dados e gera as informações sobre o desempenho dos serviços, como também auxilia o monitoramento diário dos mesmos, tais como mostrado no Quadro 3.1.

Quadro 3.1 - Indicadores chave de desempenho de gerenciamento de incidentes

Perspectiva	Indicador
Eficiência	· Índice de evolução da quantidade de incidentes;
Eficácia	· Índice de incidentes encerrados no primeiro nível de suporte;
Efetividade	· Índice de incidentes encerrados dentro do prazo estabelecido;
Economicidade	· Índice de redução do prazo médio de incidentes de Categoria 1;
	· Índice de incidentes encerrados no primeiro atendimento;
	· Índice de satisfação dos usuários com o atendimento de incidentes;
	· Índice de incidentes resolvidos de forma remota;
	· Índice de evolução do custo médio por incidente encerrado.

Fonte: Magalhães (2007)

É importante ressaltar sobre o Índice de Satisfação dos Usuários (IMF), ou seja, a avaliação do usuário sobre o atendimento prestado. Sempre que um incidente é encerrado, um formulário é enviado por e-mail ao usuário que recebeu o atendimento. Este formulário contém 4 (quatro) perguntas relativas ao tempo de resolução, sobre a comunicação e linguagem utilizada pelo analista, e também o conhecimento demonstrado sobre o problema. A última pergunta refere-se à satisfação geral com o atendimento fornecido pelo grupo que atendeu o incidente. Esta informação fica armazenada juntamente com os demais registros dos incidentes, a qual é disponibilizada em relatórios mensais como um dos atributos do incidente.

Como o volume de informações é grande, cerca de 90 mil incidentes por ano, com aproximadamente 50 campos distintos capturados da ferramenta de gerenciamento, a análise

das informações se restringe, na maioria das vezes, nas informações consolidadas pela ferramenta de relatório, focada nos indicadores e não na investigação caso a caso de incidentes.

Outras limitações, tais como limite de banda dos *links* internacionais, restringem a obtenção de dados de um período maior, prejudicando possíveis análises para melhoria de processos sobre o ambiente.

4 MINERAÇÃO DE DADOS

Para Carvalho (2005), *data mining* pode ser definido como o uso de técnicas automáticas para exploração de grandes quantidades de dados que permitam descobrir novos padrões e relações que, devido ao volume de dados, não seriam facilmente descobertos a “olho nu” pelo ser humano. *Data mining* pode ser aplicado a qualquer massa de dados, sejam eles oriundos das áreas de medicina, economia, astronomia, geologia, entre outras áreas de estudo.

Atualmente possuímos uma grande quantidade de dados devido à facilidade em armazenar arquivos em computadores pessoais, bem como o maior acesso a discos de vários *gigabytes/terabytes*. Estes fatos muitas vezes fazem postergar facilmente decisões sobre o que fazer com tanta informação. Registramos cada momento de nossas vidas, e cada movimento é um registro em um banco de dados (CHAKRABARTI et al., 2008 – tradução nossa).

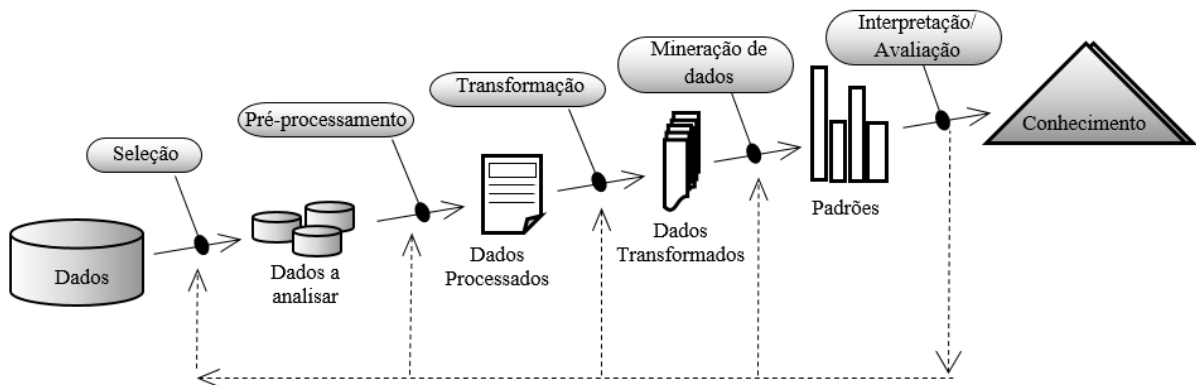
Segundo o IDC (2014 apud Grego, 2014), estima-se que os dados no mundo dobrem a cada 2 anos e, embora isto pareça difícil de se justificar, é só observarmos a quantidade de e-mails que recebemos diariamente, o conteúdo compartilhado em redes sociais e as facilidades que os mecanismos de busca oferecem para obtermos informações. Hoje em dia podemos acessar os dados de qualquer lugar ou dispositivo, até mesmo através de serviços de armazenamento online. Porém, do total de dados no mundo, apenas 22% contêm informação útil e somente 5% foram analisados e utilizados de alguma forma. Esses percentuais sugerem que há muito espaço para o crescimento da análise de *big data* no planeta, e por consequência, *data mining*, por ser uma de suas aplicações.

Com o crescimento em complexidade do mundo, *data mining* se transforma em um instrumento valioso para a elucidação de padrões que nos permeiam, como também trazer ideias e vantagens competitivas no âmbito comercial. Como um exemplo para ilustrar isto, se uma loja de roupas, um mercado que é extremamente competitivo, tiver um banco de dados com os perfis de clientes baseados nas escolhas de clientes antigos, isto seria a chave para o problema. Padrões de escolha podem ser analisados para identificar características distintas daqueles que são fieis de um determinado produto ou têm tendência a mudarem suas escolhas. Este grupo de clientes fieis pode ser marcado como o grupo de clientes especiais, a fim de serem tratados de forma diferenciada ou até mesmo lhes enviar ofertas com promoções especiais (CHAKRABARTI et al., 2008, tradução nossa).

4.1 KDD (KNOWLEDGE DISCOVERY IN DATABASES)

Segundo Braga (2005), a mineração de dados está inserida em um processo maior denominado descoberta de conhecimento em banco de dados, ou *Knowledge Discovery in Databases (KDD)*. Para Fayyad et al. (1996), a mineração de dados é um processo da descoberta de conhecimento, consistindo em algoritmos específicos que, sobre algumas limitações de eficiência computacional, produzem um número enumerado de padrões.

Figura 4.1 - Visão geral dos processos que envolvem a descoberta de conhecimento



Fonte: Adaptado de Fayyad et al,(1996) - tradução nossa

Segundo Fayyad, conforme ilustrado na Figura 4.1, procurando estabelecer uma sequência genérica de etapas para um projeto de descoberta de conhecimento, teríamos:

1. Desenvolver um entendimento prévio sobre o domínio da aplicação e os objetivos do usuário final;
2. Criar um conjunto de dados base: selecionar um conjunto de dados que tenha relevância, nos quais a descoberta será aplicada;
3. Limpeza e pré-processamento dos dados: operações básicas para remover ruídos ou pontos fora da curva se apropriado;
4. Redução e projeção de dados: procurar por funções úteis para representação dos dados dependendo do objetivo da tarefa. Utilizando redução de dimensionalidade ou métodos de transformação para reduzir o número efetivo de variáveis em consideração a fim de encontrar representações consistentes dos dados;
5. Escolha da tarefa de mineração de dados: decidir qual o objetivo da descoberta de conhecimento, se classificação, regressão, agrupamento, etc.
6. Escolha dos algoritmos de mineração de dados: selecionar os métodos a serem utilizados para a pesquisa de padrões no conjunto de dados. Isto significa decidir quais modelos e parâmetros serão apropriados e se encaixam com um método

específico de mineração de dados com os critérios gerais do processo de descoberta de conhecimento;

7. Mineração de dados: aplicação dos métodos escolhidos a fim de procurar por padrões interessantes que tenham uma forma de representação particular ou um conjunto dessas representações: regras de classificação ou árvores, regressão, agrupamento, entre outros;
8. Interpretação dos padrões descobertos e possíveis retornos para os passos 1-7 para execução de mais iterações;
9. Consolidação do conhecimento descoberto: incorporar o conhecimento descoberto no sistema de desempenho, ou simplesmente documentá-lo em relatórios para as partes interessadas.

4.2 TÉCNICAS

A mineração de dados pode ser realizada de três maneiras diferentes, cada uma baseada em função do nível de conhecimento que se tem sobre o problema a ser analisado. Se nada se sabe, pode-se deixar com que as técnicas automáticas de *data mining* procurem relações “novas” escondidas, as quais não poderiam ser identificadas a “olho nu”. Isto é chamado de descoberta não supervisionada de relações. Quando se possui algum conhecimento sobre a relação nova que está buscando, pode-se definir uma hipótese e verificar sua confirmação através da metodologia denominada teste de hipótese. Quando se tem um nível maior de conhecimento da área e da relação que se deseja estudar, procede-se com a metodologia de modelagem de dados (CARVALHO, 2005, p. 19).

Qualquer uma das três metodologias de *data mining* usa, basicamente, as mesmas técnicas para a sua realização. As técnicas são de caráter genérico e podem ser implementadas através de ferramentas diferentes, baseadas em Estatística, Redes Neurais Artificiais, Inteligência Artificial Simbolista, dentre outras (CARVALHO, 2005).

Existe um grande número de técnicas básicas, contudo, cinco técnicas gerais englobam didaticamente todas as outras formas de apresentação: Classificação, Agrupamento, Estimativa, Previsão e Afinidade (CARVALHO, 2005, p. 19). Para esta pesquisa, além da análise de regras de associação, também foram imprescindíveis as duas primeiras técnicas, as quais serão detalhadas nas seções seguintes.

4.2.1 Classificação

A classificação é uma das técnicas mais utilizadas de *data mining*, simplesmente porque é uma das tarefas cognitivas humanas mais realizadas no auxílio à compreensão do ambiente em que vivemos. O ser humano está simplesmente classificando o que está à sua volta (CARVALHO, 2005).

Esta técnica é uma abordagem sistemática para a construção de modelos de classificação a partir de um conjunto de dados de entrada. Ela emprega um algoritmo de aprendizagem para identificar um modelo que seja mais adequado para o relacionamento entre o conjunto de atributos e o rótulo (atributo alvo) da classe de dados de entrada. O modelo gerado pelo algoritmo de aprendizagem deve se adaptar bem aos dados de entrada e prever corretamente os rótulos de classes de registros que ele nunca viu antes. Portanto, o objetivo chave do algoritmo de aprendizagem é construir modelos com boa capacidade de generalização.

Um modelo de classificação é útil para os propósitos de modelagem descritiva e preditiva. Pode ser tratado como uma caixa preta que atribui automaticamente um rótulo de classe quando recebe um conjunto de atributos de um registro desconhecido.

Um modelo de **classificação descritiva** pode servir como ferramenta explicativa para se distinguir entre os objetos e classes diferentes. Um modelo de **classificação preditiva** pode ser usado para prever o rótulo da classe daqueles registros não conhecidos (CARVALHO, 2005).

Um conjunto de dados pode ser visto como uma coleção de objetos de dados. Por sua vez, objetos de dados são descritos por um número de atributos que capturam características básicas de um objeto, como a identificação de um incidente, a sua descrição e sua prioridade. Outros nomes para um atributo são *variável*, *característica*, *campo recurso* ou *dimensão*. Ainda, podem ser definidos quatro tipos de atributos: nominal, ordinal, intervalo e proporção. Atributos nominais e ordinais são chamados coletivamente de atributos **categorizados** ou **qualitativos** (ex. ID de um usuário). Já os tipos de atributos intervalar e proporcional são chamados coletivamente de **quantitativos** ou **numéricos** (TAN, STEINBACH e KUMAR, 2009, p. 39).

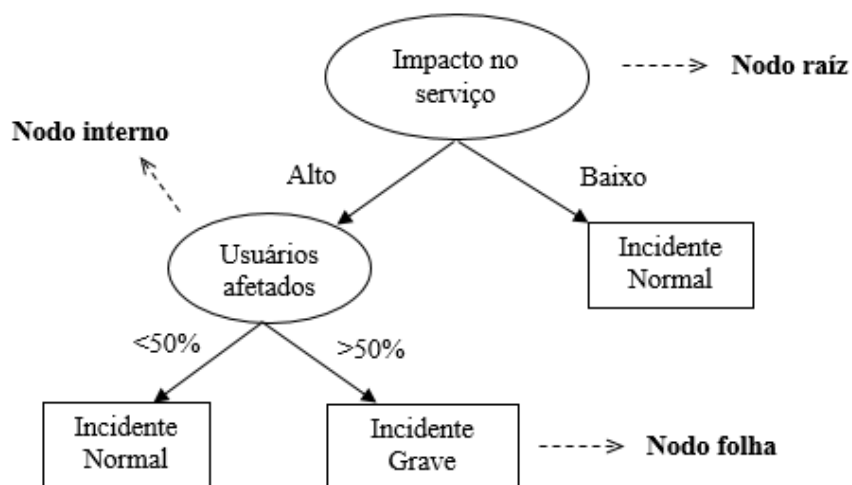
As técnicas de classificação são mais apropriadas para prever ou descrever conjuntos de dados com categorias nominais (diversas categorias) ou binárias (somente duas categorias, 0 ou 1). Elas são menos efetivas para categorias ordinais porque não consideram a ordem implícita entre as categorias (**numéricos**) (TAN, STEINBACH e KUMAR, 2009, p. 173).

4.2.1.1 Árvores de Decisão

Uma técnica de classificação simples, porém muito utilizada, é a árvore de decisão (TAN, STEINBACH e KUMAR, 2005). Para Han e colaboradores (2011, p. 331), a construção de árvores de decisão para classificação é tão popular porque ela não exige domínio de conhecimentos específicos ou parametrização de configurações, e também é apropriado para descoberta de conhecimento. Os algoritmos de construção de árvores de decisão podem manipular dados multidimensionais, então, a representação do conhecimento adquirido em forma de árvore é mais intuitiva e facilmente assimilável para os humanos.

A técnica de árvore de decisão consiste em separar os indivíduos (ou o que se deseja classificar) em grupos distintos de categorias e detalhar os mesmos em subgrupos de forma que se possam extrair todas as características que se deseja classificar. A árvore de decisão é organizada em categorias e subcategorias com estrutura hierárquica consistindo de nodos e arestas direcionadas, conforme ilustrado na Figura 4.2 (TAN, STEINBACH e KUMAR, 2005, p. 176).

Figura 4.2 - Classificando a prioridade de um incidente



Fonte: Elaborado pelo autor

A fim de ilustrar o exemplo da Figura 4.2, suponha que um incidente esteja sendo aberto pelo *Service Desk* e é desejado saber se o gerente de incidentes deve ser acionado ou não, ou seja, se o incidente é grave.

Se houver uma base de registros de incidentes, uma árvore de decisão pode ser montada de acordo com os dados pré-existentes sendo possível resolver este problema de classificação. A cada resposta recebida, uma questão seguinte é feita até que se chegue a uma conclusão sobre o rótulo da classe do registro. A primeira questão que pode ser feita é qual o tipo de impacto o

incidente gera no serviço afetado, se é alto ou baixo. Se o impacto for baixo, então, definitivamente o incidente é normal. Caso contrário, pode ser um incidente normal ou grave. No último caso, é necessário saber qual a quantidade de usuários é afetada pelo incidente. Se for menos de 50% da base de usuários, o incidente é normal, se for acima disso, o incidente é grave.

Esta série de questões e suas respostas possíveis podem ser organizadas em forma de árvore de decisão com estrutura hierárquica, consistindo de nodos e arestas direcionadas.

- Um nodo raíz é aquele que não possui arestas chegando, e zero ou mais arestas saindo;
- Nodos internos, são aqueles onde há exatamente uma aresta chegando e duas ou mais nodos saindo;
- Nodos folha ou terminais, possuem exatamente uma aresta chegando e nenhuma saindo (TAN, STEINBACH e KUMAR, 2009, p. 177).

Algoritmos para construção de Árvores de Decisão

Durante os anos 70-80, J. Ross Quinlan, um pesquisador em aprendizado de máquina, desenvolveu um algoritmo de árvores de decisão conhecido como ID3. Como sucessor deste trabalho, foi desenvolvido o algoritmo C4.5, que se transformou em uma referência entre os algoritmos existentes até hoje. Mais tarde, um grupo de pesquisadores de estatística publicou o livro *CART (Classification and Regression Trees)*, descrevendo a geração de árvores de decisão binárias. Os algoritmos ID3, C4.5 e CART adotaram uma abordagem na qual as árvores de decisão são construídas recursivamente de cima para baixo, a partir da estratégia de “dividir para conquistar” (HAN, KAMBER, PEI, 2011, p. 332).

Inicialmente, existem muitas árvores de decisão que podem ser construídas a partir de conjuntos de dados ou atributos. Embora algumas árvores sejam mais precisas que outras, encontrar uma árvore ótima é computacionalmente inviável devido ao tamanho exponencial do espaço de pesquisa.

Entretanto, alguns algoritmos têm sido construídos para indução de árvores de decisão, os quais são razoavelmente precisos, embora não perfeitos, em uma quantidade razoável de tempo empregado para isto. Estes algoritmos basicamente utilizam-se de estratégias para tomada de decisões, ou seja, tomam decisões localmente ótimas para particionar os dados (TAN, STEINBACH e KUMAR, 2009, P. 179).

O **Algoritmo de Hunt** é a base de muitos algoritmos conhecidos, tais como **ID3** (*Induction Decision Tree*), **C4.5** e **CART** os quais utilizam-se dos mesmos princípios porém com adaptações para determinados tipos de problemas e situações.

O algoritmo de Hunt utiliza-se de técnicas recursivas para o particionamento dos registros de treino em sucessivos subconjuntos mais puros, a fim de construir o modelo de classificação. Supondo que D_t seja um conjunto de registros de treino que estão associados ao nodo t e $y = \{y_1, y_2, \dots, y_c\}$ sejam os rótulos das classes. Se todos os registros em D_t forem da mesma classe y_i , então t é um nodo folha rotulado como y_i . Se D_t contiver registros que tenham mais de uma classe associada, uma **condição de teste atributo** é selecionada para particionar os registros em subconjuntos menores. Para cada resultado de condição de teste, um nodo filho será criado e os registros de D_t são distribuídos para os filhos baseados nos resultados. O algoritmo então é aplicado recursivamente a cada nodo filho (TAN, STEINBACH e KUMAR 2009, p. 180).

Para exemplificar, considera-se a Tabela 4.1 como o conjunto de dados obtido através da análise de dados de tomadores de empréstimos anteriores. A aplicação do algoritmo de Hunt encontrará um modelo de classificação que possa prever se um solicitante de empréstimo pagará suas obrigações ou se tornará inadimplente, por isso, a informação a ser descoberta envolve a classe Inadimplente, sendo esta que ficará nos nodos folha.

Tabela 4.1- Conjunto de treino para prever tomadores de empréstimos que ficarão inadimplentes

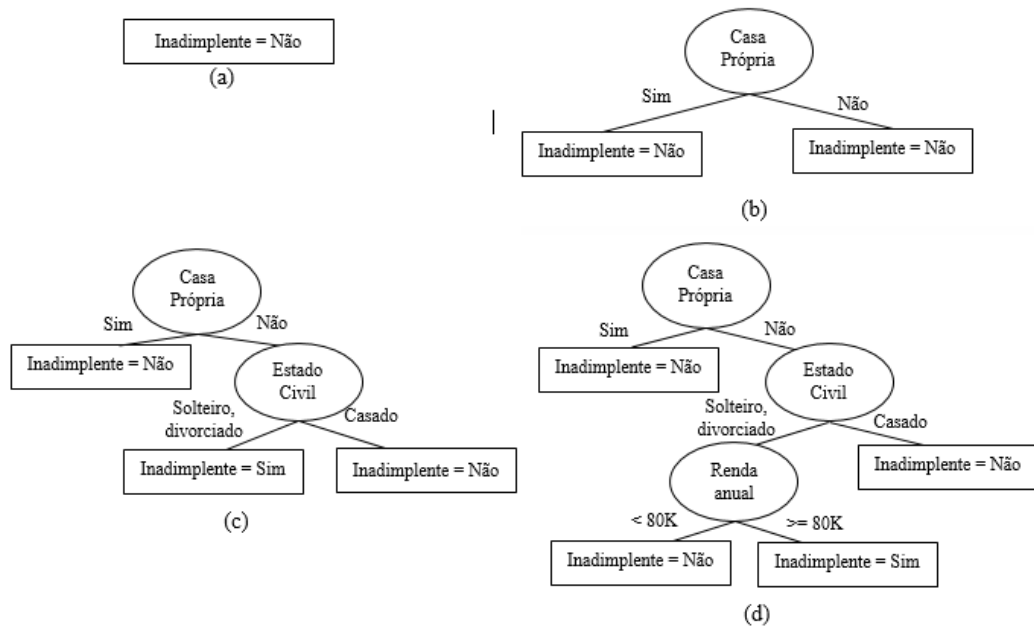
Tipo:	Binário	Catégorico	Contínuo	Classe
Tid	Casa Própria	Estado Civil	Renda Anual	Inadimplente
1	Sim	Solteiro	125K	Não
2	Não	Casado	100K	Não
3	Não	Solteiro	70K	Não
4	Sim	Casado	120K	Não
5	Não	Divorciado	95K	Sim
6	Não	Casado	60K	Não
7	Sim	Divorciado	220K	Não
8	Não	Solteiro	85K	Sim
9	Não	Casado	75K	Não
10	Não	Solteiro	90K	Sim

Fonte: Tan, Steinbach e Kumar (2009, p. 180)

Inicialmente, a árvore terá um único nodo que será a classe “Inadimplente” = Não (a) (Figura 4.3), o que significa que a maioria dos tomadores de empréstimo paga suas obrigações. Refinando a árvore, visto que existem registros de ambas as classes, estes são divididos em subconjuntos menores baseados nos resultados da condição de teste “Casa Própria” (b),

supondo, por enquanto, que este seria o melhor critério para divisão dos dados neste ponto. A partir disto, o algoritmo de Hunt é aplicado recursivamente até que todos os registros pertençam à mesma classe (c) e (d). As árvores resultantes são mostradas na Figura 4.3 (TAN, STEINBACH e KUMAR, 2009).

Figura 4.3 - Algoritmo de Hunt para induzir árvores de decisão



Fonte: Tan, Steinbach e Kumar (2009)

O algoritmo de Hunt funcionará se toda combinação de valores de atributos estiver presente nos dados de treino e se cada combinação possuir um único rótulo de classe. Estas suposições devem ser analisadas com cuidado, pois não se aplicam na maioria das situações práticas.

4.2.2 Agrupamento

A técnica de Agrupamento (ou *Clustering*) divide os dados de um conjunto, em grupos para que tenham um significado. A análise de grupos, ou agrupamento, é útil para compreensão dos dados e resumo deles. Também pode ser utilizada por diversas áreas, desde biologia, estatística, reconhecimento de padrões, aprendizagem de máquina e mineração de dados.

O objetivo é que os objetos dentro de um grupo sejam semelhantes entre si e diferentes dos objetos de outros grupos. Quanto maior a semelhança dos objetos de dentro de um grupo, e maior a diferença entre os grupos existentes, melhor ou mais distinto será o agrupamento.

No contexto da compreensão de dados, grupos são potenciais classes e a análise de grupos é o estudo de técnicas para automatizar a descoberta de classes. (TAN, STEINBACH e KUMAR, 2009, p. 582)

Já o agrupamento por utilidade fornece uma abstração de objetos individuais de dados para os grupos nos quais esses objetos de dados residem. No contexto da utilidade, a análise de grupos é o estudo de técnicas para encontrar os protótipos de grupos mais representativos.

A análise de grupos está relacionada a outras técnicas que são usadas para dividir objetos de dados em grupos. O agrupamento pode ser considerado como uma forma de classificação, pelo fato de criar rótulos nos objetos das classes (grupos).

Tipos de grupos:

- Hierárquico versus particional

O tipo mais comumente discutido é se o conjunto de grupos está aninhado ou não, ou ainda, hierárquico ou particional. Um agrupamento particional é simplesmente um conjunto de objetos de dados em subconjuntos (grupos) não interseccionados, de modo que cada objeto de dado esteja exatamente em um subconjunto.

Caso subgrupos sejam permitidos, então será obtido um agrupamento hierárquico, o qual corresponde a um conjunto de grupos aninhados organizados como uma árvore. (TAN, STEINBACH e KUMAR, 2009, p. 587)

- Exclusivo versus Interseccionado versus Difuso

O **agrupamento exclusivo** se caracteriza quando cada objeto pertence a um único grupo. Quando isto não é possível, o agrupamento se caracteriza como não-exclusivo. Em uma abordagem geral, um agrupamento não-exclusivo, o mesmo que **interseccionado**, é usado para representar um objeto que pode pertencer simultaneamente a mais de um grupo (classe).

Já no agrupamento **difuso**, cada objeto pertence a um grupo com um determinado peso, variando de 0 (não pertence totalmente) a 1 (pertence totalmente). Isto porque um objeto pode representar um usuário do tipo empregado e analista, no caso de um cadastro de usuários da Central de Serviços, um analista, que faz parte de um grupo especialista, também é um empregado, porém, será atribuído um peso maior (e.g. 0,8) para o perfil “analista” pois é o grupo pelo qual o perfil é mais representativo. Ou seja, quando um objeto pode, arbitrariamente, pertencer a mais de um grupo.

- Completa versus Parcial

Um agrupamento **completo** é quando um objeto pode ser atribuído a um grupo, enquanto no **parcial** isto não se aplica. Isto porque um objeto dentro de um conjunto de dados pode parcialmente não pertencer a grupos bem definidos, ou porque são ruídos, elementos externos ou fundamentalmente não têm relevância para a análise.

4.2.2.1 *Diferentes tipos de agrupamentos*

O agrupamento tem por objetivo identificar grupos de objetos, os quais sejam úteis e de acordo com a análise de dados. A seguir são descritos alguns tipos de agrupamento que podem ser encontrados, segundo Tan e colaboradores (2009, p. 589).

- Bem Separados

Um grupo pode ser definido como um conjunto de objetos os quais tem características iguais ou muito semelhantes a outros objetos do mesmo grupo, ou seja, não é parecido com objetos que não pertençam a este grupo. Para isto, pode-se utilizar um limite para especificar que todos os objetos do mesmo grupo possuem valores próximos entre si. Em outras palavras, a distância entre dois pontos quaisquer em grupos diferentes é maior do que a distância entre dois pontos quaisquer dentro do mesmo grupo.

- Baseados em Protótipos

Um grupo é um conjunto de objetos no qual cada objeto está mais próximo, mais semelhante, do protótipo que define o grupo do que do protótipo que define qualquer outro grupo. Para dados com atributos contínuos, um protótipo pode ser a média dos pontos do grupo, ou seja, o ponto mais central (grupos baseados no centro), tendendo a ser globulares.

- Baseados em Grafo

Grupos baseados em grafos podem ser vistos, se representados em um grafo onde os nodos são os objetos e os links representam as conexões, como componentes conectados. Pode-se citar, por exemplo, objetos que sejam conectados entre si e que não possuam conexões com objetos de fora do grupo.

- Baseados em Densidade

Um grupo também pode ser representado como uma região densa de objetos rodeada por uma região de baixa densidade. Uma definição baseada em densidade de um grupo muitas vezes é empregada quando os grupos são irregulares ou entrelaçados e, quando existem ruídos ou elementos externos.

- Propriedades Compartilhadas (Grupos Conceituais)

Como o nome já revela, pode ser definido como um grupo de objetos que possuem propriedades compartilhadas, falando-se das definições prévias de um grupo. Um algoritmo de agrupamento precisa de um conceito muito bem definido de um grupo para detectar com sucesso estes grupos e este processo pode ser denominado agrupamento conceitual, segundo Tan e colaboradores (2009, p. 591).

4.2.2.2 Algoritmo *K-Means*

K-means é um algoritmo particional de agrupamento amplamente utilizado e bastante tradicional. Este algoritmo é baseado em protótipos, e tenta encontrar um número de grupos (k) especificado pelo usuário, que são representados pelos seus centróides.

O centróide é uma medida alternativa para avaliar a qualidade de grupos e é definido como a média da distância de cada objeto de um grupo ao centróide do grupo (ponto médio no espaço dentro do grupo), enquanto medóides são objetos do próprio conjunto de dados, utilizados em análises nas quais não é possível determinar o centróide (análise genética, 3D, etc.) (HAN, KAMBER, PEI, 2011, p. 108 – tradução nossa).

Existe uma série de algoritmos de agrupamento, mas aqueles que mais se destacam são *K-means* e *K-medoid*. *K-means* define grupos a partir de um centróide, que é geralmente obtido através do cálculo da média de um grupo de pontos e é geralmente aplicado aos objetos em um espaço n -dimensional contínuo. *K-medoid* define um protótipo em termos de um medóide, que é o ponto mais representativo para um grupo de pontos e pode ser aplicado a uma extensa faixa de dados, já que se baseia na medida de proximidade entre um par de objetos. (TAN, STEINBACH e KUMAR, 2009, p. 593).

Segundo Han et al. (2011, p. 451), o algoritmo *K-means* define o centróide de um grupo como sendo o ponto médio dos valores do conjunto. Primeiramente, seleciona k centróides aleatoriamente, dentro do grupo para representar o ponto médio ou centro inicial. Para os objetos restantes, um objeto é atribuído ao grupo mais semelhante baseando-se na distância Euclidiana entre o objeto e o a média k do grupo. O algoritmo *k-means*, iterativamente, melhora a variação interna do agrupamento. Para cada grupo, ele calcula o novo centro usando os objetos atribuídos para o grupo na iteração anterior. Sendo assim, todos os objetos são novamente atribuídos usando o novo centro do grupo. As iterações continuam até que as atribuições tornem-se estáveis, ou seja, quando os objetos agrupados no ciclo atual são os mesmos que na iteração anterior, conforme pode-se observar na Figura 4.4.

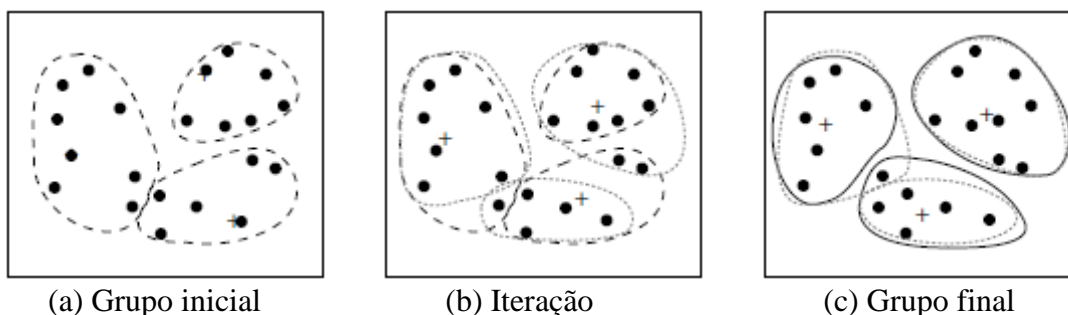
A seguir, uma breve descrição do funcionamento do algoritmo k-means básico (Quadro 4.1), o qual pode ser vinculado à Figura 4.4. No primeiro passo (Figura 4.4(a)), os pontos (+) são atribuídos aos centróides iniciais, onde estão os grupos com mais pontos. Do segundo passo (b) até o 4, os pontos são atribuídos aos centróides atualizados e os mesmos são atualizados sucessivamente. Quando o algoritmo termina (Figura 4.4(c)), no passo 5, os centróides ficam no centro dos grupos e estes ficam bem definidos.

Quadro 4.1 - Algoritmo K-means básico

- 1: Selecione K pontos como centróides iniciais.
- 2: **Repita**
- 3: Forme K grupos atribuindo cada ponto ao seu centróide mais próximo.
- 4: Recalcule o centróide de cada grupo/
- 5: **Até que** os centróides não mudem.

FONTE: Tan, Steinbach e Kumar (2009, p. 594)

Figura 4.4 - Agrupamento de objetos utilizando o algoritmo *k-means*



FONTE: Han, Kamber, Pei (2011)

4.2.3 Regras de associação

Há alguns anos, um grande número de lojas *online* foram criadas para vender livros. Ao longo do tempo, foram coletadas informações sobre os livros que cada cliente adquiriu. Utilizando análise de associação, foi possível identificar grupos de livros similares nos quais o cliente se interessou ao longo do tempo. Através desta informação, foi possível desenvolver sistemas de recomendação que informam ao usuário sobre outros exemplares nos quais ele poderia estar interessado, baseando-se no livro que está interessado em comprar (WILLIAMS, 2011).

A técnica de Análise de Associação identifica relações ou correlações entre as observações e/ou entre variáveis em um conjunto de dados. Estes relacionamentos são então expressados como regras de associação. A abordagem tem sido aplicada em um grande número

de bancos de dados transacionais, como cestas de compras e compras online de clientes (WILLIAMS, 2011).

Regras de associação são similares a regras de classificação. Elas podem ser encontradas da mesma maneira, executando a indução de regras através de procedimentos “dividir e conquistar” para cada expressão possível que possa ocorrer no lado direito da regra, ou seja, o resultado. Mas não somente isto, uma simples regra de associação frequentemente produz o resultado de um ou mais atributos associados. (WITTEN, 2005, p. 112)

A base de um algoritmo de análise de associação é a geração de conjunto de itens mais frequentes. Um conjunto de itens frequentes são aqueles que ocorrem juntos com alguma repetição suficiente para ser considerado como um candidato para a geração de uma regra de associação (WILLIAMS, 2011).

Para encontrar estas regras, deve-se executar a um algoritmo de indução de regras de associação para cada possível combinação de atributos juntamente com todas as possibilidades de valores. Isto gerará uma grande quantidade de regras de associação que devem ser eliminadas baseando-se na sua *coverage* (ou cobertura – número de instâncias estimadas corretamente) e sua *accuracy* (precisão – o mesmo número expresso como uma proporção do número de instâncias nas quais a regra se aplica). (CHAKRABARTI, 2008, p. 160)

Uma vez que todos os conjuntos de regras foram gerados com a cobertura (*coverage*) necessária o próximo passo é investigar cada regra, ou conjunto de regras, com a mínima precisão especificada. Alguns conjuntos de variáveis irão gerar mais de uma regra, outras, nenhuma. (CHAKRABARTI, 2008, p. 161)

4.2.3.1 Gerando regras com eficiência

Para gerar regras com eficiência existem basicamente duas etapas: geração de conjuntos de regras com mínima cobertura e para cada conjunto de regras, em seguida determinar quais atendem o requisito mínimo de precisão.

A primeira etapa consiste em gerar um conjunto de regras, dado um nível mínimo de precisão pré-estabelecido e então utilizar isto para gerar conjuntos de regras de duas variáveis, três, etc. Cada etapa consiste em uma passagem no conjunto de dados utilizado para gerar as regras para contar quantos itens correspondem àquela regra. Após isto, uma lista com as regras aprovadas é armazenada em uma tabela de dispersão. (WITTEN, 2011)

A partir de regras de uma variável são gerados candidatos com duas variáveis, então novamente devem ser feitos testes no conjunto de dados e removidos aqueles que não atendem ao requisito mínimo de cobertura e precisão. Os candidatos de duas variáveis são simplesmente

todas as regras de uma variável agrupadas em pares, porque regras de duas variáveis não tem a mesma cobertura a menos que ambas variáveis de uma regra tenham o mesmo mínimo de cobertura. Um conjunto de regras de três variáveis somente podem atender um mínimo de cobertura se todas as três regras de duas variáveis também tenham um mínimo de cobertura, e assim por diante, caso contrário, serão eliminadas.

4.2.3.2 Representando as regras

A representação de regras de associação é necessária para identificar relacionamentos entre itens nos registros. Supondo que cada transação seja como uma lista de compras (que pode ser representado como $\{A, B, C, D, E, F\}$). O objetivo é identificar coleções de itens que aparecem juntos em algumas listas de compras (suponha que $\{A, C, F\}$ aparecem em várias listas de compras). A partir destes conjuntos de itens, podem ser identificadas regras como $A, F \Rightarrow C$, ou seja, quando A e F aparecem juntos em uma lista, então C também aparece tipicamente. (WILLIAMS, 2011 p. 196)

Basicamente duas medidas são utilizadas na análise de regras de associação: *support* (cobertura) e *confidence* (precisão). A mínima cobertura é expressa como a porcentagem do número total de transações no conjunto de dados. Informalmente, simplesmente quão frequente os itens aparecem juntos entre o resto das transações. Formalmente, pode ser definido como a quantidade de itens I como proporção de todas as transações, nas quais todos os itens I aparecem expressando a cobertura para a regra de associação como

$$support(A \Rightarrow C) = P (A \cup C).$$

Junto com cada regra, são informados alguns parâmetros como *confidence* (confiança, precisão). A confiança mostra a proporção de transações que contém A e que também contém C . A confiança especifica uma mínima probabilidade para a regra de associação. Por exemplo, se for escolhido para gerar somente regras que correspondem como verdade pelo menos 90% das vezes, isto quer dizer que, quando A aparece nas listas, C também aparece na mesma lista pelo menos 90% das vezes. Formalmente pode ser representada como a probabilidade condicional

$$confidence(A \Rightarrow C) = P (A / C) = P (A \cup C) / P(A)$$

ou também pode ser expresso em termos de cobertura (*support*):

$$confidence (A \Rightarrow C) = support(A \Rightarrow C) / support (A) - (WILLIAMS, 2011 p. 196).$$

O algoritmo Apriori, usado nesta pesquisa, utiliza um método de busca “gera e testa”. Somente após explorar todas as possibilidades de associação contendo k itens, todos os candidatos são verificados para determinar quais deles tem a cobertura necessária especificada.

Durante a pesquisa diversas ferramentas foram encontradas disponíveis no mercado. Neste capítulo serão citadas as principais ferramentas, bem como suas características.

4.2.4 WEKA

O *software* Ambiente Waikato para Análise de Conhecimento (tradução direta), é um *software* de domínio público desenvolvido na linguagem Java, pela Universidade de Waikato, Nova Zelândia. O WEKA contém diversos algoritmos para mineração de dados como *J48*, *Redes Neurais*, *K-Means*, entre outros. Grande parte de seus componentes são resultantes de teses de dissertações de grupos de pesquisa da mesma universidade. O *software* foi construído no intuito de investigar técnicas e aprendizado de máquina. Inicialmente foi direcionado para a agricultura, a principal área econômica da Nova Zelândia.

Possui uma interface gráfica amigável e intuitiva e dispõe de algoritmos que fornecem relatórios com dados analíticos e estatísticos como também se comunica com as principais arquiteturas de banco de dados utilizadas atualmente.

Verificou-se também que existem diversas pesquisas que utilizaram este *software*, bem como documentações *online* abrangentes sobre o código fonte. Por ser desenvolvido em Java, este possui interoperabilidade com diversas plataformas de sistemas operacionais, porém uma limitação que deve ser considerada no contexto desta pesquisa é o tamanho fixo do volume de dados a ser manipulado.

O pacote de instalação, assim como documentações e fóruns de discussão estão disponíveis no endereço <http://www.cs.waikato.ac.nz/ml/weka/>.

4.2.5 SPSS

O SPSS, ferramenta de propriedade da IBM, fornece uma gama de ferramentas modernas para mineração de dados. Oferece uma grande variedade de técnicas de aprendizado de máquina e modelagem estatística. Conforme a página da internet do produto, algumas aplicações são: análise preditiva de fraudes, atividades suspeitas, análise de texto, mídias sociais, *big data*, entre outros. Utiliza arquitetura aberta, a qual oferece suporte à conexão com diversos tipos de fontes de dados e sistemas de informação.

4.2.6 Linguagem R

R é uma linguagem de programação e um ambiente para computação estatística. É similar à linguagem S desenvolvida pela *AT&T Bell Laboratories*. Existem versões para as famílias de sistemas operacionais Unix, Windows e Mac.

O desenvolvimento atual da linguagem é continuado por uma equipe de dezenas de pessoas de diferentes instituições ao redor do mundo. O desenvolvimento da linguagem utiliza uma comunidade de desenvolvedores que cresce cada vez mais, os quais cooperam no seu desenvolvimento devido à filosofia de código aberto (REVOLUTION ANALYTICS, 2014). Há quem critique os modelos de código aberto, mas este certamente não é o caso da linguagem R. Existem excelentes documentações, livros e *sites* que disponibilizam informações gratuitamente sobre ela.

Também existem muitos pacotes adicionais que podem ser baixados diretamente nos repositórios disponibilizados pela aplicação, os quais permitem que sejam exploradas disciplinas como estatística, aprendizado de máquina, reconhecimento de padrões, entre outras. (TORGO, 2010)

Na mineração de dados são comuns enormes conjuntos de dados, com milhões de tuplas. Mais frequente ainda, os dados de treinamento não cabem em memória. Desta forma, a construção de árvores de decisão torna-se ineficientes devido ao tráfego de entrada e saída dos dados na memória cache pois não cabem na mesma. Nestes casos são necessárias abordagens escaláveis, capazes de manipular grandes quantidades de dados (HAN, KAMBER, PEI, 2009, p. 347 – tradução nossa).

Dentre os principais motivos do uso da linguagem R neste trabalho, pode-se citar:

- Explorar novos conhecimentos sobre a linguagem e a ferramenta;
- Licença gratuita;
- Documentação completa;
- Interface visual intuitiva, com pacotes disponíveis para *download* e facilidade no uso;
- Gerador de relatórios intuitivos;
- Bom desempenho para grandes bases de dados;
- Possibilidade de conexão com diversos bancos de dados conhecidos;

4.2.6.1 *Rattle*

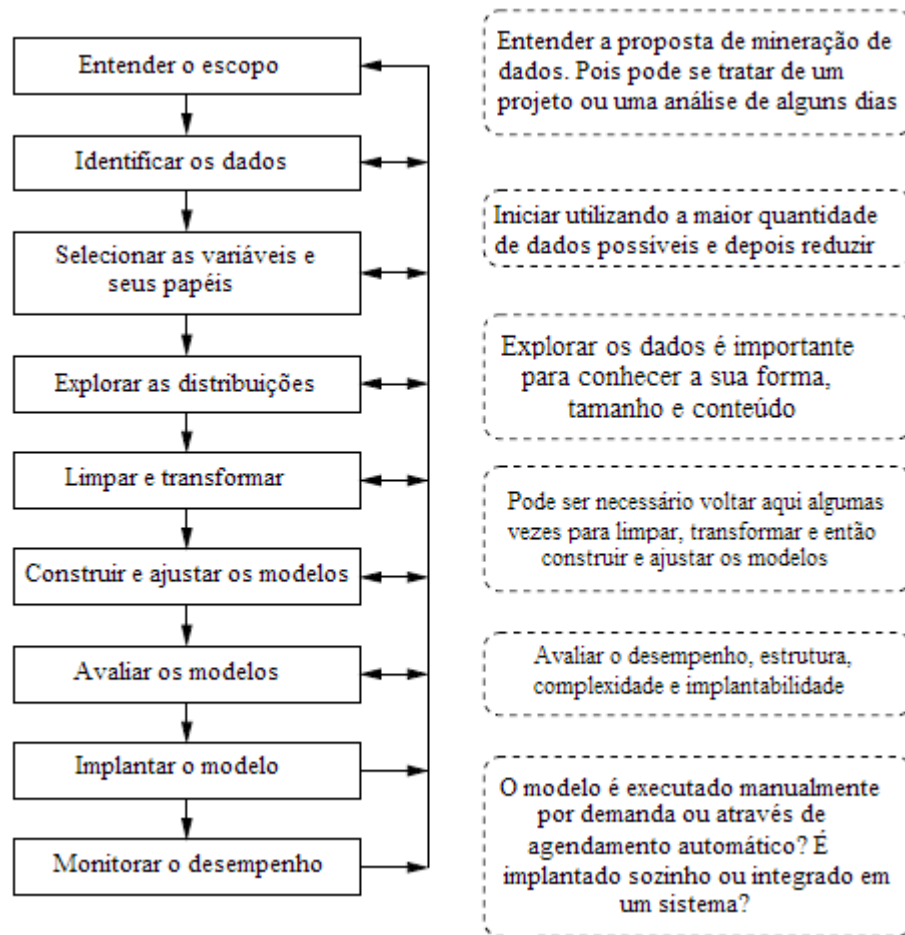
O Rattle é uma interface gráfica para mineração de dados baseada na linguagem de programação R, embora o entendimento da linguagem não seja necessário para utilizá-la. Trata-se de uma ferramenta fácil de utilizar, rápida para instalar e configurar e ainda permite trabalhar rapidamente através dos processos de processamento de dados, modelagem e avaliação, fases de um projeto de mineração de dados. (WILLIAMS, 2011)

A tela inicial do Rattle (Figura 4.7) mostra uma mensagem de boas-vindas e uma rápida introdução. Por outro lado, fornece uma linguagem muito poderosa para a realização de mineração de dados, bem além das limitações que são incorporados em qualquer interface gráfica de usuário (GUI) e as abordagens conseqüentemente envolvidas na mineração de dados (WILLIAMS, 2011).

Embora o usuário do Rattle não precisa necessariamente aprender R ele disponibiliza todo o código executado por trás da interface de modo a permitir ao usuário desenvolver diretamente do console do R, como também salvar *scripts* para futura referência. Este *script* pode ser carregado no R (fora do Rattle) para repetir qualquer tarefa de mineração de dados.

O Rattle possui tudo que um usuário necessita, especialmente no contexto de introdução à mineração de dados. No entanto, também estão disponíveis etapas mais sofisticadas para processamento e criação de modelos em R (WILLIAMS, 2011).

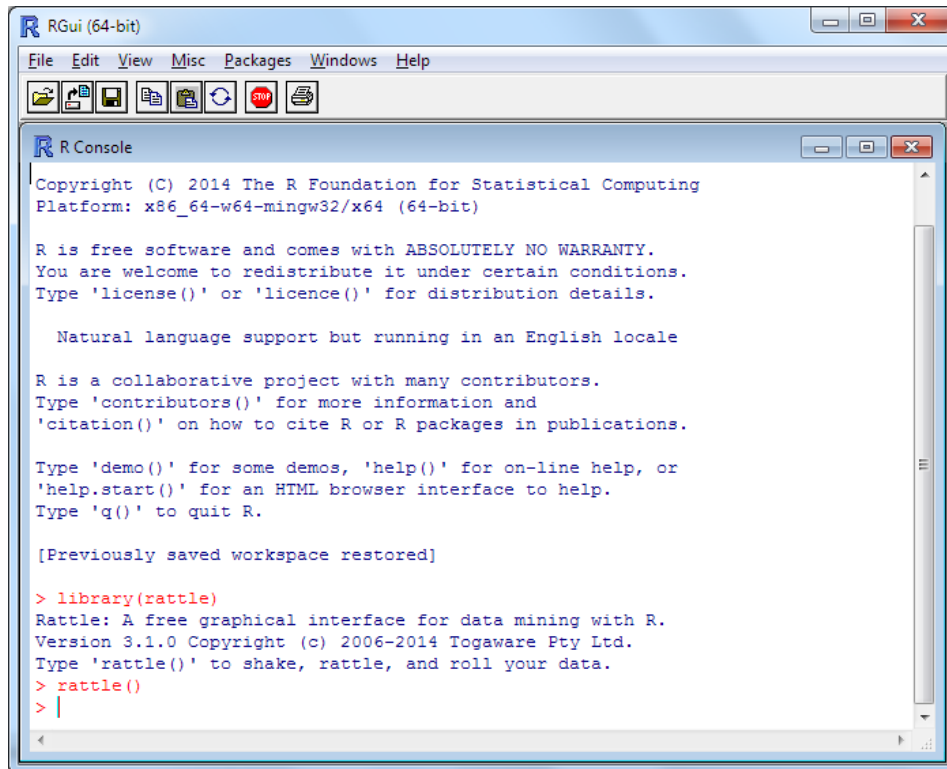
Figura 4.5 - Processo típico de um projeto de mineração de dados suportado pelo Rattle



Fonte: Willians (2011) – tradução nossa

As etapas típicas para um projeto de mineração de dados, no contexto dos menus do Rattle, estão listadas nas figuras Figura 4.6 até a Figura 4.12.

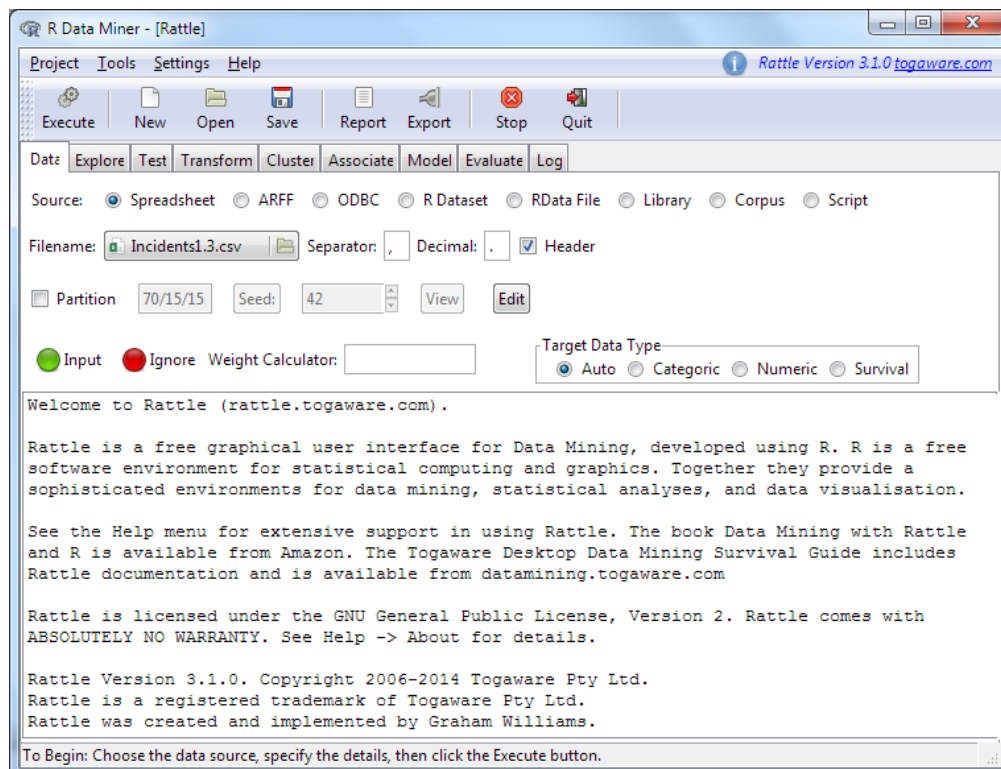
Figura 4.6 – Tela inicial do “RGui (64-bit)”



Fonte: Elaborado pelo autor

1. Carregar um conjunto de dados (*Load a dataset*);

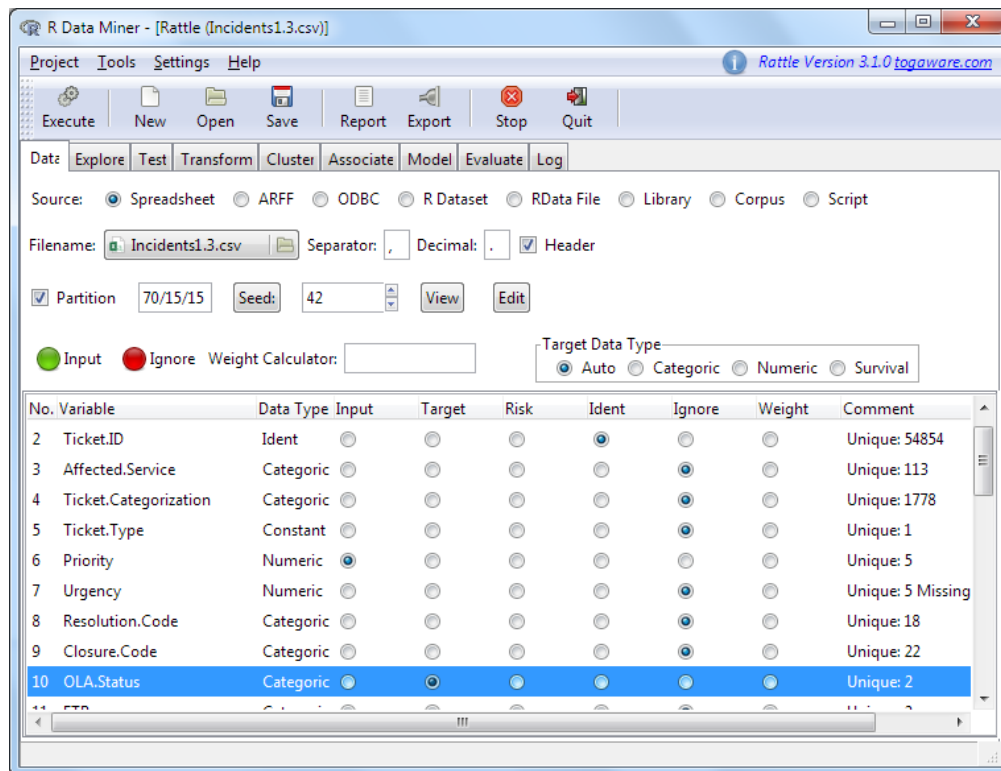
Figura 4.7 – Tela inicial do Rattle



Fonte: Elaborado pelo autor

2. Selecionar variáveis e entidades (*Select variables for exploring and mining*);

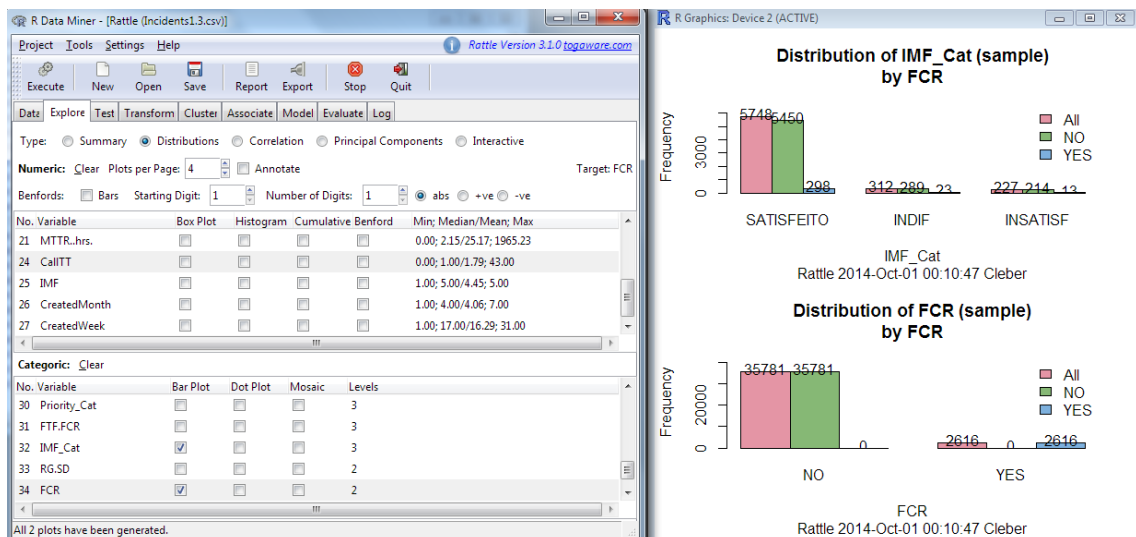
Figura 4.8 – Rattle – Tela de seleção de variáveis



Fonte: Elaborado pelo autor

3. Explorar os dados para entender como estão distribuídos e espalhados (*Explore*);

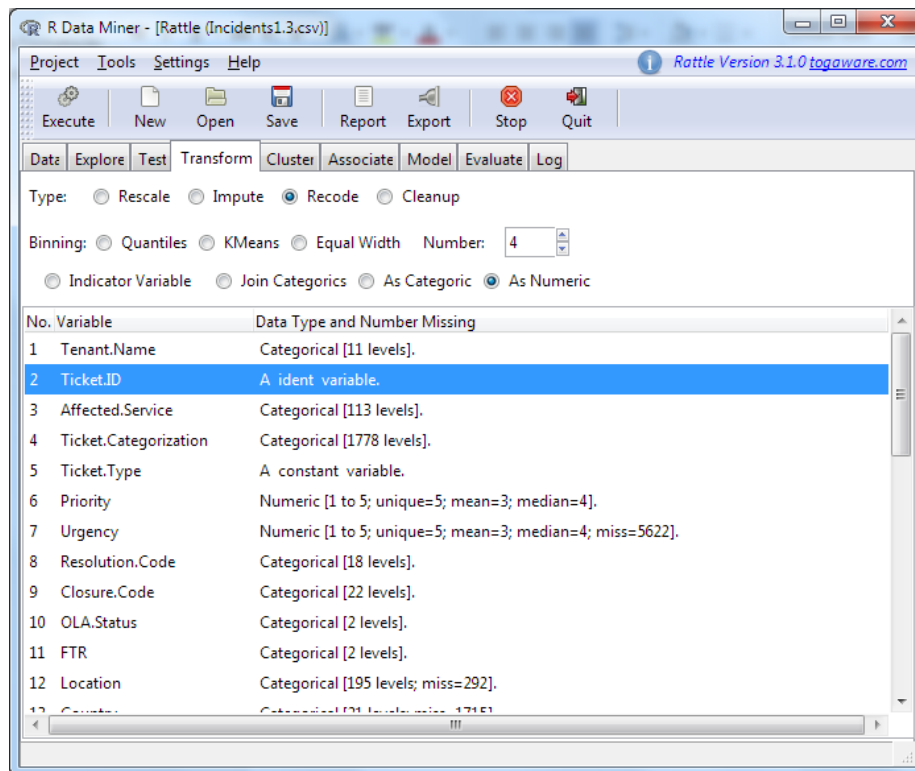
Figura 4.9 – Tela de exploração de dados



Fonte: Elaborado pelo autor

4. Transformar os dados para se adequar aos propósitos de mineração de dados (*Transform*);

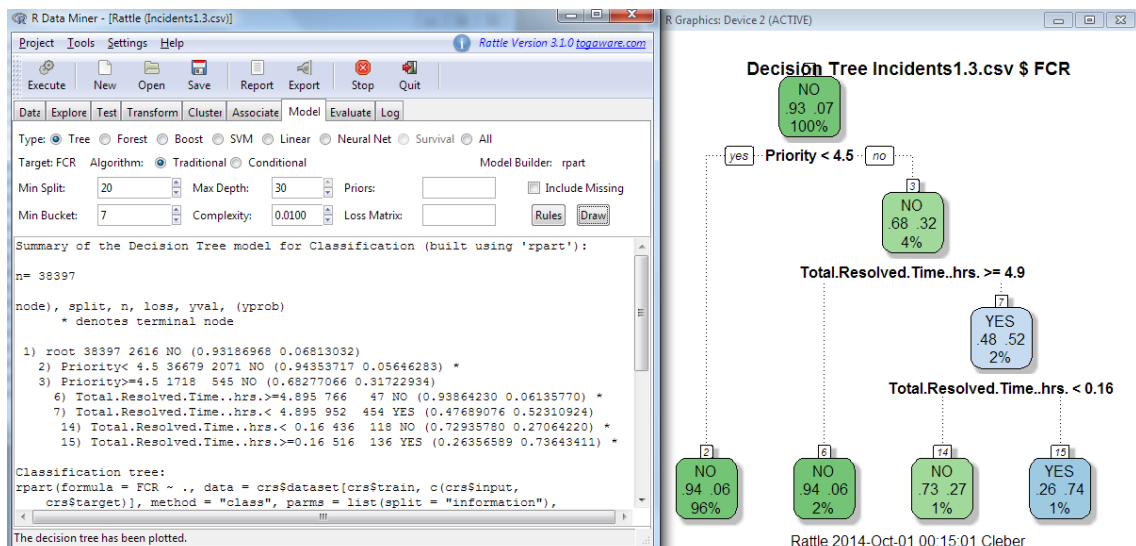
Figura 4.10 – Tela de transformação dos dados



Fonte: Elaborado pelo autor

5. Contruir modelos (*Build models*);

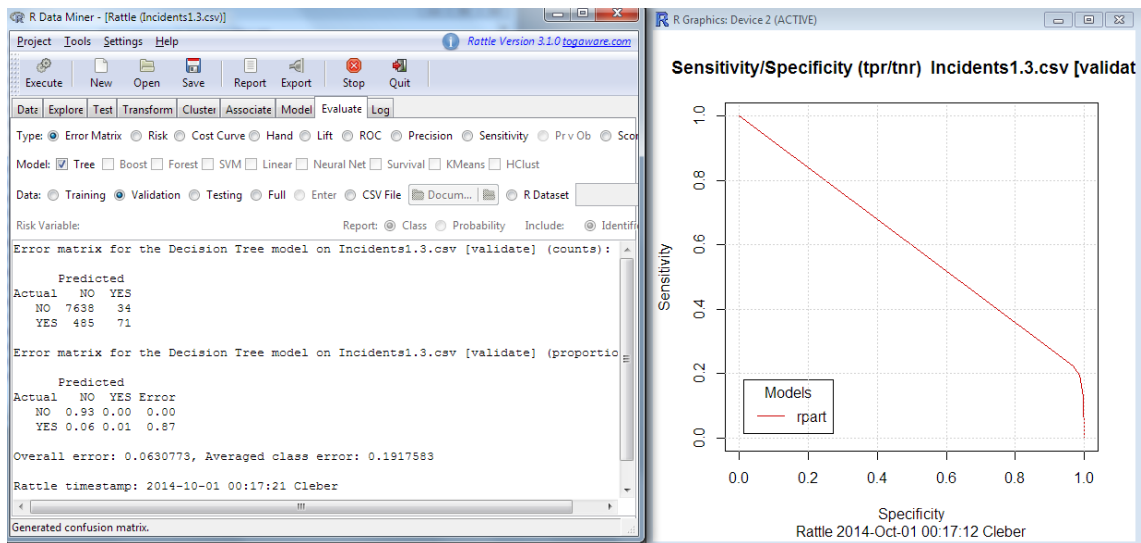
Figura 4.11 – Tela de construção de modelos



Fonte: Elaborado pelo autor

6. Avaliar os modelos (*Evaluate*);

Figura 4.12 – Tela de avaliação de modelos



Fonte: Elaborado pelo autor

7. Exportar os modelos (*Export the models*): Esta etapa está disponível também na Figura 4.12, juntamente com a etapa de avaliação do modelo. O Rattle permite exportar o conjunto de dados resultante para o formato CSV.

Falando-se especificamente de R, podem ser identificados alguns prós e contras para o uso desta linguagem de aplicação estatística, que serão descritos a seguir:

- É a linguagem para análise estatística mais fácil de compreender. Incorpora todos os padrões de testes estatísticos, modelos e análises, como também permite o gerenciamento de manipulação de dados;
- Como a linguagem e o ambiente foram desenvolvidos por pesquisadores e profissionais da estatística, há comunidade muito competente sobre o assunto;
- As funções gráficas do R são excelentes, fornecendo uma linguagem de programação repleta que supera a maioria das demais aplicações para análise estatística;
- Possui mais de 480 pacotes disponíveis, em múltiplos repositórios, especializados em assuntos como econometria, mineração de dados, análise espacial e bioinformática;
- R funciona muito bem importando dados de outras ferramentas como, por exemplo, arquivos CSV, SAS, SPSS, diretamente do *Microsoft Excel*, *Oracle*, *MySQL* e *SQLite*. Também é capaz de exportar figuras nos formatos de imagem como PDF, JPG, PNG, SVG e também tabelas como LATEX e HTML;

A seguir serão descritas algumas de suas desvantagens:

- Possui uma curva de aprendizagem longa– isto quer dizer que pode demorar algum tempo para que o usuário possa fazer uso do real poder da linguagem, mesmo assim, mais curto que qualquer outra linguagem;

- A qualidade de alguns pacotes não é perfeita, no entanto, se um pacote é útil a várias pessoas, pode rapidamente envolver esforços para melhorá-lo através do trabalho colaborativo;

- Embora os comandos normalmente não consumam uma grande quantidade de memória, alguns podem comprometê-la completamente. Isto pode ser uma restrição para a mineração de dados, porém existem várias soluções incluindo sistemas operacionais 64 *bits* que podem acessar muito mais memória do que sistemas operacionais 32 *bits*.

Com exceção da etapa de transformação, que foi executada com auxílio do *Microsoft Excel*, todas as demais foram executadas utilizando o Rattle. O *Microsoft Excel* é uma ferramenta muito ágil e eficiente no que diz respeito à aplicação de testes, utilizando simples fórmulas para transformar os atributos numéricos em categóricos. Como o autor possui domínio sobre ele, foi a melhor escolha para este caso. Nesta pesquisa, optou-se em utilizar a linguagem R e o Rattle para que fossem agregados conhecimentos sobre estas promissoras ferramentas, de modo a encorajar novos pesquisadores a explorá-las.

5 ANÁLISE DE INCIDENTES DE UMA CENTRAL DE SERVIÇOS DE TI

O *Service Desk* representa o primeiro nível de atendimento e registra cerca de 85% dos incidentes que são tratados pelas equipes especialistas de primeiro nível (internamente) e também de segundo e terceiro níveis de atendimento. Destes, cerca de 20% são resolvidos pelo próprio *Service Desk*, sem intervenção de outras equipes. O restante, em média 15%, são eventos proativos ou abertos diretamente por outras equipes de segundo nível. Estes incidentes são comumente informados de três maneiras: Telefone, E-mail ou enviados automaticamente por um Evento de Sistema, informações estas que são registradas em cada incidente, como pode ser visto na Figura 5.1, armazenados no atributo *Ticket Record Method*.

Figura 5.1 – Captura de tela da aplicação do *Service Desk*

The screenshot displays the TRIOLE® Service Desk interface. At the top, it shows the user 'Morales, Cleber on (TFSS01SS03) (Log Out)'. The main content area is titled '602 Incident Detail **' and includes several data fields:

Affected End User	Reported By	Preferred Language	Active?
Porras, Andres		Spanish	YES

Classification	Status	Incident Type	Recorded Method
1FJBAT.Application.Global IAM	Acknowledged	Regular	Telephone

Logged By	Assignee	Group	2nd'ary Assignment Group
Barrantes, Paola		BAT-ServiceDesk	

No. of Users	Impact	Urgency	Priority
Single user	3	3	3

Fonte: Elaborado pelo autor

Os incidentes abertos por telefone, são registrados através de um número gratuito disponibilizado pela companhia e são atendidos por analistas que falam o mesmo idioma do usuário, que deve se identificar por um número de registro. Quando o usuário tem problemas com seu número de identificação, o sistema o direciona através da identificação do telefone de origem da região onde ele se encontra.

Incidentes abertos por e-mail, que podem ser originados de um portal disponibilizado na *intranet* da companhia ou do próprio cliente de e-mail do usuário, possuem texto livre e dependem de inclusão e classificação manual por algum analista posterior ao recebido na caixa de e-mails do *Service Desk*. Após isso, o usuário afetado recebe um e-mail com os dados do incidente aberto. Existe uma meta de 2 horas para o registro do incidente após o recebimento do e-mail do usuário, que recebe uma notificação assim que o chamado é aberto (caso o e-mail não seja o serviço afetado).

Já os Eventos de Sistema podem ser gerados através de *softwares* de monitoramento de serviços de e-mail, backup, antivírus, bancos de dados, *links* de rede e internet, entre outros.

Os incidentes resolvidos no primeiro nível (FTF ou FCR) são sempre aqueles comunicados pelo usuário através do telefone. Esta é uma premissa, pois se considera que casos menos urgentes são abertos através de outras ferramentas como o e-mail e Eventos de Sistemas. Quando o usuário entra em contato através do telefone, a chamada deve ser atendida por um agente em menos de trinta segundos. Caso seja constatado que se trata de um problema conhecido e acordado como escopo do *Service Desk*, há grandes chances que este possa ser resolvido ainda com o usuário na linha.

Observando-se estas medidas de desempenho, foram identificadas algumas oportunidades de análise porém ainda sem uma ferramenta definida para analisá-las. Foi então que, através de pesquisas bibliográficas, observou-se que as técnicas de mineração de dados poderiam lidar com grandes volumes de dados com mais robustez e trazer respostas de forma eficiente.

O processo de descoberta de conhecimento é um processo baseado em técnicas de *data mining* (algoritmos) para identificar, o que é considerado conhecimento, informações de acordo com o grau e os limites, utilizando uma base de dados disponível na qual são necessárias etapas de pré-processamento, amostragem e transformação desta base de dados (FAYYAD et al, 1996).

No contexto prático, estas informações serão utilizadas como apoio para que sejam identificadas oportunidades de redução de custos, melhoria de processos, definição de metas estratégicas, entre outras possibilidades para a equipe de gestão de serviços de tecnologia da informação da companhia.

O processo de aquisição de conhecimento é interativo e iterativo, envolvendo diversas etapas com algumas decisões tomadas pelo pesquisador (FAYYAD et al, 1996). As etapas necessárias para a descoberta de conhecimento, de acordo com FAYYAD et al (1996), serão descritas, no contexto das informações utilizadas, nos próximos subcapítulos.

5.1 COMPREENSÃO DO ESCOPO

Primeiramente, se definiu claramente o objetivo da pesquisa, pois este influenciará no rumo de cada etapa do processo. Através de reuniões com o gerente de incidentes regional da companhia, verificou-se que uma das iniciativas que ainda possui oportunidades é melhorar a

eficiência do *Service Desk Global*, situado na Costa Rica, no sentido de que mais incidentes sejam resolvidos no primeiro nível de atendimento.

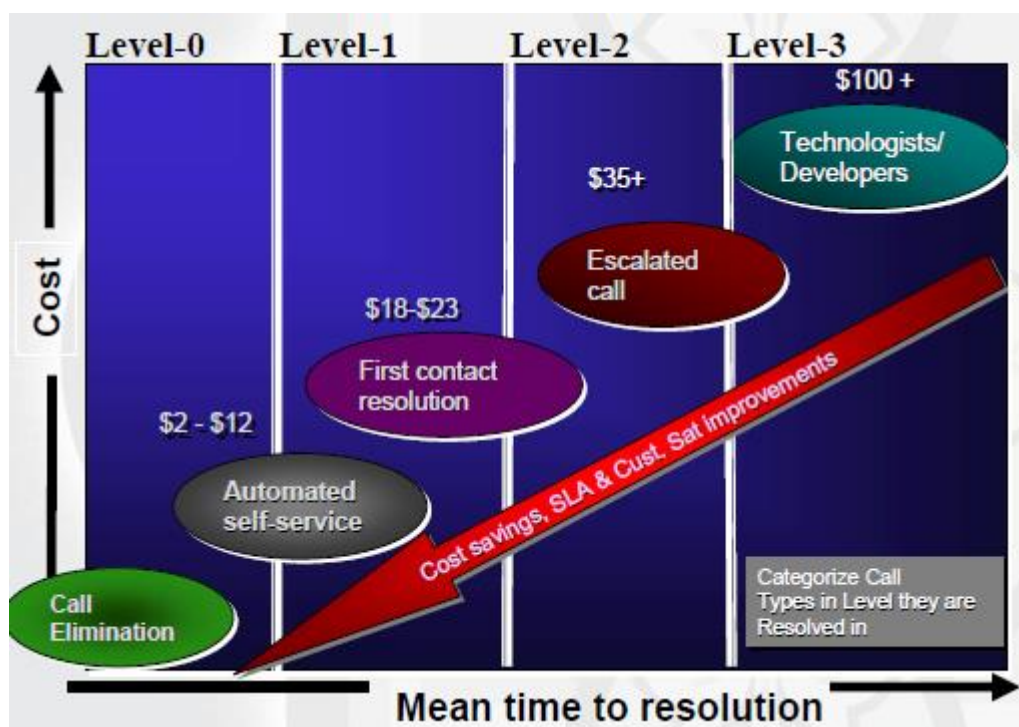
Atualmente os incidentes resolvidos no primeiro nível correspondem a cerca de 20% dos incidentes que são abertos, porém não se sabe ao certo o quanto a equipe ainda poderia ser explorada com pequenos ajustes. Segundo Delbridge et al. (2014), algumas companhias, também com ambientes complexos como este, chegam a taxas de 15% de resolução no nível 0 (ferramentas de auto-ajuda), 71% de resolução no primeiro nível e 11% resolvidos remotamente (FCR) pelo primeiro nível. Somente 3% são resolvidos no local.

Ao observar estas informações, verificou-se que poderiam existir muitas oportunidades de melhoria de eficiência no ambiente analisado onde, atualmente, são resolvidos 20% dos incidentes no primeiro nível de atendimento.

Sabe-se que, por falta de conhecimento, direito de acesso ou ferramentas necessárias, o *Service Desk* deixa de resolver incidentes nos quais seria capaz. Na maioria destes casos, a transferência de conhecimento é suficiente para que estes incidentes sejam resolvidos no primeiro nível. O objetivo do trabalho é identificar perfis de incidentes FTF e FCR de modo que se possa identificar uma linha de base de forma a estabelecer uma meta para o ano de 2015, ou seja, daqueles incidentes que não foram resolvidos no primeiro nível, quantos seriam candidatos a FTF ou FCR. Após isto, fazer uma rápida comparação de custo aproximado por incidente antes e após a meta atingida.

De acordo com a Figura 5.2, o custo de um incidente resolvido fora do *Service Desk* pode duplicar de valor a cada mudança de nível – no momento em que um incidente troca de um nível de suporte mais baixo para um nível mais elevado, nível 1 para nível 2 –, ou seja, quanto maior for o nível de atendimento, maior o custo.

Figura 5.2 - Níveis de serviço e o valor (estratégia *shift right-to-left*)



Fonte: McGarahan (2010)

A cada incidente que o *Service Desk* não consegue resolver, automaticamente o custo do serviço será maior a cada alteração de nível realizada.

5.2 ETAPA DE SELEÇÃO

Na primeira etapa do trabalho de conclusão, a proposta seria utilizar a base completa de 2013. Contudo, observou-se que o ambiente de Gerenciamento de Serviços passou por diversas mudanças, tanto de grupos, classificações, quanto de melhoria de maturidade, se comparado com 2014. Portanto, optou-se por utilizar todos os incidentes abertos na região Américas, de 1º de janeiro a 31 de julho de 2014. Desta forma, o estudo terá uma aplicação mais próxima do ambiente atual.

A etapa de seleção é propriamente obter os dados. Os dados utilizados na pesquisa foram adquiridos através da ferramenta de BI (*Business Intelligence*) já citada anteriormente, que é utilizada para calcular as métricas mensais das equipes, definidas de acordo com ITIL, pela companhia.

Por questões de segurança, previamente acordadas com a companhia, nenhuma informação, a respeito de fornecedores, valores ou sistemas, poderá ser divulgada. Alguns nomes foram editados, porém não removidos, para que fosse mantida a forma e o sentido original da base de dados.

Os 57 atributos, descritos no apêndice A, estão disponíveis no *software* de BI (*Business Intelligence*) utilizado pela companhia, SAP Business Objects, e podem ser exportados em arquivos no formato Microsoft Excel.

Com a base de dados completa e o objetivo da pesquisa estabelecido, novas reuniões com a equipe de *Service Desk* foram feitas e, concluiu-se que a análise de FTF e FCR necessitaria dos atributos do Quadro 5.1

Quadro 5.1 - Atributos selecionadas para o processamento inicial

Ticket ID	Location
Ticket Categorization	Country
Affected Service	VIP
Call Total Transfers	Resolving Group
Priority	Create Date
Resolution Code	Closed Date
Closure Code	Total Resolved Time (hrs)
OLA Status	Ticket Record Method

Fonte: Elaborado pelo autor

Inicialmente 58063 registros de incidentes estavam disponíveis na base de dados original. Destes, 57964 classificados como incidentes e 99 como problemas (ver seções 3.1 e **Erro! Fonte de referência não encontrada.**). Dos 57964 incidentes, grupo alvo da pesquisa, 949 (17%) foram resolvidos no primeiro nível e em menos de 10 minutos, considerados *FTF* (*First Time Fix*), e 3764 *FCR* (*First Call Resolution*), aqueles incidentes resolvidos no primeiro nível, abertos por telefone, porém, após o agente desligar o telefone. Como oportunidade, restaram 44251 incidentes, os quais não foram resolvidos pelo *Service Desk*.

A etapa principal desta parte do processo é selecionar os dados que serão utilizados na análise. A partir de conversas com o gerente de incidentes e do *Service Desk Global*, os atributos que influenciam na tomada de decisão para definir incidentes que podem ser resolvidos no primeiro nível são:

- *Total Resolved Time(hrs)*: Esta informação indica o tempo total desde a criação do incidente até a sua resolução (não encerramento). O valor é numérico, contínuo e deve ser positivo.

- *Call Total Transfers*: Indica quantas vezes o incidente trocou de grupo especialista. Por exemplo, foi aberto pelo *Service Desk*, transferido para um grupo especialista de 2º nível e voltou ao novamente ao primeiro nível, totalizando 2 transferências. O campo é composto por valores numéricos e discretos.

- *Priority*: Este atributo indica a prioridade definida para o incidente, de acordo com ITIL, a prioridade é definida a partir da quantidade de usuários afetados (impacto) e urgência. O valor é numérico e discreto, variando de 1 a 5, onde 1 é a maior prioridade.

5.3 PRÉ-PROCESSAMENTO

A etapa de limpeza e organização dos dados é demorada e intensiva, mas é um procedimento extremamente necessário para obtenção de resultados com sucesso.

O primeiro desafio surgiu na etapa de agrupamento dos dados. Como o servidor da aplicação de *Business Intelligence* fica hospedado em um *data center* na Malásia, os dados não poderiam ser adquiridos em um lote único para que não fossem corrompidos. Então, os dados foram adquiridos mês a mês, baseando-se na data de modificação. Em cada mês, foram removidos os *tickets* que não estavam com seu estado “encerrado”, para que não fossem repetidos nos meses subsequentes até que fossem encerrados.

Os dados foram finalmente agrupados em um arquivo do *Microsoft Excel* para que os processos seguintes fossem iniciados. A partir deste arquivo, após o final das etapas de pré-processamento, a planilha foi exportada para o formato CSV (*comma separated values* – “valores separados por vírgula” na tradução direta), formato este suportado pela linguagem R e Rattle.

Este tipo de formato foi escolhido pois é o mais comumente utilizado nas bibliografias disponíveis sobre o Rattle, além de gerar arquivos de menor tamanho se comparado aos do *Microsoft Excel*. A proporção de tamanho observada foi de 10 para 1, ou seja, um arquivo do *Excel* com 10MB, ao ser exportado para CSV, gera um arquivo de 1MB.

Também foram removidos acentos e caracteres especiais que estavam contidos nos valores de algumas classificações (*Ticket Categorization*) e localidades (*Location*). Eles foram substituídos pelas letras correspondentes, como por exemplo, “São Paulo” foi substituído por “Sao Paulo” e assim por diante.

5.3.1 Exclusões

Conforme conversa com o time que gerencia os indicadores e relatórios de desempenho de TI da companhia, alguns registros deveriam ser removidos da base pois são incidentes que devem ser desconsiderados. Um exemplo seria: um caso onde o usuário solicita manutenção de um ponto de energia elétrica danificado, que não pertence ao escopo de atendimento do *Service Desk* e de nenhuma outra equipe de TI.

Esta classificação é feita no primeiro nível, durante a validação do incidente, ou até mesmo na resolução. Foram identificados 3109 registros com estas características entre outras, portanto, dos 57964 incidentes, restaram 54855.

Os atributos que identificam os incidentes que devem ser desconsiderados são “*Resolution Code*”(Código de Resolução) e “*Closure Code*” (Código de encerramento), e os valores dos atributos que devem ser ignorados na análise estão destacados em negrito na Tabela 5.1.

Destes casos, por exemplo, *withdrawn* (“remover”, na tradução direta), é atribuído pelo analista para sinalizar que o incidente apresentou algum tipo de problema durante o andamento da resolução. Por exemplo, quando um usuário solicita a abertura de um incidente e por diversas circunstâncias desiste do atendimento por sua vontade, ou por falta de tempo, porque a solução não se aplica mais, ou por entrar em período de viagem ou férias. Ou seja, o *Service Desk* fica impedido de realizar qualquer ação, pois uma ação do usuário é necessária e o mesmo não se encontra disponível.

Tabela 5.1 – Valores das variáveis excluídas

Resolution Code	Closure Code
Access	Auto-Closure
Administration	Advice and Guidance
Advice and Guidance	Duplicate
Authentication Failure	First Time Fix
Capability Unit Resolution	Fault Fixed
Capacity Management	Unidentified Root Cause
Component Replacement	Out of Scope
Data	User Request to Close
Duplicate	Not Supported
Entitlement Failure	Administration
Error Control	Others
Fault Found	No Resolution
Knowledge	Authentication Failure
Knowledge Base Resolution	Permanent Solution
Known Error	Failed Change
Major Incident	Mislogged
Non Recurring	Other Fixes
Not Applicable	User Error
Other	Capacity Management
Out of Scope	Non Recurring
Permanent Resolution	Known Error
Problem Closed	Temporary Fix
Quality	Workaround Solution
Regress Change	Entitlement Failed
Regress Release	Hardware Error
Release Management	Release Management
Temporary Fix	Network Error
Unidentified Root Cause	Component Replaced
User Error	Not Applicable
User Request to Close	Field Resource Controller
	Existing Known Error
	Withdrawn
	New Known Error
	System Bug
	Major Incident

Fonte: Elaborado pelo autor

No caso de *Duplicate* (duplicado, na tradução direta), o incidente foi registrado pelo mesmo usuário ou mesmo sistema (no caso de eventos proativos), para o mesmo problema especificamente, registrado no mesmo período, portanto deve ser classificado como tal para que não seja considerado nas métricas mensais.

Mislogged (mal registrado na tradução direta - registrado incorretamente) pode acontecer caso o usuário não desejasse abrir um chamado, confundindo o endereço de e-mail pra abertura de chamados.

5.3.2 Valores ausentes

Os valores ausentes apresentam desafios para a mineração de dados e modelagem em geral. Existem muitos fatores que implicam na ausência de valores, incluindo o fato de que podem ter sido coletados erroneamente e por consequência estejam com valores ausentes ou pelo simples fato de que valores ausentes realmente podem fazer algum sentido no contexto das informações. Por isso o entendimento da base de dados é muito importante (WILLIAMS, 2011).

Por exemplo, caso um registro de incidente estivesse sem um valor na variável *Total Resolved Time (hrs)* este poderia significar zero, como poderia ocorrer em casos onde um incidente é registrado após o caso ter sido resolvido, o que pode ocorrer com as equipes de suporte local. Porém, de acordo com a análise feita e após conhecer a gama dos valores da base de dados, no caso desta variável específica, os incidentes conforme o caso acima estavam com valores “0,00”.

Os valores considerados como ausentes *Total Resolved Time (hrs)* foram 7 valores negativos encontrados. Estes foram tratados como exceção e removidos da base, pois se trata de uma variável crítica para obtenção de resultados precisos.

Alguns registros com valores ausentes foram identificados nos atributos selecionados, portanto, de acordo com os padrões da linguagem R e por consequência suportado pelo Rattle, eles foram substituídos pelos caracteres “NA”. Assim como valores ausentes, outros valores foram substituídos por “NA”, como valores que deveriam ser somente positivos porém continham valores negativos (no caso do atributo *Total Resolved Time(hrs)* citado acima, por exemplo) Outros, ainda, com valores diferentes das possibilidades disponíveis, também tiveram seus valores substituídos por “NA”.

Através da função “Mice” (versão 2.8) para exploração de dados, disponível no Rattle, foi possível resumir as variáveis que possuem valores ausentes e que foram inicialmente selecionadas para prosseguir com as análises, ilustradas na Tabela 5.2.

Tabela 5.2 – Resumo de valores ausentes obtidos através da função Mice

	53138	1423	1	292	
Ticket.ID	1	1	1	1	0
Affected.Service	1	1	1	1	0
Ticket.Categorization	1	1	1	1	0
Priority	1	1	1	1	0
Resolution.Code	1	1	1	1	0
Closure.Code	1	1	1	1	0
OLA.Status	1	1	1	1	0
VIP	1	1	1	1	0
Create.Date	1	1	1	1	0
Total.Resolved.Time..hrs.	1	1	1	1	0
Ticket.Record.Method	1	1	1	1	0
CallTT	1	1	1	1	0
Resolving.Group	1	1	0	1	1
Location	1	1	1	0	292
Country	1	0	1	0	1715
	0	1	1	2	2008

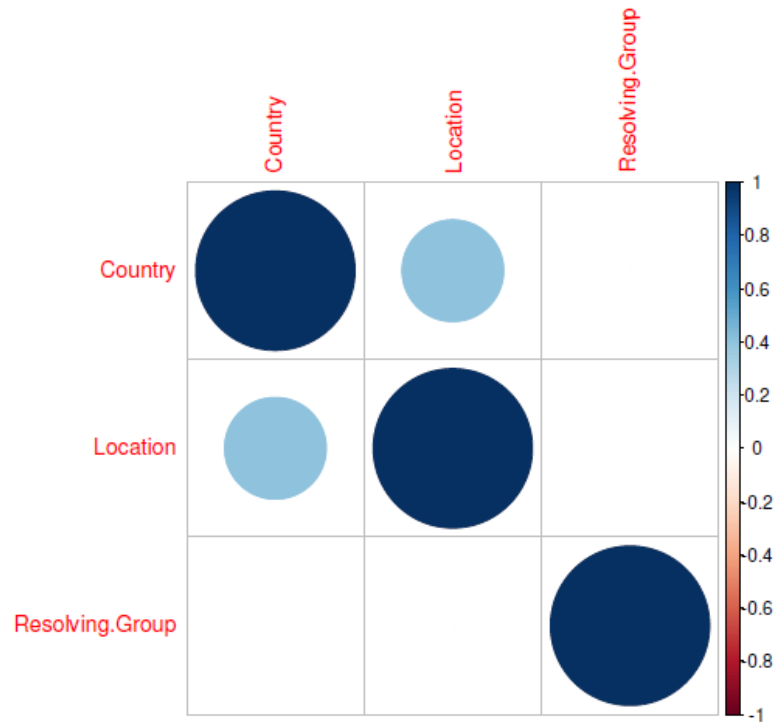
Fonte: Elaborado pelo autor através do Rattle

A tabela mostra, para cada variável, um padrão de valores ausentes. Dentro da tabela, o número 1 indica um valor presente, enquanto 0 indica um valor ausente (WILLIAMS, 2011). As colunas registram o número de observações que combinam o padrão correspondente de valores ausentes, ou “NA”. Existem 53138 observações sem valores ausentes (cada uma que possui valor 1 na coluna correspondente), enquanto 1423 valores ausentes na variável “*Country*” exclusivamente, e destes, 292 também ausentes em “*Location*”, e 1 observação com valor ausente em “*Resolving Group*”.

As colunas e linhas estão ordenadas em ordem crescente de acordo com a quantidade de valores ausentes. Então, geralmente, a primeira coluna registra o número de observações nas quais não existem valores ausentes. A última coluna registra o número de valores ausentes de todo o conjunto de dados para cada variável. O total de todos os valores ausentes fica no canto inferior direito (2008, neste caso).

Além da função “Mice”, também pode-se observar a correlação dos valores ausentes utilizando o algoritmo de Pearson, confirmando a análise anterior, conforme a Figura 5.3.

Figura 5.3 – Correlação de valores ausentes utilizando o algoritmo de Pearson



Fonte: Elaborado pelo autor através do Rattle

Quanto maior a escala exibida na direita, maior o grau de correlação entre as variáveis. Nota-se que a variável “*Resolving Group*”, portanto, possui valores ausentes, porém sem correlação com “*Location*” e “*Country*”. Já “*Location*” e “*Country*”, que possuem valores ausentes, compartilham alguns entre si com menor número em *Location* (292 conforme a análise anterior, correspondente ao círculo menor, de tonalidade clara).

5.4 TRANSFORMAÇÃO DE ATRIBUTOS

O formato “CSV” foi escolhido como método padrão de entrada de dados para a análise suporta basicamente dois tipos de tipos de dados: nominal e numérico. Valores dispostos em letras, ou *strings*, são convertidos em categóricos. Valores numéricos podem ser convertidos tanto como categóricos, quanto numéricos (ou ainda normalizados) dependendo do objetivo da análise a ser aplicada (WITTEN, 2011, p. 56).

Por exemplo, um atributo que contém valores numéricos pode ser convertido em uma escala ordinal, utilizando uma simples comparação de maior ou menor que um determinado valor.

Alguns valores numéricos, no entanto, não tem sentido de serem utilizados, por exemplo o número de identificação de um chamado. Ou seja, não faz sentido comparar se é menor do que um valor ou maior do que outro. Estes devem ser convertidos em variáveis categóricas

(WITTEN, 2011, p. 57). No Rattle, tratando-se deste caso, há uma opção para tipos de variáveis únicas chamado de “Ident”, a qual deve ser selecionada para que o atributo seja considerado um campo categórico. É importante manter este campo, para que haja, futuramente a possibilidade de analisar o incidente, caso necessário.

É muito comum também utilizar atributos que contém valores ordinais como nominais. De fato, alguns algoritmos de aprendizado de máquina, como por exemplo, regras de associação, tratam somente elementos nominais (WITTEN, 2011, p. 57).

As variáveis alvo que contém valores numéricos, tiveram estes valores transformados em discretos e foram utilizados os limiares inicialmente definidos pela governança de FTF/FCR e fornecidos pela equipe de gerenciamento de incidentes. As variáveis numéricas foram discretizadas conforme a Tabela 5.3.

Tabela 5.3 – Valores utilizados para discretização dos atributos

Atributo	Regra	Valor nominal	Consideração
<i>Total Resolved Time (hrs)</i> para TTR_Category	≤ 2	BAIXO	considerou-se uma situação ótima onde o incidente foi resolvido ainda no mesmo turno ou ainda, ao telefone (atendendo, em parte, requisitos de FTF)
	≤ 8	MEDIO	considerou-se que o incidente seria resolvido no mesmo dia de trabalho (atendendo aos requisitos de FCR)
	≤ 24	ALTO	considerando que o incidente seria resolvido em no máximo 1 dia após ter sido aberto
	> 24	MUITO_ALTO	esta categoria foi incluída para que, se necessário, possa ser realizada uma análise sobre incidentes com altos tempos de resolução
<i>Call Total Transfers</i> para CallTT_Category	≤ 1	BAIXA	Incidente resolvido pelo <i>Service Desk</i> (0 transferência) ou pelo segundo nível (1)
	≤ 3	MEDIA	Incidente resolvido com ajuda de um <i>Resolving Group</i> global de segundo nível
	≤ 5	ALTA	Incidente resolvido com ajuda de um <i>Resolving Group</i> global e outro(s) local(is)
	> 5	MUITO_ALTA	Esta categoria foi incluída para que, se necessário, possa ser realizada uma análise sobre incidentes com muitas transferências
<i>Priority</i> transformado em Priority_Category	1 ou 2	ALTA	Esta prioridade se aplica para incidentes com vários usuários afetados ou de grande impacto no negócio
	3	MEDIA	Incidentes com prioridade média seriam aqueles que estariam com prioridade baixa, porém com algum impacto sobre poucos usuários
	4 ou 5	BAIXA	Este seria o cenário onde a maioria dos casos de FTF e FCR se encaixariam, ou seja, impacto sobre um usuário e urgência média ou baixa

Fonte: Elaborado pelo autor

Árvores de decisão utilizando o algoritmo *Ctree* (Figura 5.4) foram geradas a fim de compreender as melhores distribuições de limiares para a discretização dos valores.

Além das variáveis anteriormente citadas, criou-se também um atributo categórico chamado “FTF.FCR” o qual indica, a partir das premissas levadas em conta pelo Service Desk, se o incidente foi FTF, FCR ou nenhum dos dois, utilizando as regras da Tabela 5.4.

Tabela 5.4 – Condições utilizadas para discretização do atributo FCR.FTF

Atributo	Regra	Valor nominal
FCR.FTF	Atributo "Ticket Categorization" inicia com "FTF."	FTF
	Atributo "Resolving Group" contém valor = "BAT-Servicedesk", "End Market Stack", "BAT-CENTRE OF EXCELLENCE", "FUJITSU 3rd Party", "BAT-AdminTeam", "BAT PSE" ou "BAT-Admin-IAM" & Atributo "Total Call Transfers" contém valor ≤ 3 & Atributo "Ticket Categorization" não inicia com "FTF."	FCR
	Os que não se encaixam nas regras anteriores	NO

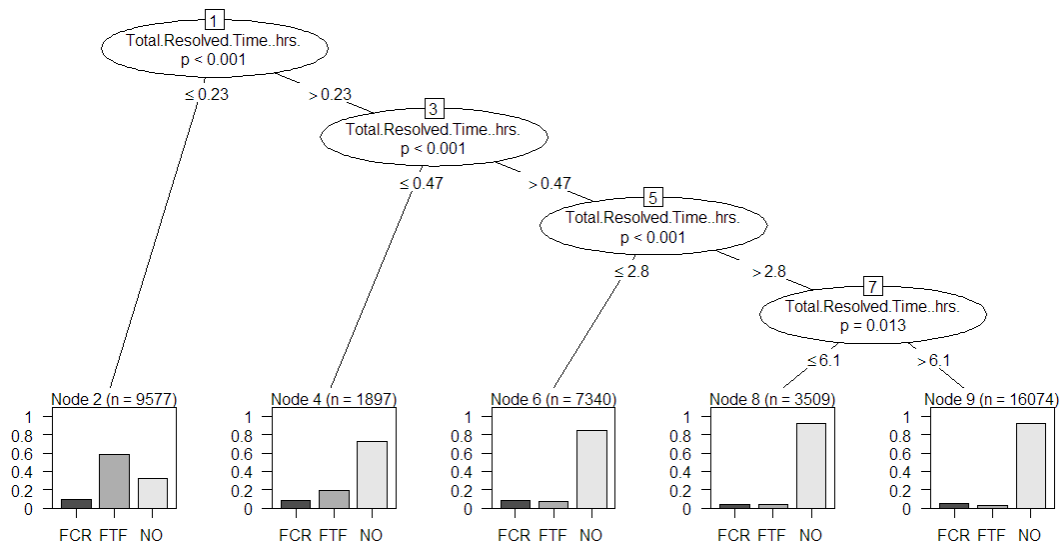
Fonte: Elaborado pelo autor

Ou seja, este é um mapeamento do que já é feito pelo Service Desk, sendo que a área de oportunidade se concentra nos valores “NO” deste atributo, isto é, os incidentes que não foram resolvidos no primeiro nível. Este atributo foi analisado juntamente com os demais para que assim se pudesse chegar a um valor próximo do real e de acordo com a governança de FTF e FCR.

5.4.1 Transformação da variável *Total Resolved Time(hrs)*

A fim de melhorar a divisão entre os limiares de “Total Resolved Time (hrs)”, foi gerada uma árvore de decisão utilizando o algoritmo *Ctree*, configurado como variável de entrada “*Total Resolved Time(hrs)*” e FCR.FTF como variável *target* (“meta”, na tradução literal) e obteve-se a árvore de decisão mostrada na Figura 5.4.

Figura 5.4 - Árvore de decisão Ctree da variável Total Resolved Time(hrs)



Fonte: Elaborado pelo autor a partir do Rattle

Os resultados gerados pela árvore de decisão mostraram que grande parte dos incidentes com características de FTF estão abaixo de 0,23h, ou seja, 13,8 minutos, resultado qual confirma a governança estabelecida. Já os registros com características de FCR se concentram abaixo de 6,19h. Não se pode afirmar que os valores de FCR são o intervalo acima de 0,23h e abaixo de 6,19h, pois existem outras variáveis que devem ser analisadas, como prioridade, total de transferências de um grupo para outro, características estas que são consideradas para determinar se o registro é um candidato a FCR ou não. Por exemplo, pode haver um incidente resolvido em menos de 0,23h, Não FTF/FCR, mas que tenha sido transferido para mais de 3 grupos (fora da governança de FCR), ou seja, o atributo *Call Total Transfers*, além do *Total Resolved Time(hrs)*, também é exclusivo.

O Quadro 5.2 mostra alguns detalhes dos nós de cada divisão gerada pelo algoritmo o qual gerou a árvore mostrada na Figura 5.4.

Quadro 5.2 – Resumo do modelo de classificação árvore de decisão Ctree para o atributo *Total Resolved Time(hrs)*

<p>Summary of the Conditional Tree model for Classification (built using 'ctree'):</p> <p>Conditional inference tree with 5 terminal nodes</p> <p>Response: FTF.FCR Input: Total.Resolved.Time..hrs. Number of observations: 38397</p> <p>1) Total.Resolved.Time..hrs. \leq 0.23; criterion = 1, statistic = 1045.986 2)* weights = 9577</p> <p>1) Total.Resolved.Time..hrs. $>$ 0.23 3) Total.Resolved.Time..hrs. \leq 0.47; criterion = 1, statistic = 96.542 4)* weights = 1897</p> <p>3) Total.Resolved.Time..hrs. $>$ 0.47 5) Total.Resolved.Time..hrs. \leq 2.8; criterion = 1, statistic = 48.677 6)* weights = 7340</p> <p>5) Total.Resolved.Time..hrs. $>$ 2.8 7) Total.Resolved.Time..hrs. \leq 6.1; criterion = 0.987, statistic = 8.668 8)* weights = 3509</p> <p>7) Total.Resolved.Time..hrs. $>$ 6.1 9)* weights = 16074</p>
--

Fonte: Elaborado pelo autor a partir do Rattle

O Quadro 5.2 exibe basicamente as mesmas informações do modelo de classificação por árvore de decisão mostrado na Figura 5.4 para caráter demonstrativo.

A partir destes resultados, fez-se uma releitura dos valores de *Total Resolved Time(hrs)* para que fossem gerados resultados de forma mais eficiente sobre os quais o *Service Desk* já executa no dia-a-dia, de acordo com a Tabela 5.5, alterando as faixas exibidas previamente na Tabela 5.3.

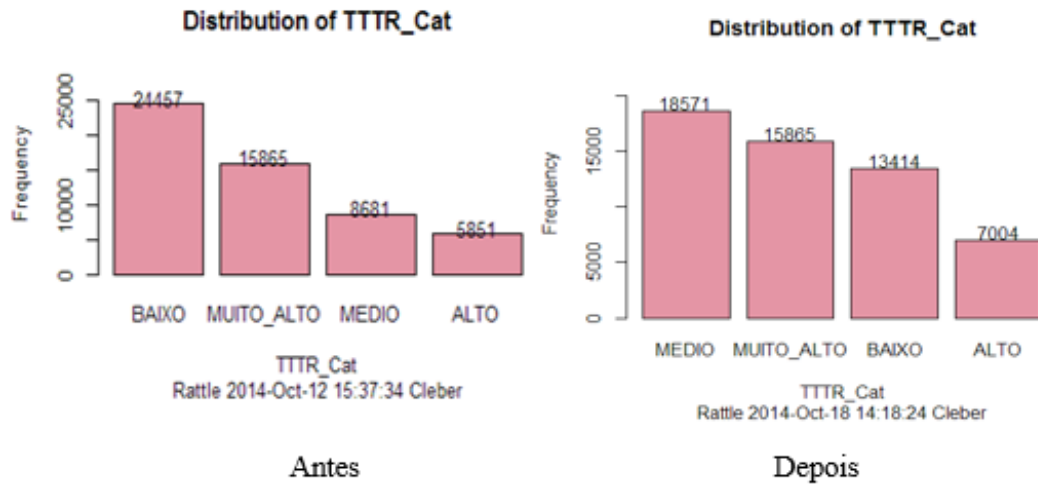
Tabela 5.5 – Valores de *Total Resolved Time(hrs)* discretizados para TTTR_Cat

Atributo	Regra ANTES	Regra DEPOIS	Valor nominal
<i>Total Resolved Time (hrs) para TTTR_Category</i>	≤ 2	$\leq 0,23$	BAIXO
	≤ 8	$\leq 6,19$	MEDIO
	≤ 24	≤ 24	ALTO
	> 24	> 24	MUITO_ALTO

Fonte: Elaborado pelo autor

A nova distribuição de frequência dos limites do atributo “TTTR_Cat”, aplicando o novo limite observado na árvore de decisão e na Tabela 5.5, pode ser comparado na Figura 5.5 como também na Figura 5.6.

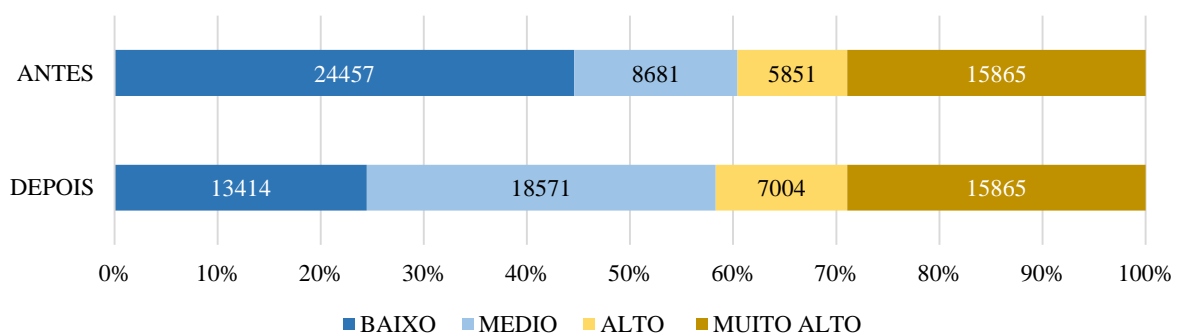
Figura 5.5 - Distribuição de frequências antes e depois do ajuste dos limites de TTTR_Cat



Fonte: Elaborado pelo autor através do Rattle

Observando a Figura 5.5, gerada pelo Rattle, nota-se que muitos registros passaram da categoria BAIXO para MEDIO, ou seja, 11043 registros tem valores de *Total Resolved Time(hrs)* entre 0,23 e 2h. Outros 1153 incidentes saíram da categoria MEDIO para ALTO, estes com valores de *Total Resolved Time(hrs)* entre 6,19 e 8h. Estas diferenças podem ser melhor observadas na Figura 5.6 Existe ainda uma pequena diferença, de 54 registros que foram adicionados à categoria ALTO.

Figura 5.6 – Comparação da distribuição de frequências do atributo TTTR_Cat antes e depois do ajuste



Fonte: Elaborado pelo autor

Para TTTR_Category = BAIXO, existe uma chance maior do registro ser candidato a FTF, sendo que para TTTR_Category = MEDIO, maior é chance de ser candidato a FCR. Estes valores estão de acordo com as características modificados a partir da árvore de decisão e com

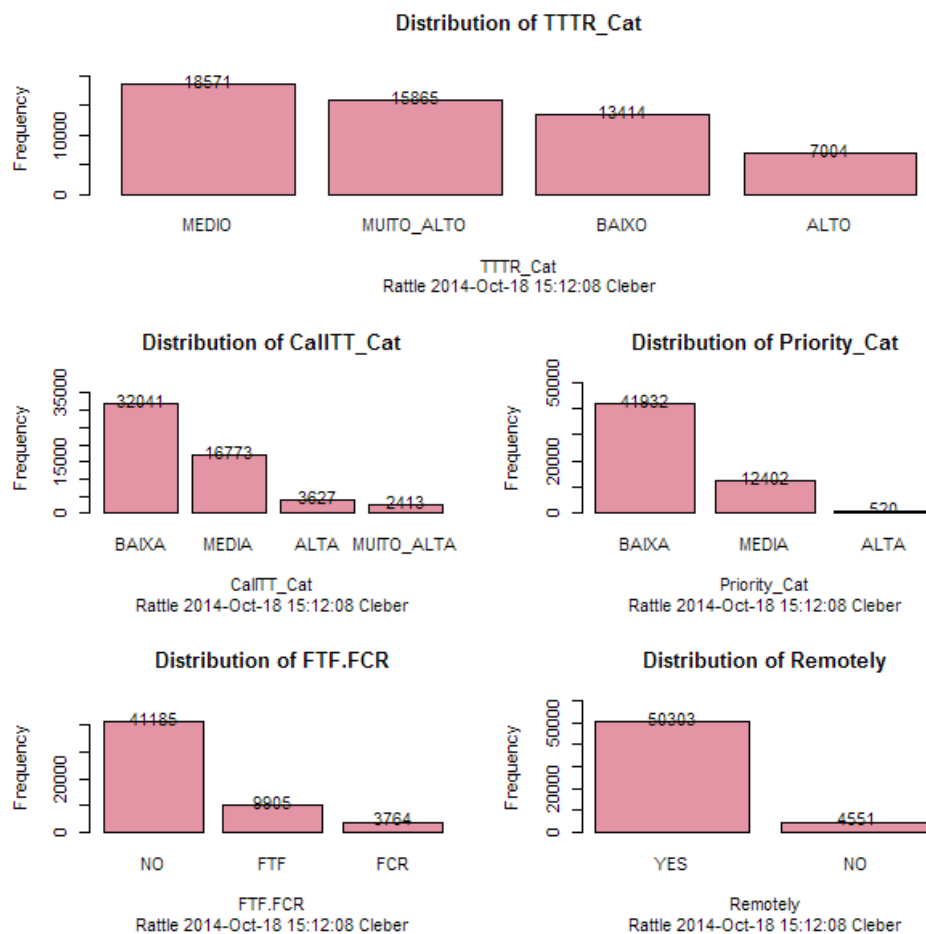
a governança de gerenciamento de incidentes da companhia, relacionadas ao tempo de resolução.

5.4.2 Descrição dos dados

Após a etapa de transformação dos dados concluída, iniciou-se então a criação dos modelos de dados. Porém, antes disto foi necessário conhecer o significado de cada atributo e a influência do mesmo para a interpretação correta dos resultados que serão gerados.

Com o intuito de satisfazer a condição de trabalhar somente com valores nominais, imposta por alguns algoritmos, criaram-se novos atributos discretizados conforme detalhado na seção anterior. Os principais atributos utilizados para as análises foram TTTR_Cat, CallTT_Cat, Priority_Cat, FTF.FCR, conforme os gráficos de distribuição de frequências mostrados na Figura 5.7.

Figura 5.7 – Distribuição de frequências dos atributos discretizados



Fonte: Elaborado pelo autor

Além dos atributos já descritos na seção 5.4, pode-se observar um novo atributo criado, chamado “*Remotely*”. Este atributo foi criado para sinalizar se o incidente teria condições de ser resolvido remotamente pelo *Service Desk* através de ferramentas de acesso remoto, e-mail ou telefone. Ou seja, mesmo que o cenário dos demais atributos indique um candidato a FTF ou FCR, deve-se considerar se o analista do *Service Desk* poderá resolvê-lo sem que seja necessário estar fisicamente presente e que tenha direitos de acesso às ferramentas necessárias para solução do problema.

Para a criação deste atributo, cujos valores são do tipo Booleano (*YES/NO*), foi verificado se no atributo “*Ticket Categorization*” continha as seguintes palavras-chave:

- “*Data Cent*”, “*Server*” e “*Infrastructure*”: Estas categorias foram excluídas dos candidatos a FTF e FCR devido às características dos incidentes não corresponderem à governança estabelecida, já que exigem acesso a configurações avançadas de sistemas como *storages*, servidores, sistemas de *backup*, configuração de memória em servidores, etc.; Foram ocultadas as letras finais de “*Data Centre*” pois também existem categorias de incidentes nomeadas utilizando o inglês americano, “*Data Center*”.

Exemplo de valores excluídos das possibilidades para o atributo “*Ticket Categorization*”:

- Infrastructure.Server.Windows.Others;
- Data Centre.Tier 2/3.DC Database.SQL;
- Infrastructure. Directory Services. AD. Server. Disk Utilisation Alert;
- Infrastructure.Server.Windows.Others;

Para os registros os quais continham pelo menos uma das palavras acima, foi atribuído “NO” ao valor do atributo “*Remotely*”, sinalizando que não é possível resolvê-lo remotamente pelo *Service Desk*, como pode ser visto na Tabela 5.6.

Tabela 5.6 – Descrição dos atributos categóricos

Atributo	Níveis	Valor mais frequente
Ticket Record Method	4	E-mail
Ticket Categorization	1778	Application.Lotus Notes/Domino
Affected Service	113	Local Support – Brazil
Resolution Code	18	Problem Closed
Closure Code	22	Auto-Closure
OLA Status	2	Within OLA
Location	195	Rio de Janeiro
Country	21	Brazil
VIP	2	No
Resolving Group	286	<i>Brazil Hands and Feet/Field Service Support</i>
TTTR_Cat	4	BAIXO
CallTT_Cat	4	BAIXA
Priority_Cat	3	BAIXA
FTF.FCR	3	NO
Remotely	2	NO

Fonte: Elaborado pelo autor

Para as demais categorias, exceto *Ticket ID* que é um atributo de identificação, pode-se observar na Tabela 5.7 onde são mostradas algumas características dos atributos numéricos, assim como na Tabela 5.6, onde estão descritas de forma resumida algumas características dos atributos categóricos.

Tabela 5.7 - Descrição dos atributos numéricos

Atributo	Mínimo	Mediana	Média	Máximo
Call Total Transfers	0	1	1,79	43
Priority	1	4	4,8	5
Total Resolved Time (hrs)	0	3	41,82	2662,43
Created Month	1	4	4,056	7
Created Week	1	17	16,29	23

Fonte: Elaborado pelo autor

5.5 APLICANDO TÉCNICAS DE MINERAÇÃO DE DADOS

Assim que o conjunto de dados passou por todos os processos iniciais, como seleção, limpeza e transformação, iniciou-se mais uma das etapas de KDD, a etapa de aplicação das técnicas de mineração de dados. Esta etapa tem como saída os padrões observados após a aplicação das técnicas escolhidas, padrões estes que poderão ser analisados através de gráficos, regras de associação e árvores de decisão.

Nas próximas subseções serão detalhados os algoritmos usados com foco nos atributos principais: *Ticket.Record.Method*, *TTTR_Cat*, *CallTT_Cat*, *Priority_Cat*, *RG.SD*, *Remotely*, *FTF.FCR*, ressaltando a importância de cada um deles para com o objetivo deste trabalho.

5.5.1 Escolha das tarefas e algoritmos

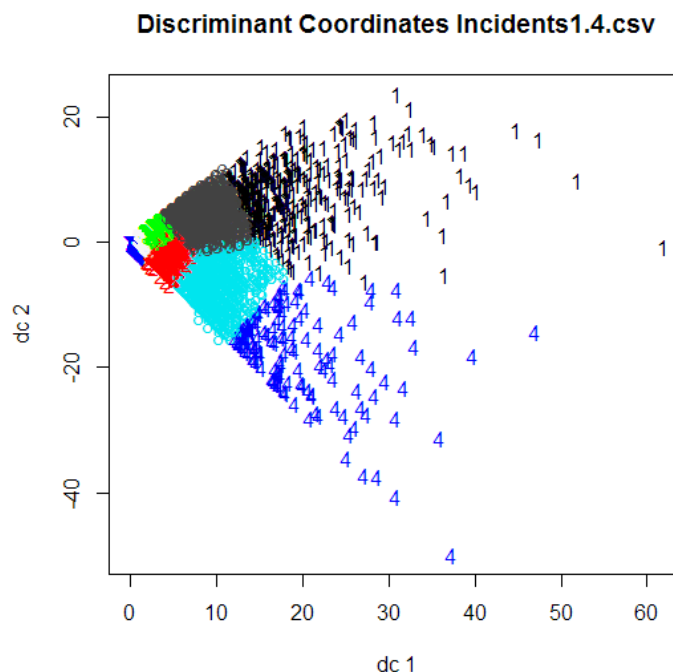
No decorrer dos estudos de cada técnica de mineração de dados, alguns testes foram realizados com diversos atributos, a fim de testar o desempenho de cada um com os diversos algoritmos existentes.

Inicialmente, técnicas de agrupamento foram aplicadas utilizando o algoritmo K-Means. Como o algoritmo trabalha, por padrão, com atributos numéricos e as análises dependem basicamente de um atributo categórico (FTF.FCR), foi possível executá-lo utilizando somente 3 dos 7 atributos essenciais, *CallTT*, *TTTR* e *Priority*. O resultado da execução do algoritmo gerou grupos de incidentes relacionados a infraestrutura de servidores, telefonia, sistemas legados, como também FTFs e FCRs pois os valores dos atributos eram semelhantes.

Como se pode observar na Figura 5.8, os grupos ficaram visualmente bem definidos, porém, a combinação com os demais atributos continha incidentes que não poderiam ser considerados candidatos a FTF ou FCR, juntamente com outros que correspondiam ao perfil.

Enfim, se os atributos categóricos não fossem comparados nas análises, o resultado procurado não ficaria consistente, ou seja, um grupo de candidatos a FTF ou FCR adequado depende da combinação com os demais atributos categóricos.

Figura 5.8 – Grupos gerados com o algoritmo K-Means exibidos como coordenadas discriminantes



Fonte: Elaborado pelo autor através do Rattle

Iniciou-se então um estudo sobre algoritmos de árvores de decisão e regras de associação, pois estes algoritmos trabalham especificamente com atributos categóricos. Foram utilizados os algoritmos Apriori para regras de associação e Ctree para árvores de decisão.

Os resultados obtidos a partir destes algoritmos serão detalhados nas próximas subseções.

5.5.2 Árvores de decisão

As árvores de decisão são técnicas frequentemente utilizadas em mineração de dados. Os modelos gerados, a partir destas técnicas, são tão utilizados principalmente por serem de fácil compreensão. O algoritmo que foi explorado para que os modelos pudessem ser gerados foi basicamente o Ctree (que gera árvores de inferência condicional).

É aplicável a todos os tipos de problemas de regressão, incluindo nominal, ordinal, numérico, censurado, bem como variáveis de resposta multivariada e escalas de medição arbitrárias dos covariantes (HOTHORN et al, 2006).

A aplicação das árvores de inferência condicional, no contexto deste trabalho, ocorreu no suporte à definição dos limites (numéricos) que foram utilizados para a validação dos resultados de FTF e FCR. A partir das análises das árvores de inferência condicional, foram definidos os limites de TTTR daqueles incidentes já resolvidos como FTF e FCR. Isto foi

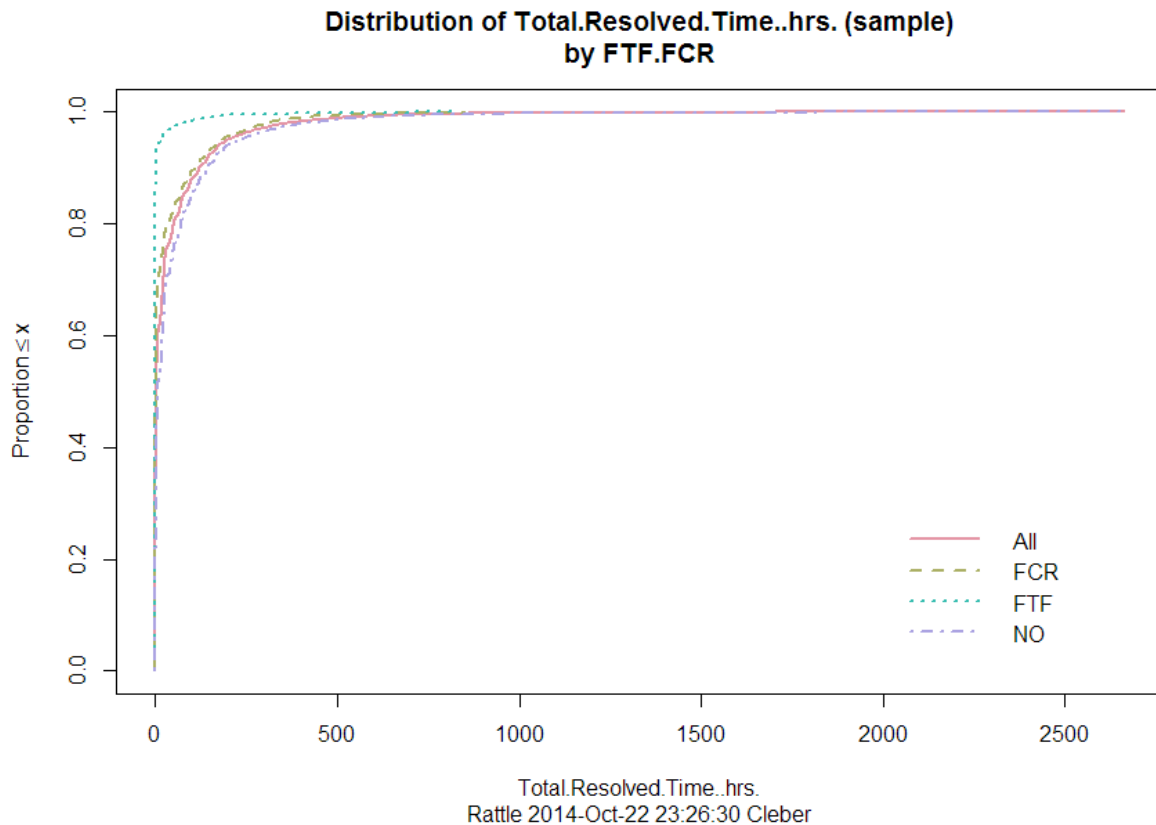
necessário para a criação de um atributo de TTTR categórico (TTTR_Cat) e que, desta forma, permitisse a análise utilizando regras de associação.

Assim que os valores foram descobertos, foram validados pelo gerente de incidentes através da análise individual de diversos incidentes que seriam candidatos a FTF e FCR e continham o valor da categoria adequada (TTTR_Cat=BAIXO e TTTR_Cat=MEDIO). Aqueles com TTTR_Cat=ALTO e MUITO_ALTO, também foram validados e descartados como candidatos neste momento. A faixa de valores dos limiares definidos pelas árvores se encontra na Tabela 5.5 (subseção 5.4.1).

A categoria de TTTR foi necessária pois, tanto incidentes FTF quanto FCR, se baseiam no tempo de resolução, este que deve ser estar adequado tanto para as expectativas do *Service Desk* quanto para usuário final. Mesmo que o tempo seja relativamente simples de ser estabelecido para FTF (em torno de dez minutos, definido na governança), para FCR o mesmo não ocorre, pois também se baseia em outros atributos, tais como *Call Total Transfers* e *Ticket Record Method e Remotely*. Como o intervalo de valores de *Total Resolved Time(hrs)* para incidentes FTF e FCR vai de 0 a mais de 2500 horas, conforme pode ser observado na Figura 5.9, utilizou-se esta técnica para que se pudesse alcançar valores adequados na prática.

Na Figura 5.9, encontra-se um gráfico onde é exibido o tempo de resolução por tipo de incidente FTF, FCR ou nenhum dos dois (“NO”). No eixo *x* estão os valores do atributo *Total Resolved Time (hrs)* e no eixo *y* a proporção de registros (em escala de 0 a 1, ou 0 a 100%) por tipo.

Figura 5.9 – Proporção de TTTR para os incidentes FTF, FCR, NO e All



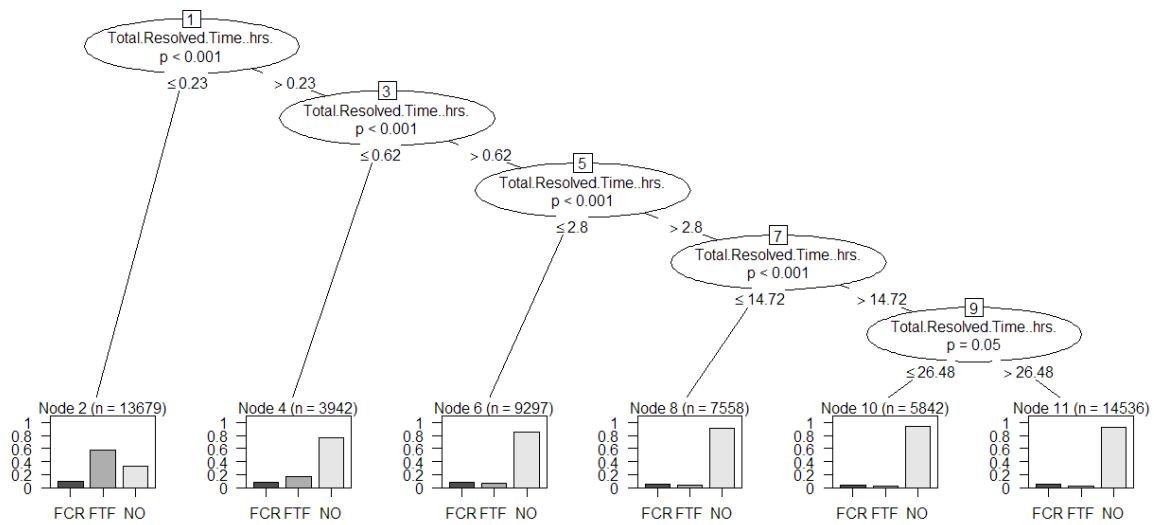
Fonte: Elaborado pelo autor

Pode-se observar que os valores de TTTR na curva FCR se distanciam de zero após $y=0,7$ (70%), enquanto para FTF, em sua maioria, ficam próximos a 0 até $y=0,95$, ou seja, mais de 95% dos incidentes FTF possuem TTTR próximos a zero.

Como também se pode observar no gráfico da Figura 5.9, em torno de 60% dos registros que não foram FTF nem FCR (Curva “NO”) possuem valores de TTTR próximos a zero, representando as oportunidades procuradas por este trabalho.

Buscando identificar estes tipos de registro, foi gerada uma árvore de inferência condicional com auxílio do *Ctree*. A árvore exibida na Figura 5.10 mostra que os incidentes que possuem TTTR menor ou igual que 0,23h (nodo 2), em torno de 50% são FTF, uma pequena parte é FCR e mais de 30% são incidentes normais, não resolvidos pelo *Service Desk*. No nodo 3, onde a segunda divisão é feita, um novo teste define que, para incidentes com TTTR menor que 0,62h e maiores que 0,23h, em torno de 20% são FTF, aproximadamente 10% FCR e em torno de 70% não foram resolvidos pelo *Service Desk*. As divisões seguem assim por diante, até 26,48h no nodo 9, onde ainda existem incidentes FTF e FCR.

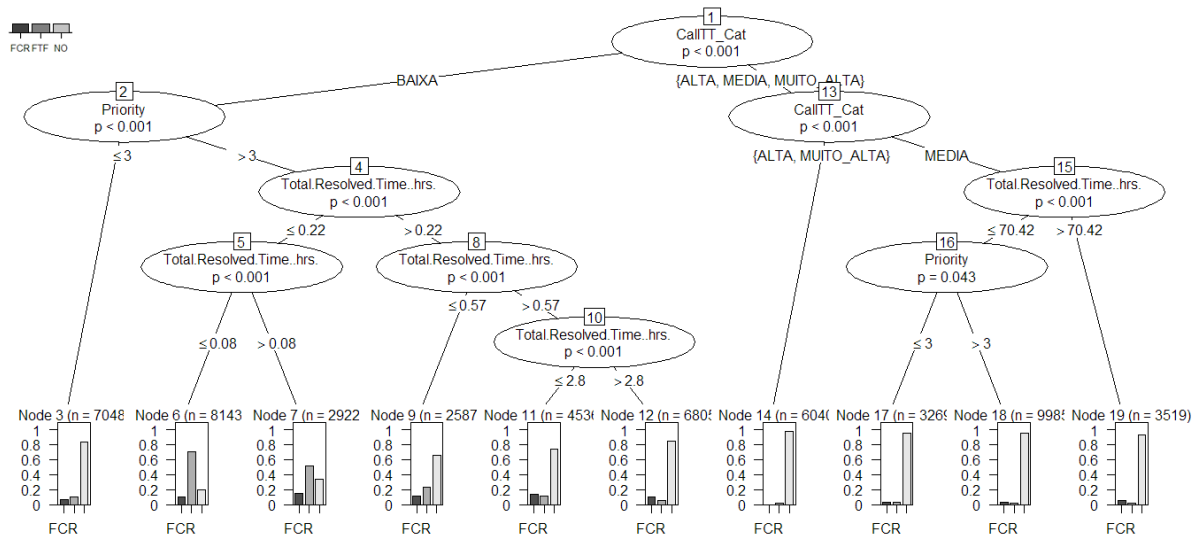
Figura 5.10 – Árvore de inferência condicional: TTTR para FCR.FTF



Fonte: Elaborado pelo autor através do Rattle

Na Figura 5.11, foram adicionados outros atributos relevantes para a análise de oportunidades, como *Call Total Transfers (CallTT_Cat)*, *Priority*, além de TTTR.

Figura 5.11 – Árvore de inferência condicional: CallTT_Cat, Priority e TTTR para FTF.FCR

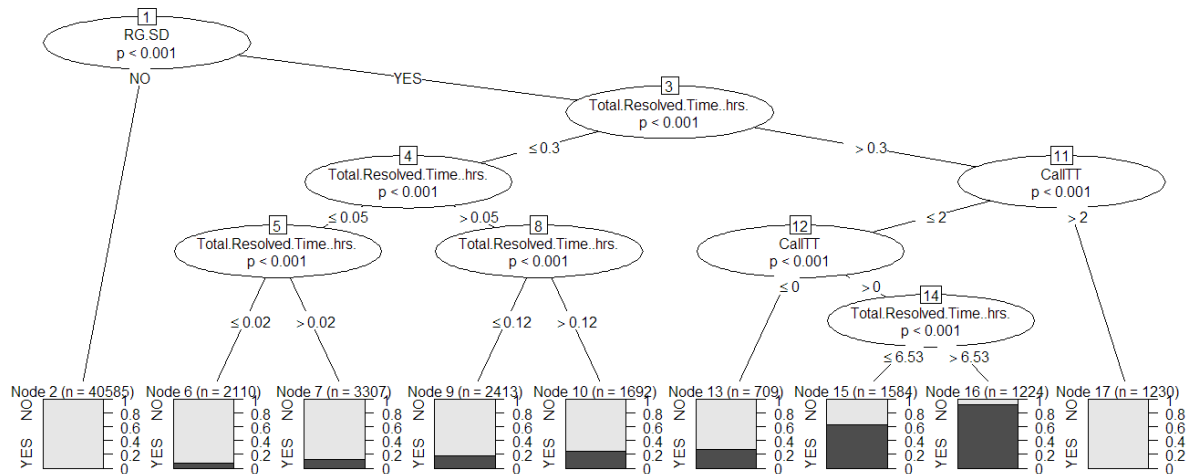


Fonte: Elaborado pelo autor

É possível identificar com mais clareza o perfil de incidentes FTF, que tendem a ter $CallTT_Cat = BAIXA$ (nodo 1), $Priority > 3$ (nodo 2) e $TTTR \leq 0,22$ (nodo 4), alguns maiores que 0,22 e poucos com mais de 2,8h (nodo 10). Para FCR, pode-se observar um perfil semelhante, porém com TTTR tendendo a ser maior que 0,57 (nodo 8) e menor ou igual a 2,8h, lembrando que na sua grande maioria, a prioridade é maior que 3 e a categoria de CallTT é BAIXA.

No caso dos incidentes FCR, a árvore de inferência condicional explicou o perfil dos incidentes, conforme a Figura 5.12.

Figura 5.12 – Árvore de inferência condicional – RG.SD, CallTT e TTTR para FCR



Fonte: Elaborado pelo autor

Nela se pode observar que os incidentes FCR têm valores de CallTT (numérica) ≤ 2 e TTTR, em sua maioria, $\leq 6,53$. Aqueles incidentes que tem TTTR maior que isto, conforme observado no nó 14, possuem CallTT menores ou iguais a 2 (nó 12). Como os incidentes FTF tem um perfil melhor traçado pelo atributo TTTR, eles acabam ocultando os números de FCR, que tem uma maior variação de valores de TTTR. Para isso, foram criados dois atributos adicionais, nos quais RG.SD¹(booleano) indica se o incidente foi resolvido pelo *Service Desk* no primeiro nível e FCR (booleano) que indica se o incidente é FCR ou não. Após análise com estes atributos, o algoritmo teve melhores resultados de TTTR, gerando uma árvore pequena e de fácil compreensão, conforme a Figura 5.12.

As árvores de decisão ilustradas nesta seção, construídas através do algoritmo Ctree, conseguem ilustrar perfeitamente os limiares que os atributos categóricos devem possuir, baseados naqueles incidentes que o *Service Desk* já resolveu. Isto garantiu que as análises

¹ RG.SD (*Resolver Group.Service Desk*) = "Yes"/"No". Este atributo indica se o *Service Desk* foi quem resolveu o incidente. Caso RG.SD="YES" (*Service Desk* resolveu) e o incidente não é FTF ou FCR, há grandes chances de ser um candidato caso estabeleça uma combinação nas regras de associação.

realizadas com as regras de associação (descritas na próxima seção) gerassem candidatos (nem FTF, nem FCR) que correspondessem ao mesmo padrão de incidentes já resolvidos como FTF e FCR.

5.5.3 Regras de associação

A etapa de aplicação de regras de associação iniciou-se assim que a transformação e análise dos atributos envolvidos foram concluídas. Nesta última etapa, os resultados foram obtidos através das regras de associação para que assim fosse possível uma análise detalhada dos incidentes candidatos a FTF e FCR atributo a atributo e valor por valor. A linguagem R, assim como o Rattle, suporta esta etapa utilizando o algoritmo Apriori, um dos mais utilizados e conhecidos em mineração de dados.

Sabe-se que esta etapa também poderia ter sido executada utilizando árvores de decisão. Basicamente, através da combinação do valor individual de cada nível dos atributos categóricos, também seria possível chegar às oportunidades buscadas por este trabalho. Como o número de atributos envolvidos e os níveis dentro de cada um deles é relativamente grande, a árvore de decisão obtida em testes feitos no Rattle, teria em torno de 53 níveis ou mais, e nem todas as combinações necessárias seriam compreendidas pela árvore.

As primeiras execuções do algoritmo foram realizadas ainda na etapa de transformação dos atributos, o que permitiu validar o desempenho do algoritmo já com os valores atribuídos a eles. Durante esta etapa observou-se a necessidade de dividir o atributo “FTF.FCR” que, inicialmente tinha três níveis de valores a fim de detalhar e facilitar a análise dos perfis dos incidentes: FCR, FTF e NO. Ele foi separado em outros dois atributos, chamados “FCR” e “FTF”, cada um com valores booleanos (YES/NO). Cabe salientar que, quando os valores dos atributos “FCR” e “FTF” forem iguais a “NO”, significa que o registro não foi resolvido no primeiro nível de atendimento e que, portanto, estas são as áreas de oportunidade procuradas neste trabalho. Nesta etapa verificou-se que os incidentes que não foram resolvidos no primeiro nível como FTF ou FCR representam 41185 (cerca de 75% do total), de janeiro a julho de 2014.

Uma vez que os atributos foram escolhidos e observados os parâmetros de cobertura e confiança mínimos, uma combinação será formada e uma regra gerada. Desta forma, a simples mudança do valor de um dos atributos indicaria uma combinação diferente, e assim, uma possibilidade (regra) diferente. Por exemplo, em uma situação ótima para candidatos a FTF, a regra gerada seria formada a partir da combinação de todos os atributos e valores exibidos no Quadro 5.3.

Quadro 5.3 – Combinação de atributos e seus valores para potenciais oportunidades de FTF

```
{Ticket.Record.Method=Telephone,
TTTR_Cat=BAIXO,
CallTT_Cat=BAIXA,
Priority_Cat=BAIXA,
RG.SD=NO,
Remotely=YES} e
{FTF.FCR=NO}
```

Fonte: Elaborado pelo autor através do Rattle

A combinação do Quadro 5.3 poderia ser diferentemente interpretada caso o atributo CallTT_Cat fosse igual a “MEDIA” (2 ou 3 transferências entre grupos de resolução). Ainda assim, seria uma área de oportunidade para FTF, mas ao mesmo tempo existe o desafio de transferência de conhecimento de um grupo de segundo nível para o *Service Desk*, pois haveria a necessidade de evitar uma transferência de grupo (CallTT) adicional. Outra possibilidade para a combinação com “CallTT_Cat” = “MEDIA” seria FCR, porém, como o “TTTR_Cat” = BAIXO (menor ou igual que 0,23h), a melhor escolha seria FTF. Lembrando a governança de FCR permite até 2 transferências entre grupos, ilustrado pela Figura 5.12.

A cobertura (*support*) definida para que um conjunto de regras adequado fosse gerado foi de 2% e a confiança (*confidence*) de 5%. Estes números foram atingidos após diversas execuções e análises das regras geradas pelo algoritmo, nas quais se observou que, a partir de 2% de cobertura eram geradas regras interessantes para o cumprimento do objetivo deste trabalho.

Utilizando os atributos listados no Quadro 5.3, foram geradas 70 regras de associação, exibidas no apêndice B. as quais tiveram as características, do ponto de vista de cobertura e confiança, exibidas na Tabela 5.8.

Tabela 5.8 – Características de cobertura e confiança das regras de associação

support	Confidence
Min. :0.02284	Min. :0.1179
1st Qu.:0.03528	1st Qu.:0.7096
Median :0.04234	Median :0.9249
Mean :0.05645	Mean :0.8030
3rd Qu.:0.06361	3rd Qu.:0.9845
Max. :0.13036	Max. :1.0000

Fonte: Elaborado pelo autor através do Rattle e algoritmo Apriori

5.5.3.1 Análise das regras de associação obtidas

Na Tabela 5.9 são exibidas as regras selecionadas a partir dos critérios e atributos relevantes com o propósito de melhorar a eficiência no *Service Desk*. Elas serão descritas de acordo com sua importância sob o ponto de vista do gerenciamento de incidentes.

Tabela 5.9 - Regras de associação selecionadas

Ident	lhs (antecedente ou <i>Left Hand Side</i>)	rhs (consequente ou <i>Right Hand Side</i>)	support (Cobertura)	confidence (Confiança)
10	{Ticket.Record.Method=Telephone, TTTR_Cat=MEDIO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.10646443	0.9859868
17	{Ticket.Record.Method=E-mail, TTTR_Cat=MEDIO, CallTT_Cat=MEDIA, Priority_Cat=BAIXA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.06360521	0.9923208
31	{Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.04397127	0.9962825
44	{Ticket.Record.Method=Telephone, TTTR_Cat=MEDIO, CallTT_Cat=BAIXA, Priority_Cat=MEDIA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.04017939	0.9782512
52	{Ticket.Record.Method=E-mail, TTTR_Cat=ALTO, CallTT_Cat=MEDIA, Priority_Cat=BAIXA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.03527546	0.9933265
59	{Ticket.Record.Method=Telephone, TTTR_Cat=ALTO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.02554053	0.9831579

Fonte: Elaborado pelo autor através do Rattle e algoritmo Apriori

- Regra 10

A regra 10 define que 10,6% dos registros da base possuem os valores dos antecedentes e do consequente combinados. De todos os registros que possuem os valores dos atributos antecedentes, 98,5% deles também possuem “FTF.FCR” = “NO” (consequente).

Esta regra foi selecionada pois trata de características exclusivas de incidentes FCR, os quais não foram resolvidos no primeiro nível. O atributo “*Remotely=YES*” informa que, de acordo com a categorização do incidente, ele pode ser solucionado remotamente. Esta regra cobre 5840 de 41185 incidentes que poderiam ter sido resolvidos no primeiro nível através de uma ferramenta de acesso remoto, caso necessário intervenção no equipamento do usuário.

A implementação desta regra, assim como as demais, exigirá um esforço de cada equipe de segundo nível para analisar os incidentes marcados como candidatos a FCR, de forma a definir os processos executados para solução dos incidentes, como também executar um processo formal de transferência de conhecimento para a equipe de gestão do conhecimento do *Service Desk*. Como as características combinam com as premissas de FCR, ou seja, o tempo não é tão agressivo (“*TTTR_Cat*” = “MEDIO”) quanto um perfil de incidente FTF (“*TTTR_Cat=BAIXO*”, onde $BAIXO \leq 0,23h$), um menor esforço será necessário, pois o tempo já corresponde ao esperado. O que reforça esta oportunidade é o valor do atributo “*CallTT_Cat=BAIXO*”, informando que foi necessário no máximo 1 transferência entre equipes para que o incidente fosse resolvido.

Há chances ainda, em alguns casos, que estes candidatos possam se tornar candidatos a FTF, ou até mesmo passível de automatização do atendimento através de centrais de resposta por voz ou portais de autoajuda, onde o usuário teria acesso às ferramentas necessárias para a solução de um problema específico, uma vez que se consiga fazer com que não haja transferências entre grupos de resolução.

- Regra 17

A próxima regra selecionada, 17, representa 3489 incidentes, nos quais as características se adequam ao perfil de candidatos a FCR ou nível 0 (portal de autoajuda). Esta regra diz que, em 6,36% (*support* = 0.063) dos registros da base de incidentes, os atributos e respectivos valores antecedentes e consequentes aparecem combinados. De todos os antecedentes, 99,23% (*confidence* = 0.9923) possuem “FTF.FCR” = “NO” (consequente). Uma questão a ser analisada posteriormente é o modo de abertura do incidente. Por ter sido comunicado pelo usuário através de e-mail (“*Ticket Record*

Method=E-mail”), o mesmo pressupõe, implicitamente, que existe uma baixa urgência na resolução do incidente e que, se uma ferramenta de apoio ou portal de informações estivesse disponível, ele mesmo poderia buscar a solução para o problema e resolvê-lo. Isto pode ser concluído, pois o esforço necessário para buscar uma solução e escrever o e-mail é relativamente parecido, na prática. Neste caso, as oportunidades serão consideradas com perfil FCR, com destaque para as possibilidades de implementação, ainda, através de um portal de autoajuda (nível 0).

Outro motivo pelo qual o processo de abertura de um incidente por e-mail se torna mais demorado, é porque todos eles são revisados por um grupo de analistas *Service Desk*, assim que o mesmo é registrado no sistema. O processo de registro de um incidente, quando o usuário envia e-mail, é feito automaticamente e pode demorar até 2 horas para ser revisado.

Por definição, o *Service Desk* atua remotamente somente se o usuário entra em contato através do telefone. No caso de um incidente FCR ou FTF, não interagir com o usuário, através do telefone, evita insatisfação por parte do mesmo, caso esteja em reunião, em férias, entre outras situações em que o *Service Desk* poderia criar uma situação incômoda ou insatisfatória.

- Regra 31

A regra 31 corresponde a todos os requisitos de um incidente candidato a FTF, pois o incidente foi comunicado através de uma ligação do usuário, teve seu tempo de resolução menor que 0,23h, 0 ou 1 transferência entre equipes, prioridade baixa, não foi resolvido no primeiro nível e poderia ter sido resolvido remotamente. Esta regra representa um menor esforço de implementação por parte das equipes em tornar 2412 incidentes candidatos a FTF, pois todos os atributos atendem às premissas. Um dado importante, que reforça a possibilidade destes incidentes tornarem-se FTF, é o IMF (índice de satisfação do usuário – ver p. 29). Analisando os 2412 incidentes candidatos, observou-se que o índice é de 92% de satisfação com o atendimento do chamado em geral. A taxa de respostas dos usuários é de aproximadamente 20%, ou seja, 20% dos usuários respondeu ao questionário de IMF.

- Regra 44

A regra 44 atende aos requisitos de FTF, pois as características dos incidentes possuem CallTT=BAIXO, podem ser resolvidos remotamente, TTTR=MEDIO, foram registrados por telefone, porém a prioridade é média. Neste caso, uma revisão de prioridades será necessária a fim de verificar realmente quais incidentes tem mais de um usuário impactado ou até mesmo maior impacto no trabalho do usuário. O que explicaria, neste

caso, uma prioridade média, são incidentes relacionados ao serviço de e-mail da companhia, pois caso este não funcionar corretamente, prejudicaria diretamente o trabalho do usuário. Através de uma validação feita diretamente na base de dados, verificou-se que mais de 50% destes incidentes candidatos correspondem a uma categoria relacionada ao serviço de e-mail, o que reforça esta afirmativa.

Dentre os incidentes não-FTF ou FCR, 41185 foram analisados e, dentre estes, 9329 (22%) podem ser considerados como *quick wins*² (“ganho rápido” na tradução direta), pois atendem às especificações necessárias para que sejam incidentes FCR.

Nos demais 31856 incidentes, 2412 atendem aos requisitos para candidatos a FTF, representando cerca de 6% do total de incidentes não-FTF ou FCR analisados.

- Regras 52 e 59

As demais regras, 52 e 59, atendem aos requisitos de FCR, porém não correspondem a oportunidades de melhoria direta, pois o tempo de atendimento necessário foi entre 6.19 e 24h. Através de melhorias de processo e transferência de conhecimento, seria possível chegar ao tempo determinado para atendimento em primeiro nível. Estas regras representam 3336 incidentes.

Conforme o Quadro 5.4, através das regras de associação, foi possível identificar 17279 oportunidades adicionais de melhoria na eficiência do *Service Desk*, além daqueles incidentes que já são resolvidos pelo mesmo. Estes incidentes possuem perfil para solução no primeiro nível e representam cerca de 42% daqueles que não eram resolvidos pelo *Service Desk*.

² Segundo Magalhães (2007), *quick wins* são metas que podem ser alcançadas em um curto espaço de tempo, garantindo entrega periódica de resultados para a organização.

Quadro 5.4 – Resumo de tipo de oportunidades de melhoria por regra

Regra	Cobertura (%)	Número de Registros	Tipo de oportunidade
Regra 10	10.65%	5840	FCR <i>quick win</i>
Regra 17	6.36%	3489	FCR <i>quick win</i>
Regra 31	4.40%	2412	FTF <i>quick win</i>
Regra 44	4.02%	2204	FTF
Regra 52	3.53%	1935	FCR
Regra 59	2.55%	1401	FCR
Total	31,5%	17279	-

Fonte: Elaborado pelo autor

Além de fornecer uma visão geral sobre os demais candidatos que não foram selecionados neste momento, as regras geradas são de fácil compreensão.

5.6 AVALIAÇÃO DOS RESULTADOS

A mineração de dados aplicada neste estudo demonstrou ser muito eficiente na obtenção de resultados. Isto se deve às diversas ferramentas disponíveis para aplicação de diversas técnicas com diferentes propósitos, como também processos bem definidos e uma extensa bibliografia acessível ao usuário.

O Rattle, assim como a linguagem R, na qual a ferramenta é baseada, demonstrou ser funcional em todos os recursos que foram necessários durante os testes e aplicação dos algoritmos. A interface é intuitiva e ideal para usuários iniciantes no campo de mineração de dados. Devido à complexidade na linguagem de programação R, o Rattle se tornou imprescindível para a obtenção dos resultados.

Através das etapas descritas anteriormente, como limpeza da base de dados, transformação de atributos, seleção, análise de árvores de indução condicional e regras de associação, chegou-se a 42% de incidentes candidatos para melhoria de eficiência do *Service Desk*, completamente descritos no que diz respeito à priorização de implementação, desde os mais rápidos (*quick wins*) até os que exigirão esforço adicional das equipes. Juntamente com os incidentes já executados como FTF e FCR, esta linha de base propõe que o *Service Desk* tem condições de chegar a 56% de resolução no primeiro nível, 31,5% a mais do que o número atual.

A partir da sinalização dos candidatos, os mesmos serão separados por equipe e categoria, para então passarem por uma análise profunda de forma a documentar os processos necessários para a solução de cada tipo de incidente no primeiro nível de atendimento. O

processo será guiado a partir da equipe de gerenciamento de incidentes e terá um prazo de aproximadamente 1 (um) ano até que os perfis de FTF e FCR sejam implementados.

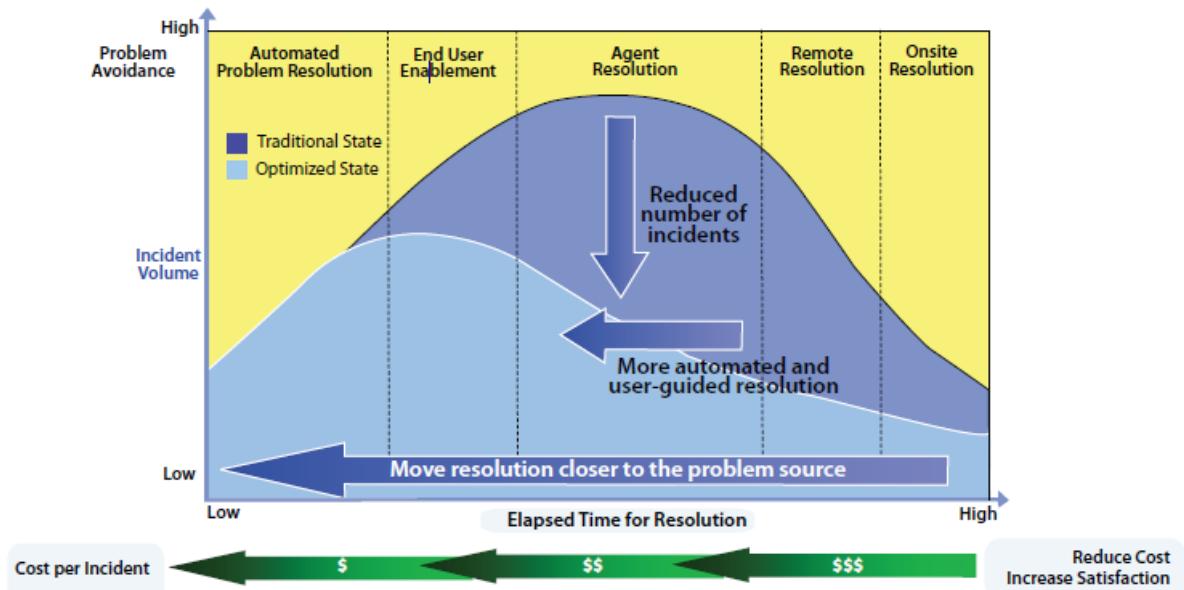
Através do modelo gerado para transformação dos atributos, também será possível analisar, mês a mês, quais incidentes o *Service Desk* poderia ter resolvido durante a ligação (FTF) ou após algum tempo de pesquisa com auxílio de uma equipe interna de atendimento remoto (FCR), de modo a manter a eficiência através deste processo básico de melhoria contínua.

5.6.1 Redução de custos

Segundo Rumburg (2010), a IBM, um dos principais fornecedores de suporte a usuários de TI na América do Norte, o índice de resolução de incidentes no primeiro nível é uma das métricas mais importantes no que diz respeito à redução de custos e retorno de investimento de implementação e manutenção do *Service Desk*. Ainda segundo um artigo publicado pela IBM em junho de 2011, esta estratégia, chamada de “*shift right-to-left*”, tem como objetivo transferir continuamente os altos custos de um modelo de atendimento local, no qual um analista de suporte local se desloca até o usuário (custo elevado), para o modelo do *Service Desk*, melhor estruturado e orientado a custo.

Segundo a mesma fonte, esta estratégia também proporciona melhoria de produtividade do usuário, melhora a satisfação, reduz custos trabalhistas e viagens (pois reduz a necessidade de analistas de suporte local), elimina ligações, fazendo uso de tecnologias disponíveis para a aproximação do analista de primeiro nível com o usuário. No modo de suporte no local, segundo o artigo, é o modo de custo mais elevado para resolução de incidentes, além de ter maior impacto negativo na produtividade do usuário, conforme ilustrado na Figura 5.13 (*onsite resolution*).

Figura 5.13 – Ilustração da estratégia *shift right-to-left*



Fonte: IBM (2011)

A proposta principal desta pesquisa é trazer o grande volume de incidentes resolvidos por equipes de suporte locais para o *Service Desk*, chamados de FTF (*agent resolution*) ou *remote resolution*, nome dado ao que referimos como FCR. Conforme a Figura 5.13, existem casos em que será possível mover a resolução mais à esquerda ainda, para o nível *end user enablement* (“capacitação do usuário” – tradução nossa), ou seja, disponibilizar o conhecimento necessário para a resolução de problemas comuns diretamente ao usuário, seja através de um portal ou através de informativos e ferramentas de autoajuda.

Segundo Rumburg (2011), para efeitos de estimativa de redução de custos, o valor pago para cada *ticket* atendido por um analista de suporte local na América do Norte é de \$62, aproximadamente 150 reais. Para os incidentes solucionados pelo primeiro nível (*Service Desk*), o custo é de \$22, aproximadamente 50 reais, ou seja, cerca de 35% do custo do suporte com atendimento no local.

Em termos práticos, o cenário de custo atual para suporte de TI, na região Américas, conforme os números da companhia em questão, poderia ser ilustrado da seguinte forma:

- Custo de incidentes atualmente resolvidos no segundo nível:
 $41185 \times R\$150 = R\$ 6.177.750$ Por mês: $R\$ 6.177.750 / 7 \text{ meses} = \mathbf{R\$ 882.535}$
- Custo de incidentes resolvidos no primeiro nível:
 $13669 \times R\$50 = R\$ 683.450$
 Por mês: $R\$ 683.450 / 7 \text{ meses} = \mathbf{R\$ 97.635}$

- Total: R\$ 6.861.200, ou **R\$ 980.171 por mês**

Utilizando os mesmos números, de forma a simular a economia caso os candidatos a FTF e FCR já estivessem implementados, o cenário seria o seguinte:

- Custo de incidentes resolvidos no segundo nível:
 $41185 (-) 17279 \text{ (candidatos)} = 23906 \times R\$150 = \mathbf{R\$ 3.585.900}$, ou **R\$ 512.271 por mês**, economia de aproximadamente 42% no custo de suporte de segundo nível, ou R\$ 370.264 por mês.
- Custo de incidentes resolvidos no primeiro nível:
 $13669 \text{ atuais} + 17279 \text{ candidatos} = 30948 \times R\$50 = \mathbf{R\$ 1.547.400}$
 Por mês: R\$221.057, incremento proporcional ao número de incidentes de 85%.
- Total do custo de suporte de TI: $R\$ 1.547.400 + R\$ 3.585.900 = \mathbf{R\$5.133.300}$, redução de 25% no custo total de suporte de TI.

O total estipulado de economia caso os candidatos já estivessem implementados é de R\$ 1.727.900 em 7 meses, ou R\$ 246.842 por mês, ou seja, uma redução de aproximadamente 25% do custo atual.

O custo é baseado na conversão da média de custo de atendimento de incidentes na América do Norte e representa um custo completo, incluindo salário e benefícios dos analistas, salários e benefícios das pessoas indiretamente ligada (por exemplo, agente de viagem, secretária), gastos com instalações (desgaste de carro, aluguel, depreciação de equipamentos), seguro, treinamentos, telecomunicações, *internet*, licença de *software*, equipamentos (*notebooks*, computadores), materiais de escritório, etc.

O total de economia pode ser ainda maior, sabendo que o custo de um incidente incrementa a cada mudança de nível em um processo no qual o incidente é submetido, ou seja, o custo de cada equipe de segundo nível na qual o incidente necessita alguma ação, é cumulativo (Rumburg, 2011). Como as características de FTF e FCR preveem até 1 transferência, o custo fica limitado ao *Service Desk* e não aumenta pois não são necessários diferentes níveis de suporte.

Além de redução de custos, a estratégia *shift right-to-left* também traz outros benefícios, tais como:

- Melhoria da percepção da TI por parte dos usuários, pois a TI fica mais eficiente;
- Aumenta a produtividade dos usuários, pois os incidentes demoram menos de 10 minutos para serem resolvidos;

- Traz uma abordagem proativa de análise de incidentes, levando à redução de incidentes relacionados a interrupções ou degradação na qualidade dos serviços;
- Força a documentação de processos, fazendo com que a TI fique mais eficiente uma vez que os processos podem ser compartilhados e são frequentemente revisados;
- Encoraja os *stakeholders* a investirem em novas tecnologias, pois demonstra o comprometimento da área de TI com o negócio.

6 CONCLUSÃO

Este trabalho manteve o foco em descobrir estratégias que possam melhorar a eficiência de atendimento de um *Service Desk*. O *Service Desk* do presente estudo está embasado nas melhores práticas de ITIL e, para descoberta destas estratégias, realizou-se um estudo de técnicas e ferramentas de mineração de dados, conhecimento dos dados e processos envolvidos para descoberta de conhecimento.

Foram abordados processos gerais de mineração de dados, utilizando a linguagem R e a interface gráfica Rattle, para encontrar oportunidades de melhoria de eficiência do *Service Desk*, analisando uma base de gerenciamento de incidentes e indicadores de desempenho.

O Rattle mostrou ser muito fácil de utilizar, o que garantiu que o objetivo proposto fosse atingido mesmo que o autor não tivesse qualquer experiência anterior com linguagens aplicadas a mineração de dados. Acredita-se que isto possa encorajar novos “mineradores” a trazer desafios do dia-a-dia de forma a buscar a análise e resolução dos mesmos por meio da tecnologia de mineração de dados e a linguagem R.

Um dos maiores desafios encontrados neste trabalho foi desvendar o que estava por trás de cada técnica e algoritmo de mineração de dados, de modo a selecionar o mais adequado em relação aos tipos de atributos disponíveis na base de dados e, desta forma, tirar total proveito dos resultados fornecidos pelos algoritmos. Outro desafio, que correspondeu a cerca de 70% do tempo empenhado nas análises, foi a etapa de pré-processamento dos dados. Esta etapa foi repetida diversas vezes até que a base tivesse uma consistência aceitável, para que então se obtivesse o melhor resultado.

Devido a uma completa documentação, livros sobre o tema, e os colaboradores online (fóruns de discussão, exemplos) da linguagem R, assim como a interface fácil e amigável de utilizar do Rattle, foi possível uma rápida curva de aprendizado e chegar ao entendimento de cada etapa com clareza.

Por fim, foram obtidas experiências com a linguagem R, totalmente novas para o autor. Através da exploração da base de dados usando histogramas, gráficos de caixa e gráficos dinâmicos disponíveis no Rattle, também foi gerado conhecimento sobre as características da base de dados de incidentes, a relação entre os seus atributos e os processos envolvidos com o *Service Desk*.

Após a aplicação das árvores de decisão e regras de associação, identificou-se mais de 40% de oportunidades nas quais o *Service Desk* tem condições de atender, aplicando-se a

estratégia *shift right-to-left*. Estas descobertas, após implementadas, corresponderão em aproximadamente R\$240 mil reais por mês de economia no custo dos serviços de TI, conforme foi estipulado.

A partir dos resultados deste trabalho e das oportunidades encontradas, sugere-se que uma meta seja estabelecida para estes sejam implementados de forma planejada e em um tempo previamente estipulado, para que assim se garanta o processo de melhoria contínua mencionado pela ITIL.

Além do conhecimento gerado, que possibilitará a aplicação das técnicas em problemas que não estraram no foco deste estudo, foram elucidadas oportunidades antes não tangíveis pelos gestores, trazendo à tona a importância da análise de incidentes e a busca constante de oportunidades de melhoria no serviço e no valor da TI como um todo, e, principalmente, quando possível, chegar à redução de custos mensuráveis.

REFERÊNCIAS BIBLIOGRÁFICAS

ABBOTT, Martin L.; FISHER, Michael T. **The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise**. United States: Addison-Wesley, 2009.

BRAGA, Luis Paulo Vieira. **Introdução à Mineração de dados**. 2ª ed. Rio de Janeiro: E-Papers Serviços Editoriais, 2005. 212 p.

CARVALHO, Luís A. V. de. **Datamining – A mineração de Dados no Marketing, Medicina, Economia, Engenharia e Administração**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2005. 225 p.

CHAKRABARTI, Soumen et al. **Data Mining: Know it all**. United States: Morgan Kaufmann, 2008. 460 p.

DELBRIDGE, Dr. Linda S., et al. **Taking end user services to the next level with IBM's Right-to-Left strategy**. NY, USA Jan. 2014. Disponível em <<http://public.dhe.ibm.com/common/ssi/ecm/en/enw03003usen/ENW03003USEN.PDF>>. Acesso em 25/09/2014.

FAYYAD, Usama M. et al. **Advances in Knowledge Discovery and Data Mining**. England: AAAI/MIT Press, 1996. 611 p.

GREGO, Maurício. Conteúdo digital dobra a cada dois anos no mundo. **Editora Abril S/A**, São Paulo, 09 abr. 2014. Disponível em <<http://exame.abril.com.br/tecnologia/noticias/conteudo-digital-dobra-a-cada-dois-anos-no-mundo>>. Acesso em: 26 out. 2014.

HAN, Jiawei; KAMBER, Micheline; PEI, Jian. **Data Mining Concepts and Techniques**. 3ª ed. United States: Morgan Kaufmann, 2011. 703 p.

HOTHORN, Torsten; HORNIK, Kurt; ZEILEIS, Achim. **Unbiased Recursive Partitioning: A Conditional Inference Framework**. Journal of Computational and Graphical Statistics. Vol. 15. 2006. P. 651 a 674.

IBM Business Analytics. **Why SPSS software?** Disponível em <<http://www-01.ibm.com/software/analytics/spss/>>. Acesso em: 28 set. 2014.

MADRUGA, Roberto. **Gestão Moderna de Call Center & Telemarketing**. São Paulo: Atlas, 2006.

MAGALHÃES, Ivan Luizio. **Gerenciamento de Serviços de TI na prática: uma abordagem com base na ITIL**. São Paulo: Novatec Editora, 2007. 667 p.

MCGARAHAN, Peter. **First Contact Resolution: The Performance Driver!** McGarahan & Associates: United States. Disponível em <<http://www.bsmreview.com/firstcontactresolution.shtml>>. Acesso em 25/09/2014.

MELLENDEZ, Rubem. **Service Desk Corporativo: Solução com Base na ITIL V3**. São Paulo: Novatec Editora, 2011.

OGC - Office of Government Commerce. **Service Operation**. United Kingdom: The Stationary Office (TSO), 2007. 263 p.

PICCOLI, Gabriele. **Essentials of Information Systems for Managers**. United States: Chapman & Hall/CRC, 2012, 423 p.

RUMBURG, Jeff. **The True Cost of Desktop Support: Understanding the Critical Cost Drivers of Desktop Support**. MetricNet: United States, 2011. Disponível em <[http://www.hdiconference.com/conference/~//media/HDICConf/2012/Files/WhitePapers/Rumburg - True Cost of Desktop Support.pdf](http://www.hdiconference.com/conference/~//media/HDICConf/2012/Files/WhitePapers/Rumburg-TrueCostofDesktopSupport.pdf)>. Acesso em 25/09/2014.

TAN, Pang; STEINBACH, Michael; KUMAR, Vipin. **Introdução ao Data Mining Mineração de Dados**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2009.

TORGO, Luís. **Data Mining with R: Learning with Case Studies**. United States: Chapman & Hall/CRC, 2010, 277 p.

THE R FOUNDATION FOR STATISTICAL COMPUTING. **R Project**. Austria. Disponível em <<http://www.r-project.org/>>. Acesso em: 04 mai. 2014.

WILLIAMS, Graham. **Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery**. England: Springer, 2011. 274 p.

WITTEN, Ian H.; FRANK, Eibe. **Data Mining: Practical Machine Learning Tools and Techniques**. 2nd. ed. United States: Morgan Kaufmann, 2005. 524 p.

APÊNDICE A - DESCRIÇÃO DOS CAMPOS DISPONÍVEIS DA BASE DE DADOS DE INCIDENTES

(continua)

Atributo	Descrição	Tipo de variável
Tenant Name	Unidade subsidiária da companhia	Nominal
Ticket ID	Código do incidente	Alfanumérico, único
Ticket Categorization	Classificação inicial do incidente (também indica o grupo responsável)	Nominal
Affected Service	Serviço de TI afetado pelo incidente	Nominal
Close Category	Classificação final do incidente	Nominal, categórico
Call Total Transfers	Quantidade de vezes que o incidente foi transferido de uma equipe para outra	Numérico, discreto
Ticket Type	Tipo de chamado (incidente/problema/mudança)	Nominal, categórico
Priority	Prioridade do incidente baseada na matriz de impacto e urgência (Pg. 141 – Magalhães, 2007)	Numérico, discreto
Urgency	Urgência: Alta, Média, Baixa, Programável. (Pg. 141 – Magalhães, 2007)	Numérico, discreto
Ticket Status	Estado atual do incidente - <i>Acknowledge, Closed, Fix in progress, Open, Resolved, Suspended, Work in progress</i>	Nominal, categórico
Resolution Code	Classificação de resolução do serviço executado	Nominal, categórico
Closure Code	Classificação de encerramento do serviço executado	Nominal, categórico
OLA Status	OLA: Operational Level Agreement - Acordo de Nível Operacional - tempo de resolução em horas acordado em contrato com o grupo de resolução. Este campo indica se o incidente foi atendido dentro ou fora do prazo acordado.	Booleano
FTR	Indica se é a primeira vez que o incidente é registrado (não confiável)	Booleano
Summary	Resumo de abertura do incidente	Texto
Description	Detalhamento de contato ou informações que possam apoiar os grupos de resolução a resolver o incidente.	Texto
Location	Unidade ou localização do usuário afetado.	Texto
Country	País do usuário afetado	Texto
User Name	Nome do usuário afetado (Ex. JORGE SILVA - Primeiro e último nome apenas)	Texto
Preferred Language	Idioma de preferência do usuário	Nominal, categórico
VIP	(YES/NO) Indica se o usuário tem prioridade de atendimento (normalmente executivos, diretores, gerentes, etc.)	Booleano

(continuação)

Atributo	Descrição	Tipo de variável
Creator Name	Nome do analista que registrou o incidente	Nominal, categórico
Assignee Group	Nome do grupo especialista do agente responsável pelo incidente	Nominal, categórico
Last Updated By	Nome do último agente a modificar informações sobre o incidente	Nominal, categórico
Last Update Text	Dados que foram modificados pelo último agente	Texto
Resolving Agent First Name	Primeiro nome do agente que marcou o incidente como resolvido	Nominal, categórico
Resolving Agent Last Name	Último nome do agente que marcou o incidente como resolvido	Nominal, categórico
Resolving Agent Middle Name	Nome do meio do agente que marcou o incidente como resolvido	Nominal, categórico
Resolving Group	Nome do grupo especialista que marcou o incidente como resolvido	Nominal, categórico
Resolving Text	Breve resumo da solução dada para o usuário afetado	Texto
Last Modified Date	Data e hora da última modificação do incidente	Data
Create Date	Data e hora do registro (criação) do incidente	Data
Assigned Date	Data e hora em que o incidente foi encaminhado a um grupo especialista	Data
Acknowledge Date	Data e hora em que um agente foi alocado para o incidente	Data
Suspended Date	Data e hora em que o incidente foi marcado como suspenso (aguardando informações/recursos/aprovações)	Data
Resolved Date	Data e hora em que o incidente foi marcado como resolvido	Data
Closed Date	Data e hora do encerramento do incidente (Servicedesk N1)	Data
Response Time (mins)	Tempo em minutos entre a Criação e Encaminhamento ao primeiro grupo especialista	Numérico, contínuo
Total Suspended Time (hrs)	Total em horas em que o incidente ficou no estado de suspensão (suspended)	Numérico, contínuo
Total Resolved Time (hrs)	Diferença em horas entre “Resolved date” e “Created date”	Numérico, contínuo
Resolved Time for RG (hrs)	Diferença em horas “Resolve Notification” – “Transfer Notification”/”Transfer”	Numérico, contínuo
Suspend Time by RG (hrs)	Tempo em horas em que o incidente ficou em espera pelo grupo especialista (RG)	Numérico, contínuo

(conclusão)

Atributo	Descrição	Tipo de variável
MTTR (hrs)	Mean Time To Repair (tempo médio para o reparo do sistema), ou seja, o tempo que a equipe de TI demanda para reparar e disponibilizar o serviço de TI afetado para o seu estado de funcionamento normal	Numérico, contínuo
Parent Ticket ID	Código do incidente de origem, se existir	Alfanumérico, único
Parent Ticket Create Date	Data de criação do incidente pai	Data
Related Problem Ticket	Código do problema que originou o incidente	Alfanumérico, único
Related Problem Ticket Create Date	Data e hora do registro do código do problema	Alfanumérico, único
Ticket Preferred Language	Idioma de preferência do incidente	Nominal, categórico
Ticket Record Method	Meio de origem do incidente (telefone/email/evento de sistema)	Nominal, categórico
Attachment Doc	Data e hora em que um anexo foi adicionado ao incidente	Data
OLA Business Schedule	Horário de atendimento do grupo especialista (incluindo feriados). Exemplo: “Mon - Fri { 8:00 AM - 5:00 PM } 01/01/12...”	Nominal, categórico
OLA Business Working Hours	Código do horário de atendimento do grupo especialista. Exemplo: “Workshift 190”	Nominal, categórico
Resolving Classification	Classificação de resolução do incidente (semelhante ao close category e ticket categorisation)	Nominal, categórico
Reopen?	(YES/NO) Indica se o incidente foi reaberto (resolvido e não encerrado -> reaberto por pedido do usuário – Non user acceptance)	Booleano
Priority Change?	(YES/NO) Indica se o incidente passou por escalação de prioridade	Booleano

APÊNDICE B – REGRAS DE ASSOCIAÇÃO GERADAS PELO ALGORITMO APRIORI

lhs	rhs	support	confidence	lift
1 {Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=FTF, RG.SD=YES}	=> {Remotely=YES}	0.13036424	0.9986035	1.0889489
2 {Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, FTF.FCR=FTF, RG.SD=YES, Remotely=YES}	=> {Priority_Cat=BAIXA}	0.13036424	0.9341607	1.2220369
3 {TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=FTF, RG.SD=YES, Remotely=YES}	=> {Ticket.Record.Method=Telephone}	0.13036424	0.9864809	1.4891825
4 {Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, Priority_Cat=BAIXA, FTF.FCR=FTF, RG.SD=YES, Remotely=YES}	=> {CallTT_Cat=BAIXA}	0.13036424	0.9997204	1.7115153
5 {Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=FTF, Remotely=YES}	=> {RG.SD=YES}	0.13036424	0.9987430	3.8394456
6 {Ticket.Record.Method=Telephone, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=FTF, RG.SD=YES, Remotely=YES}	=> {TTTR_Cat=BAIXO}	0.13036424	0.9076025	3.7114676
7 {Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, RG.SD=YES, Remotely=YES}	=> {FTF.FCR=FTF}	0.13036424	0.8689999	4.8125310
8 {Ticket.Record.Method=Telephone, TTTR_Cat=MEDIO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO}	=> {Remotely=YES}	0.10646443	0.9635374	1.0507103
9 {Ticket.Record.Method=Telephone, TTTR_Cat=MEDIO, CallTT_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO,				

Remotely=YES}	=> {Priority_Cat=BAIXA}	0.10646443	0.7170043	0.9379604
10 {Ticket.Record.Method=Telephone, TTTR_Cat=MEDIO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.10646443	0.9859868	1.3132286
11 {Ticket.Record.Method=Telephone, TTTR_Cat=MEDIO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=NO, Remotely=YES}	=> {RG.SD=NO}	0.10646443	1.0000000	1.3515831
12 {TTTR_Cat=MEDIO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO, Remotely=YES}	=> {Ticket.Record.Method=Telephone}	0.10646443	0.9587917	1.4473830
13 {Ticket.Record.Method=Telephone, TTTR_Cat=MEDIO, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO, Remotely=YES}	=> {CallTT_Cat=BAIXA}	0.10646443	0.8575624	1.4681417
14 {Ticket.Record.Method=Telephone, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO, Remotely=YES}	=> {TTTR_Cat=MEDIO}	0.10646443	0.4913343	1.4512764
15 {Ticket.Record.Method=E-mail, TTTR_Cat=MEDIO, CallTT_Cat=MEDIA, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO}	=> {Remotely=YES}	0.06360521	0.9590434	1.0458098
16 {Ticket.Record.Method=E-mail, TTTR_Cat=MEDIO, CallTT_Cat=MEDIA, FTF.FCR=NO, RG.SD=NO, Remotely=YES}	=> {Priority_Cat=BAIXA}	0.06360521	0.8370921	1.0950551
17 {Ticket.Record.Method=E-mail, TTTR_Cat=MEDIO, CallTT_Cat=MEDIA, Priority_Cat=BAIXA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.06360521	0.9923208	1.3216648
18 {Ticket.Record.Method=E-mail, TTTR_Cat=MEDIO, CallTT_Cat=MEDIA, Priority_Cat=BAIXA, FTF.FCR=NO, Remotely=YES}	=> {RG.SD=NO}	0.06360521	0.9971420	1.3477203
19 {Ticket.Record.Method=E-mail, CallTT_Cat=MEDIA, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO,				

Remotely=YES} => {TTTR_Cat=MEDIO} 0.06360521 0.4103258 1.2119977
 20 {Ticket.Record.Method=E-mail,
 TTTR_Cat=MEDIO,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {CallTT_Cat=MEDIA} 0.06360521 0.9145478 2.9909144
 21 {TTTR_Cat=MEDIO,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {Ticket.Record.Method=E-mail} 0.06360521 0.7916950 2.5920759
 22 {Ticket.Record.Method=E-mail,
 TTTR_Cat=MUITO_ALTO,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO} => {Remotely=YES} 0.05554745 0.9471557 1.0328466
 23 {Ticket.Record.Method=E-mail,
 TTTR_Cat=MUITO_ALTO,
 CallTT_Cat=MEDIA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {Priority_Cat=BAIXA} 0.05554745 0.8870451 1.1604019
 24 {Ticket.Record.Method=E-mail,
 TTTR_Cat=MUITO_ALTO,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 RG.SD=NO,
 Remotely=YES} => {FTF.FCR=NO} 0.05554745 0.9918620 1.3210537
 25 {Ticket.Record.Method=E-mail,
 TTTR_Cat=MUITO_ALTO,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 Remotely=YES} => {RG.SD=NO} 0.05554745 0.9759769 1.3191139
 26 {Ticket.Record.Method=E-mail,
 TTTR_Cat=MUITO_ALTO,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {CallTT_Cat=MEDIA} 0.05554745 0.6470588 2.1161250
 27 {TTTR_Cat=MUITO_ALTO,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {Ticket.Record.Method=E-mail} 0.05554745 0.7043458 2.3060872
 28 {Ticket.Record.Method=E-mail,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {TTTR_Cat=MUITO_ALTO} 0.05554745 0.3583441 1.2389920
 29 {Ticket.Record.Method=Telephone,
 TTTR_Cat=BAIXO,
 CallTT_Cat=BAIXA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,

RG.SD=NO}	=> {Remotely=YES}	0.04397127 0.9690639 1.0567368
30 {Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO, Remotely=YES}	=> {Priority_Cat=BAIXA}	0.04397127 0.7071240 0.9250353
31 {Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.04397127 0.9962825 1.3269414
32 {Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=NO, Remotely=YES}	=> {RG.SD=NO}	0.04397127 1.0000000 1.3515831
33 {TTTR_Cat=BAIXO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO, Remotely=YES}	=> {Ticket.Record.Method=Telephone}	0.04397127 0.9848918 1.4867836
34 {Ticket.Record.Method=Telephone, TTTR_Cat=BAIXO, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO, Remotely=YES}	=> {CallTT_Cat=BAIXA}	0.04397127 0.9717969 1.6637105
35 {Ticket.Record.Method=Telephone, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO, Remotely=YES}	=> {TTTR_Cat=BAIXO}	0.04397127 0.2029278 0.8298347
36 {Ticket.Record.Method=Telephone, TTTR_Cat=MUITO_ALTO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO}	=> {Remotely=YES}	0.04070806 0.9292551 1.0133264
37 {Ticket.Record.Method=Telephone, TTTR_Cat=MUITO_ALTO, CallTT_Cat=BAIXA, FTF.FCR=NO, RG.SD=NO, Remotely=YES}	=> {Priority_Cat=BAIXA}	0.04070806 0.7884887 1.0314738
38 {Ticket.Record.Method=Telephone, TTTR_Cat=MUITO_ALTO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, RG.SD=NO, Remotely=YES}	=> {FTF.FCR=NO}	0.04070806 0.9850022 1.3119172
39 {Ticket.Record.Method=Telephone, TTTR_Cat=MUITO_ALTO, CallTT_Cat=BAIXA, Priority_Cat=BAIXA, FTF.FCR=NO,		

Remotely=YES} => {RG.SD=NO} 0.04070806 1.0000000 1.3515831
 40 {TTTR_Cat=MUITO_ALTO,
 CallTT_Cat=BAIXA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {Ticket.Record.Method=Telephone} 0.04070806 0.9319699 1.4068932
 41 {Ticket.Record.Method=Telephone,
 TTTR_Cat=MUITO_ALTO,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {CallTT_Cat=BAIXA} 0.04070806 0.4947928 0.8470823
 42 {Ticket.Record.Method=Telephone,
 CallTT_Cat=BAIXA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {TTTR_Cat=MUITO_ALTO} 0.04070806 0.1878681 0.6495629
 43 {Ticket.Record.Method=Telephone,
 TTTR_Cat=MEDIO,
 CallTT_Cat=BAIXA,
 Priority_Cat=MEDIA,
 FTF.FCR=NO,
 RG.SD=NO} => {Remotely=YES} 0.04017939 0.9283909 1.0123840
 44 {Ticket.Record.Method=Telephone,
 TTTR_Cat=MEDIO,
 CallTT_Cat=BAIXA,
 Priority_Cat=MEDIA,
 RG.SD=NO,
 Remotely=YES} => {FTF.FCR=NO} 0.04017939 0.9782512 1.3029256
 45 {Ticket.Record.Method=Telephone,
 TTTR_Cat=MEDIO,
 CallTT_Cat=BAIXA,
 Priority_Cat=MEDIA,
 FTF.FCR=NO,
 Remotely=YES} => {RG.SD=NO} 0.04017939 1.0000000 1.3515831
 46 {TTTR_Cat=MEDIO,
 CallTT_Cat=BAIXA,
 Priority_Cat=MEDIA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {Ticket.Record.Method=Telephone} 0.04017939 0.9726390 1.4682869
 47 {Ticket.Record.Method=Telephone,
 TTTR_Cat=MEDIO,
 Priority_Cat=MEDIA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {CallTT_Cat=BAIXA} 0.04017939 0.8000000 1.3695952
 48 {Ticket.Record.Method=Telephone,
 CallTT_Cat=BAIXA,
 Priority_Cat=MEDIA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {TTTR_Cat=MEDIO} 0.04017939 0.5200566 1.5361147
 49 {Ticket.Record.Method=Telephone,
 TTTR_Cat=MEDIO,
 CallTT_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,

Remotely=YES} => {Priority_Cat=MEDIA} 0.04017939 0.2705955 1.1968427
 50 {Ticket.Record.Method=E-mail,
 TTTR_Cat=ALTO,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO} => {Remotely=YES} 0.03527546 0.9480647 1.0338377
 51 {Ticket.Record.Method=E-mail,
 TTTR_Cat=ALTO,
 CallTT_Cat=MEDIA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {Priority_Cat=BAIXA} 0.03527546 0.8319003 1.0882633
 52 {Ticket.Record.Method=E-mail,
 TTTR_Cat=ALTO,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 RG.SD=NO,
 Remotely=YES} => {FTF.FCR=NO} 0.03527546 0.9933265 1.3230043
 53 {Ticket.Record.Method=E-mail,
 TTTR_Cat=ALTO,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 Remotely=YES} => {RG.SD=NO} 0.03527546 0.9892638 1.3370722
 54 {Ticket.Record.Method=E-mail,
 TTTR_Cat=ALTO,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {CallTT_Cat=MEDIA} 0.03527546 0.8405734 2.7489903
 55 {TTTR_Cat=ALTO,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {Ticket.Record.Method=E-mail} 0.03527546 0.7976092 2.6114395
 56 {Ticket.Record.Method=E-mail,
 CallTT_Cat=MEDIA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {TTTR_Cat=ALTO} 0.03527546 0.2275667 1.7822596
 57 {Ticket.Record.Method=Telephone,
 TTTR_Cat=ALTO,
 CallTT_Cat=BAIXA,
 Priority_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO} => {Remotely=YES} 0.02554053 0.9402685 1.0253362
 58 {Ticket.Record.Method=Telephone,
 TTTR_Cat=ALTO,
 CallTT_Cat=BAIXA,
 FTF.FCR=NO,
 RG.SD=NO,
 Remotely=YES} => {Priority_Cat=BAIXA} 0.02554053 0.7512064 0.9827024
 59 {Ticket.Record.Method=Telephone,
 TTTR_Cat=ALTO,
 CallTT_Cat=BAIXA,
 Priority_Cat=BAIXA,
 RG.SD=NO,

Remotely=YES} => {FTF.FCR=NO} 0.02554053 0.9831579 1.3094608

60 {Ticket.Record.Method=Telephone,
TTTR_Cat=ALTO,
CallTT_Cat=BAIXA,
Priority_Cat=BAIXA,
FTF.FCR=NO,
Remotely=YES} => {RG.SD=NO} 0.02554053 1.0000000 1.3515831

61 {TTTR_Cat=ALTO,
CallTT_Cat=BAIXA,
Priority_Cat=BAIXA,
FTF.FCR=NO,
RG.SD=NO,
Remotely=YES} => {Ticket.Record.Method=Telephone} 0.02554053 0.9192913 1.3877537

62 {Ticket.Record.Method=Telephone,
TTTR_Cat=ALTO,
Priority_Cat=BAIXA,
FTF.FCR=NO,
RG.SD=NO,
Remotely=YES} => {CallTT_Cat=BAIXA} 0.02554053 0.7040201 1.2052782

63 {Ticket.Record.Method=Telephone,
CallTT_Cat=BAIXA,
Priority_Cat=BAIXA,
FTF.FCR=NO,
RG.SD=NO,
Remotely=YES} => {TTTR_Cat=ALTO} 0.02554053 0.1178698 0.9231336

64 {Ticket.Record.Method=Telephone,
TTTR_Cat=MUITO_ALTO,
CallTT_Cat=MEDIA,
Priority_Cat=BAIXA,
FTF.FCR=NO,
RG.SD=NO} => {Remotely=YES} 0.02284245 0.9213235 1.0046773

65 {Ticket.Record.Method=Telephone,
TTTR_Cat=MUITO_ALTO,
CallTT_Cat=MEDIA,
FTF.FCR=NO,
RG.SD=NO,
Remotely=YES} => {Priority_Cat=BAIXA} 0.02284245 0.6412487 0.8388595

66 {Ticket.Record.Method=Telephone,
TTTR_Cat=MUITO_ALTO,
CallTT_Cat=MEDIA,
Priority_Cat=BAIXA,
RG.SD=NO,
Remotely=YES} => {FTF.FCR=NO} 0.02284245 0.9638462 1.2837396

67 {Ticket.Record.Method=Telephone,
TTTR_Cat=MUITO_ALTO,
CallTT_Cat=MEDIA,
Priority_Cat=BAIXA,
FTF.FCR=NO,
Remotely=YES} => {RG.SD=NO} 0.02284245 0.9099492 1.2298719

68 {TTTR_Cat=MUITO_ALTO,
CallTT_Cat=MEDIA,
Priority_Cat=BAIXA,
FTF.FCR=NO,
RG.SD=NO,
Remotely=YES} => {Ticket.Record.Method=Telephone} 0.02284245 0.2896440 0.4372439

69 {Ticket.Record.Method=Telephone,
TTTR_Cat=MUITO_ALTO,
Priority_Cat=BAIXA,
FTF.FCR=NO,
RG.SD=NO,

Remotely=YES} => {CallTT_Cat=MEDIA} 0.02284245 0.2776424 0.9079947
70 {Ticket.Record.Method=Telephone,
CallTT_Cat=MEDIA,
Priority_Cat=BAIXA,
FTF.FCR=NO,
RG.SD=NO,
Remotely=YES} => {TTTR_Cat=MUITO_ALTO} 0.02284245 0.4621911 1.5980479

Rattle timestamp: 2014-10-26 16:47:39 Cleber