UNIVERSIDADE FEEVALE

THOMÁS AUGUSTO FINCK

RESTRIÇÃO DE ACESSO AO CONTEÚDO BASEADO EM FOLKSONOMIA E MINERAÇÃO DE DADOS

Novo Hamburgo

2015

THOMÁS AUGUSTO FINCK

RESTRIÇÃO DE ACESSO AO CONTEÚDO BASEADO EM FOLKSONOMIA E MINERAÇÃO DE DADOS

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
2015

AGRADECIMENTOS

Aos meus pais, Carlos e Tânia, por me darem a vida; meus irmãos por contribuírem na minha formação como ser humano; ao meu orientador pela atenção, ensinamentos, apoio e sua dedicação; dentre todos esses, especialmente à minha esposa Luana Prado Finck, por todo carinho, amor e apoio recebidos durante esses anos.

RESUMO

O uso dos aspectos da Folksonomia para a restrição de conteúdos impróprios da Internet, tanto em contextos corporativos quanto pessoais, pode melhorar o controle dos acessos indevidos dos usuários. Através da etiquetagem da informação, o sistema de gerenciamento tende a melhorar a forma de categorização e restrição da informação. É sabido que a Internet é um ambiente de fácil acesso a todas as espécies de conteúdos, proporcionando aos usuários o acesso simples e prático à informação. Contudo, gestores têm tomado diversas ações com objetivo de minimizar a perda da produtividade, riscos com a segurança do ambiente, o mau uso dos recursos computacionais e, por fim, o acesso aos conteúdos impróprios da Internet. Percebe-se a necessidade de novas técnicas para uma melhor eficácia no gerenciamento dos acessos à Internet em empresas, instituições e em contextos caseiros, servindo como instrumento de apoio ao gerenciamento de acessos *Web*. Este trabalho estuda e aplica os conceitos da Folksonomia para a criação de um classificador, baseado em árvores de decisão, que tem como objetivo auxiliar na gestão de acesso aos conteúdos da Internet.

Palavras-chave: Folksonomia. Restrição de Conteúdos. Controle de Acesso à Internet. Etiquetas. Árvores de Decisão. Mineração de Dados.

ABSTRACT

The use of the aspects of Folksonomy for the restriction of inappropriate Internet content in both corporate and personal contexts, can improve the control of unauthorized access of users. Through information labeling, the management system tends to improve the form of categorization and restriction of information. It is known that the Internet is an easy access to all kinds of content environment, providing users with simple and convenient access to information. However, managers have taken several actions in order to minimize the loss of productivity, risks to environmental security, misuse of computing resources and, finally, access to inappropriate Internet content. We see the need for new techniques to better efficiency in the management of Internet access in companies and institutions and in home settings, serving as support to the management *Web* access tool. This paper studies and applies the concepts of Folksonomy to create a classifier based on decision trees, which aims to assist in the management of access to Internet content.

Keywords: Folksonomy. Content Restriction. Internet Access Control. Labels. Decision Trees. Data Mining.

LISTA DE FIGURAS

Figura 1 - Esquema de funcionamento da Folksonomia. Fonte: Nascimento (2008). . 13
Figura 2 – Módulos do fluxo para geração do modelo. Fonte: elaborado pelo autor... 14
Figura 3 – Página inicial site Delicious. Fonte: Delicious (2015)..... 17
Figura 4 – Site Delicious (*Tags* relacionadas). Fonte: Delicious (2015)..... 30
Figura 5 – Processo de classificação. Fonte: CLARK (1989). 32
Figura 6 – Etapas KDD. Fonte: Fayyad et al. (1996). 35
Figura 7 – Imagem com a tabela de exemplo para a tomada de decisão (Classificação). Fonte: BARAZETTI 2001..... 39
Figura 8 – Exemplo de um algoritmo utilizando árvore de decisão. Fonte: BARAZETTI 2001. 40
Figura 9 - Entrada de dados para a descoberta de regras de associação. Fonte: RODRIGUES (2000). 43
Figura 10 - Clientes divididos em segmentos que possuem semelhanças. Fonte: RODRIGUES (2000). 44
Figura 11 – Tarefas de *data mining* de forma resumida. Fonte: Fayyad (1996)..... 46
Figura 12 – Árvore de decisão (Simples). Fonte: desenvolvido pelo autor..... 48
Figura 13 – Árvore de decisão. Fonte: desenvolvido pelo autor..... 50
Figura 14 - Exemplo de árvore de decisão com resposta binária. Fonte: Gama (1999). 53
Figura 15 – Diagrama do processo de indução de um classificador. Fonte: CHAPELLE (2006). 55
Figura 16 – Algoritmo genérico para construção de árvores de decisão. Fonte: o autor. 57
Figura 17 - Estrutura necessária do sistema desenvolvido. Fonte: SILVA (2009). 70
Figura 18 – Arquitetura em camadas do sistema. Fonte: SILVA (2009). 70
Figura 19 – Fluxo da geração do modelo de classificador. Fonte: desenvolvido pelo autor. 72
Figura 20 – Tabela “FolkUrl – Estrutura e atributos. Fonte: desenvolvido pelo autor. 73
Figura 21 – Tabela FolkUrl com atributo “acesso” preenchido. Fonte: desenvolvido pelo autor..... 74
Figura 22 – Tabela FolkUrl com *tags* do Delicious. Fonte: desenvolvido pelo autor.. 74

Figura 23 - *Script* de conexão “load_tags.php”. Fonte: desenvolvido pelo autor. 75
Figura 24 - Comandos SQL para analisar as *tags* associadas. Fonte: desenvolvido pelo autor. 76
Figura 25 – Número de *tags* cadastradas por atributos *tag1, tag2, ..., tag10*. Fonte: desenvolvido pelo autor. 77
Figura 26 - Número de registros de solicitações de acesso. Fonte: desenvolvido pelo autor. 77
Figura 27 - Percentual de acessos bloqueados e permitidos da tabela “FolkUrl”. Fonte: desenvolvido pelo autor. 78
Figura 28 - Limpeza relacionada aos símbolos “;” e transformações de conteúdos em idiomas diferentes. Fonte: desenvolvido pelo autor. 78
Figura 29 - Limpeza de aspas. Fonte: desenvolvido pelo autor. 79
Figura 30 - Arquivo “FolkUrl_Limpo.csv” gerado após a limpeza do conteúdo das *tags*. Fonte: desenvolvido pelo autor. 79
Figura 31 - Remoção de atributos dentro da ferramenta WEKA. Fonte: desenvolvido pelo autor. 80
Figura 32 - Treinamento e teste preliminar aplicando o algoritmo J48 sobre os dados. Fonte: desenvolvido pelo autor. 81
Figura 33 - Matriz de confusão gerada a partir da aplicação do algoritmo J48. Fonte: desenvolvido pelo autor. 81
Figura 34 – Consulta SQL para visualizar a variedade de termos nos atributos *tag1, tag2, ..., tag10*. Fonte: desenvolvido pelo autor. 82
Figura 35 – Consulta SQL para armazenamento das *tags* distintas. Fonte: desenvolvido pelo autor. 82
Figura 36 - Número de repetições de valores das *tags* na tabela “folkurltodastags”. Fonte: desenvolvido pelo autor. 83
Figura 37 – Comandos SQL para criação da tabela “folkurltodastags_50”. Fonte: desenvolvido pelo autor. 83
Figura 38 – *Script* PHP “transpose_tags.php”. Fonte: desenvolvido pelo autor. 84
Figura 39 – Arquivo “tags.csv”. Fonte: desenvolvido pelo autor. 85
Figura 40 – Treinamento WEKA J48 com arquivo “tags.csv”. Fonte: desenvolvido pelo autor. 86
Figura 41 - Resultado matriz de confusão algoritmo J48. Fonte: elaborado pelo autor. 86

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
BI	*Business Intelligence*
CC	*Composite Capability*
DM	*Data Mining*
DMX	*Data Mining Extensions*
HTML	*HyperText Markup Language*
HTTP	*Hypertext Transfer Protocol*
ICAP	*Internet Content Adaptation Protocol*
ISPs	*Internet Service Providers*
KDD	*Knowledge Discovery in Databases*
OWL	*Web Ontology Language*
PHP	*Hypertext Preprocessor*
PP	*Preference Profile*
RDF	*Resouce Description Framework*
RFF	*Resouce File Framework*
RNAs	Redes Neurais Artificiais
RI	Recuperação de Informação
SI	Sistema de Informação
SLA	*Service Level Agreement*
SQL	*Structured Query Language*
SMO	Sequência Mínima de Otimização
URL	*Uniform Resource Locator*
XML	*Extensible Markup Language*
W3C	*World Wide Web Consortium*
WEB	*World Wide Web*
WEKA	*Waikato Environment for Knowledge Analysis*

SUMÁRIO

1 INTRODUÇÃO 11

2 FOLKSONOMIA 16

2.1 INFLUÊNCIA DA *WEB* 2.0 NA FOLKSONOMIA 19

2.2 ASPECTOS COLABORATIVOS E SOCIAIS DA FOLKSONOMIA 22

2.3 APLICAÇÕES DA FOLKSONOMIA 24

2.3.1 SISTEMAS BASEADOS EM FOLKSONOMIA 24

2.3.2 CARACTERÍSTICAS 25

2.3.3 VANTAGENS E DESVANTAGENS DA FOLKSONOMIA 25

2.4 DELICIOUS 29

2.4.1 API DELICIOUS 30

3 CLASSIFICADORES 31

3.1 DESCOBERTA DE CONHECIMENTO (KDD – KNOWLEDGE DISCOVERY DATABASE) 34

3.2 SELEÇÃO DOS DADOS 35

3.3 PRÉ-PROCESSAMENTO 36

3.4 MINERAÇÃO DE DADOS (DATA MINING) 37

3.5 TAREFAS DE *DATA MINING* 38

3.5.1 CLASSIFICAÇÃO 39

3.5.1.1 ÁRVORES DE DECISÃO 40

3.5.1.2 REDES NEURAIS ARTIFICIAIS 40

3.5.2 ESTIMATIVA (REGRESSÃO) 42

3.5.3 ASSOCIAÇÃO 43

3.5.4 SEGMENTAÇÃO (CLUSTERIZAÇÃO) 44

3.5.5 SUMARIZAÇÃO 44

3.6 PÓS-PROCESSAMENTO 46

4 ÁRVORES DE DECISÃO 48

4.1 CLASSIFICADORES BASEADOS EM ÁRVORES DE DECISÃO 49

4.2 PRINCIPAIS CONCEITOS VINCULADOS À INDUÇÃO DE ÁRVORES DE DECISÃO 50

4.3 TOP-DOWN INDUCTION OF DECISION TREE (TDIDT) 51

4.4 ESCOLHA DOS ATRIBUTOS PREDITIVOS PARA OS NÓS DA ÁRVORE 51
4.5 CONSTRUÇÃO DE UMA ÁRVORE DE DECISÃO 52
4.6 ALGORITMO PARA A INDUÇÃO DA ÁRVORE..... 56
4.7 WEKA (Waikato Environment for Knowledge Analysis) 58
4.7.1 TREINAMENTO E EXECUÇÃO DO ALGORITMO DE MINERAÇÃO DE DADOS 60

5 TRABALHOS RELACIONADOS..... 62
5.1 O USO DE FILTROS DE CONTROLE NA INTERNET: O CASO DO SISTEMA EDUCACIONAL 62
5.2 PROTÓTIPO SISBLOQUE: TÉCNICA DE FILTRAGEM E BLOQUEIO DE CONTEÚDO WEB 64
5.3 ESTUDO SOBRE A DOAÇÃO DA FOLKSONOMIA EM SISTEMA DE INFORMAÇÃO: UMA PROPOSTA DE HIBRIDISMO 67
5.4 GERENCIAMENTO DO VOCABULÁRIO DO USUÁRIO EM SISTEMAS BASEADOS EM *TAGGING* 69

6 ESTUDO DE CASO 72
6.1 PRÉ-PROCESSAMENTO 73
6.2 TREINAMENTO E TESTE PRELIMINAR..... 79
6.3 NOVO PRÉ-PROCESSAMENTO 81
6.4 NOVO TREINAMENTO E TESTE 85

7 CONCLUSÃO..... 88

8 REFERÊNCIAS BIBLIOGRÁFICAS..... 90

APÊNDICE A 98

1 INTRODUÇÃO

A Internet consolidou-se como um dos mais populares ambientes de divulgação de informação, utilizando-se de diversos meios, tais como: documentos digitais, páginas *Web*, vídeos entre outros. Assim, ela pode ofertar diferentes conteúdos aos usuários, sendo alguns conteúdos de caráter profissional, educacional ou então de entretenimento.

O uso indevido, inadequado, do conteúdo da Internet, é um problema sério em grandes corporações empresariais que a cada ano aplicam ainda mais esforços, financeiros e técnicos, na expectativa de mitigar riscos com segurança, perda de produtividade, utilização de banda e ainda acesso a conteúdo indevidos.

Dentre as diversas atitudes tomadas por gestores para evitar o uso inadequado da Internet no ambiente computacional da empresa, cita-se o monitoramento e bloqueio de conteúdos como prioritários no gerenciamento.

Sendo assim, empresas da área de tecnologia têm desenvolvido novos métodos de controle de acesso ao conteúdo de Internet. Podem-se citar as seguintes ferramentas de controle de acesso:

- A ferramenta da empresa *Websense* (WEBSENSE, 2015), conhecida como ***Websense Web Filter*** (WEBSENSE WEB FILTER, 2015), possui uma grande quantidade de *urls*, previamente categorizadas em grupos, que podem ser bloqueados ou então liberados, conforme a política de segurança adotada na empresa que utiliza o *Websense Web Filter*.
- Outra ferramenta que possui o mesmo propósito é a da empresa *M86 (*M86, 2010*)*, chamada de ***M86 Web Filtering and Reporting Suite*** (M86 *WEB* FILTERING AND REPORTING SUITE, 2015).
- Já a ferramenta da empresa Barracuda (BARRACUDA, 2015), conhecida como ***Barracuda Web Filter*** (*BARRACUDA WEB FILTER*, 2015) é aplicada em empresas de menor porte devido ao baixo custo em relação a outras ferramentas de filtro de controle de acesso *Web* supracitadas.

As ferramentas anteriormente citadas possuem, do ponto de vista computacional, uma arquitetura similar, com o objetivo de gerenciar o acesso ao conteúdo. Essa arquitetura pode ser entendida como uma consulta aos dados previamente catalogados. O resultado da consulta pode ser positivo ou negativo, tendo vista a possibilidade de um usuário ter acesso a

uma determinada informação. Dessa forma, toda a assertividade dessa abordagem fica diretamente ligada à quantidade e a qualidade da catalogação executada pelas empresas usuárias da ferramenta.

Ainda sobre as ferramentas de controle de acesso, pode-se disser que estas são aplicações correlatas ao objetivo deste trabalho. Neste trabalho é utilizada uma abordagem de catalogação colaborativa dos dados. Desta forma, é possível favorecer a quantidade e qualidade dos dados, permitindo a categorização dos dados por várias pessoas.

Foi realizada uma pesquisa sobre trabalhos que objetivam o controle de acesso a informação, bem como a utilização de catalogação colaborativa. Existe uma técnica que formaliza a catalogação e a categorização de conteúdos de forma colaborativa, assim, diversos usuários podem criar diferentes ou então semelhantes marcas (etiquetas), conforme seu vocabulário, que classifiquem um mesmo conteúdo (O'REILLY,2005) (NASCIMENTO, 2008).

No ano de 2004, Thomas Vander Wal, cunhou o termo "Folksonomia", que representa uma nova forma de categorização de informação (VANDER WAL, 2007), baseada no ato de rotular e categorizar documentos através de etiquetas (*tags*). Essas etiquetas, por sua vez, são expressões que descrevem de forma sucinta o conteúdo ao qual está relacionada. Desde então, a Folksonomia vem sendo aplicada em diversas ferramentas para categorizar conteúdos na Internet, tais como Youtube (YOUTUBE, 2015) e Delicious (DELICIOUS, 2015).

O objetivo primário da Folksonomia (MATHES, 2004) é promover uma rápida indexação de conteúdo e facilitar as técnicas de filtragem, a fim de permitir buscas sobre os diversos tipos de informação. A Figura 1 ilustra a forma colaborativa que as pessoas podem indexar seus conteúdos (NASCIMENTO, 2008). Nela há indivíduos indexando objetos, buscando por objetos no repositório e por objetos indexados por determinado autor.

Figura 1 - Esquema de funcionamento da Folksonomia. Fonte: Nascimento (2008).

O problema apresentado, ou seja, o uso inadequado do conteúdo da Internet em grandes corporações empresariais e no âmbito doméstico, geram riscos com segurança da informação, perda de produtividade e acesso aos conteúdos indevidos, dentre outros. Diante disso foi investigada a convergência dos atuais conceitos de Folksonomia para a criação de um modelo de classificador, baseado em árvores de decisão, permitir ou restringir o acesso aos conteúdos da Internet.

A Figura 2 representa o fluxo de etapas para a geração de um modelo de classificador. Modelo este que será uma alternativa para auxiliar o controle de acesso a informação baseada em Folksonomia. O modelo baseia-se em três módulos, sendo um primeiro módulo de pré-processamento, o segundo de *data mining* e o terceiro de pós-processamento.

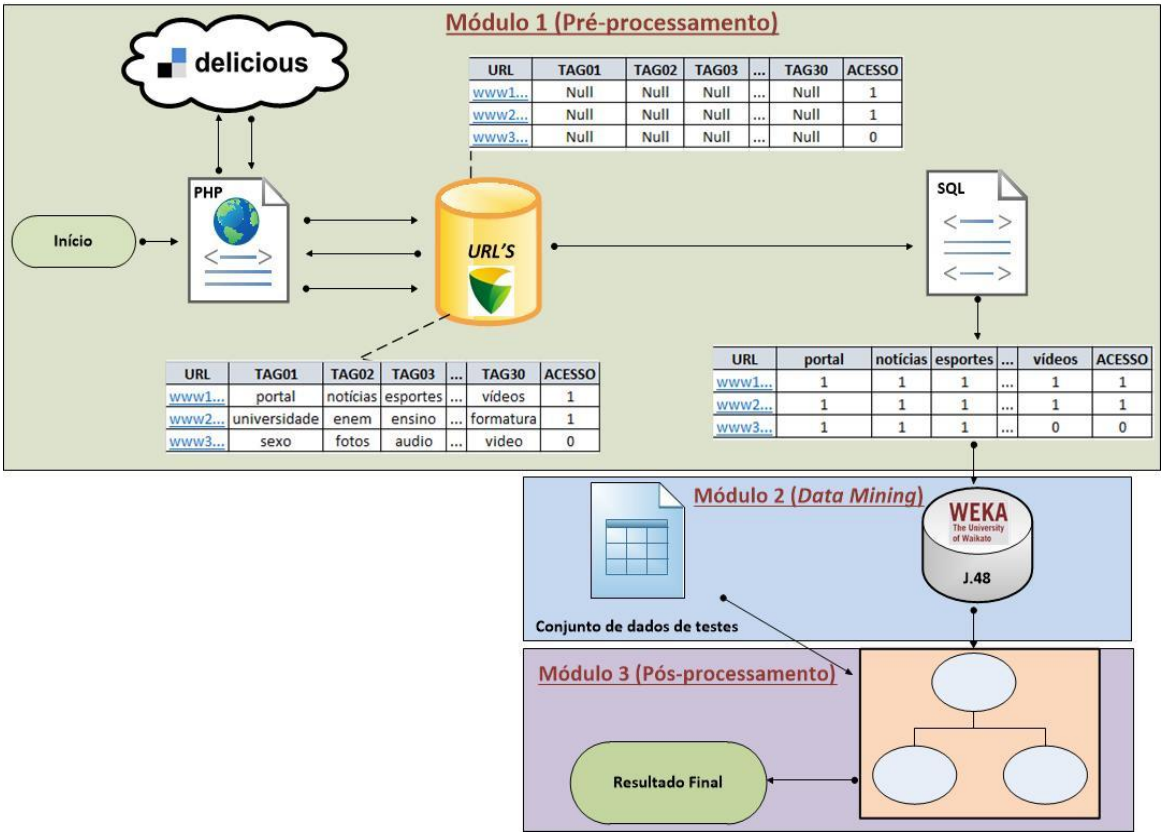


Figura 2 – Módulos do fluxo para geração do modelo. Fonte: elaborado pelo autor.

Dessa forma, **o primeiro módulo** (pré-processamento) na Figura 2, representado com fundo verde, é responsável por promover uma classificação colaborativa de *urls* que indicam sites diversos da Internet. Utilizou-se a rede social Delicious (DELICIOUS, 2015) como fonte de dados para a consulta de *urls*. A utilização do Delicious (DELICIOUS, 2015) é devida a técnica de Folksonomia que este promove e permite aos usuários da rede associar qualquer palavra (*tag*) a um determinado registro, de forma que esse possa ser recuperado, em um segundo momento, através da sua classificação.

O **segundo módulo** (*data mining*), na Figura 2, representado com fundo azul, visa gerar o conhecimento através do *data mining* utilizando um algoritmo de classificação. No **terceiro módulo** (**pós-processamento**) da Figura 2, representado com fundo lilás, será gerado o conhecimento para a indicação de acesso permitido ou negado aos conteúdos inadequados, inviabilizando que os mesmos sejam acessados. Ou seja, utilizou-se Aprendizado de Máquina (RIGO, 2010) para viabilizar contribuir com a filtragem dos conteúdos.

Este trabalho está organizado em 8 capítulos. O capítulo 1 é esta introdução. No capítulo 2 estão organizados os conceitos sobre Folksonomia. O capítulo 3 apresenta classificadores e a descoberta de conhecimento (KDD). No capítulo 4 foi explanado o classificador árvore de decisão e o WEKA. No capítulo 5, trabalhos relacionados, estão

apresentados trabalhos sobre Folksonomia e restrição de conteúdo. No capítulo 6 estão apresentadas as etapas do desenvolvimento do modelo proposto. Por fim, o capítulo 7 está a conclusão e, no capítulo 8 referências bibliográficas do trabalho.

2 FOLKSONOMIA

O termo Folksonomia foi criado em 2004 por Thomas Vander Wal, neologismo, que junta as palavras *folk* (representa povo, em inglês) com Taxonomia (ciência de classificação). Em outras palavras, a Taxonomia prevê a classificação da informação de modo hierarquizado, a partir de categorias pré-existentes. É a base de sistemas de classificação tradicionais. De acordo com PEREIRA (2006):

> Taxonomia é o estudo de classificação das coisas. É o ato de dar nomes, classificar, identificar. [...] Quando um cientista classifica um novo inseto, ele procura classificá-lo dentro de uma categoria já existente, baseado em uma lógica estabelecida, verifica qual família ele pertence e no fim encontra o nome mais adequado aquela espécie. Isso é taxonomia. (PEREIRA, 2006).

A principal característica para aliar a Folksonomia como representante oficial da classificação social da informação é a *tagging*. *Tagging* significa etiquetagem, atribuição de *tags* ou etiquetas para classificar objetos da Internet. Desta maneira, Vander Wal (2007) define que a Folksonomia é uma maneira de as pessoas etiquetarem objetos (qualquer coisa que possa estar na internet) usando o seu próprio vocabulário, a fim de facilitar a pesquisa sobre a informação etiquetada.

Já O'Reilly (2005, online) distingue a Folksonomia como uma prática coletiva de organização das informações feita pelos próprios usuários da *Web*, e não mais na Taxonomia, baseada em ordenações lineares e elaborada por um corpo específico de profissionais. Desta forma, o autor cita sistemas como o Flickr (FLICKR, 2015) e o del.icio.us (DELICIOUS, 2015) como novos formatos de representação e recuperação na *Web*, por basearem-se em Folksonomia.

A Folksonomia é o processo de indexação adotado por muitos dos sistemas de compartilhamento de arquivos na Internet, atualmente. Nestes sistemas a representação do conteúdo é feita em linguagem livre do usuário, sem qualquer tipo de intervenção direta de gestores ou mecanismos de organização especializados ou ainda indutivos. De forma concisa, a Folksonomia pode ser entendida como "(...) uma forma de os próprios usuários disponibilizarem os rótulos adotados intuitivamente no compartilhamento, organização e na recuperação futura da informação" (MOURA, 2009).

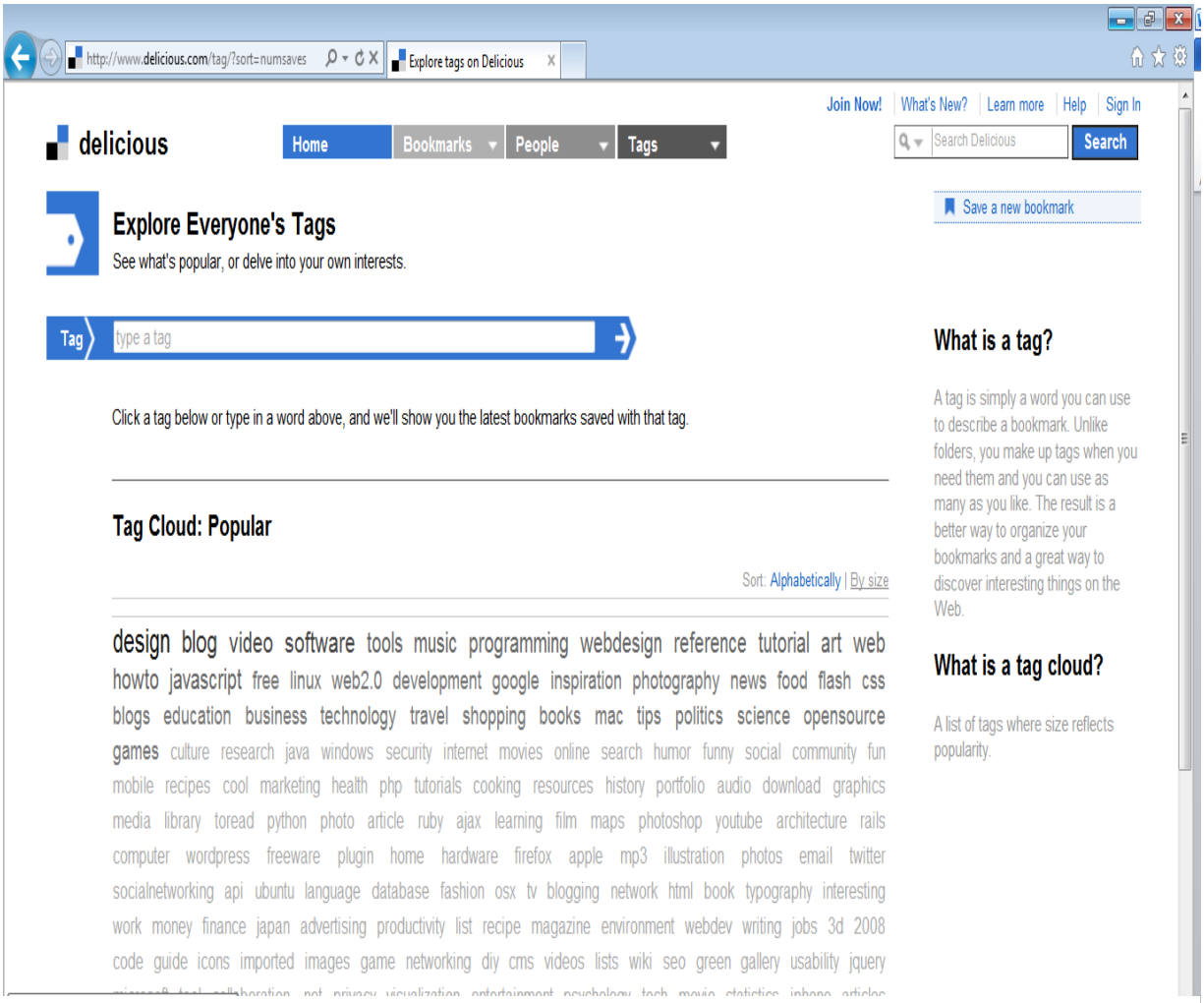


Figura 3 – Página inicial site Delicious. Fonte: Delicious (2015).

De acordo com Quintarelli (2005), segundo a utilização das etiquetas, há dois tipos de Folksonomia: as largas e as estreitas. As Folksonomias largas têm como exemplo o site Delicious (DELICIOUS, 2015), que permite adicionar e compartilhar sites preferidos de cada usuário. Este tipo de site se caracteriza por muitas pessoas utilizando um pequeno número, porém popular, de etiquetas, mesmo que essas etiquetas possam ser pouco conhecidas para descrever os itens.

Na página do Site Delicious (DELICIOUS, 2015), é possível consultar as *tags* mais populares utilizadas pelos usuários do site, como mostra a Figura 3.

Já as Folksonomias consideradas estreitas, têm como exemplo o Flickr (FLICKR, 2015). Este site hospeda e compartilha imagens fotográficas e as Folksonomias estreitas são caracterizadas por poucas pessoas utilizando uma ou mais etiquetas, o que significa compartilhar vocabulários próprios e não termos tão populares como nas Folksonomias largas.

As Folksonomias estreitas perdem a riqueza de massa, mas beneficiam o etiquetamento de objetos que não são facilmente encontrados pelas tradicionais ferramentas de busca. Pessoas que compartilham vocabulários próprios podem recuperar itens de forma mais simples e eficiente. Já as Folksonomias largas podem ser úteis para investigar os termos mais usados por grande número de pessoas e para a extração de termos que possam vir a servir de insumo para a construção de vocabulários controlados.

A tarefa de atribuição de *tags* aos conteúdos digitais realizadas por profissionais especializados pode ser incrementada a partir da observação de ocorrências como as supracitadas. Os vocabulários controlados podem vir a ser mais eficientes caso possam se beneficiar de termos usuais sugeridos pelos próprios usuários, porém muitos estudos ainda precisam ser feitos para uma melhor investigação sobre a efetiva utilidade dos termos usados nas Folksonomias largas e estreitas.

Portanto, Folksonomia é o resultado da etiquetagem dos recursos da *Web* num ambiente social (compartilhado e aberto), pelos próprios usuários da informação, visando a sua recuperação posterior. Destacam-se, portanto, três fatores essenciais: 1) é resultado de uma indexação livre do próprio usuário do recurso; 2) objetiva a recuperação a posteriori da informação e 3) é desenvolvida num ambiente aberto que possibilita o compartilhamento e, até, em alguns casos, a sua construção conjunta.

Sucintamente, pode-se dizer que as ferramentas de Folksonomia permitem aos usuários da *Web* indexar os recursos a partir da atribuição de etiquetas para seu armazenamento, organização e recuperação. Além disto, estas ferramentas permitem que as etiquetas fiquem disponíveis em rede (na *Web*), de forma que outros usuários com os mesmos interesses possam aceder aos recursos, além de mostrar as várias formas pelas quais um mesmo recurso foi indexado por outros usuários. É uma maneira colaborativa e livre de indexar, que geralmente não se pauta em nenhum vocabulário controlado ou qualquer outro sistema pré-definido de classificação tradicional.

Assim, surge uma metodologia da construção do conhecimento, a 'etiquetagem social' (YUNTA, 2009; JIMENEZ, 2009) para organizar a representação de conteúdo na *Web*. Pontos positivos destes sistemas de classificação distribuída são a constante atualização da terminologia, a dinamicidade das marcações, flexibilidade e interatividade entre aplicações. Pontos negativos também são apontados, como a imprecisão conceitual, o alto nível de subjetividade das marcações e a inconsistência no vocabulário (NORUZI, 2006).

A hibridização, em outras palavras, a comunicação entre ferramentas, favorece a eficácia desses aplicativos, junto à interoperabilidade, que: "é considerada como a capacidade de sistemas (autônomos ou não) comunicarem de modo transparente entre si, devido à adoção de padrões comuns e protocolos que permitem o uso compartilhado de informações" (MOURA, 2009).

Interoperabilidade é quando as ferramentas sociais, como twitter e youtube, por exemplo, estão em sintonia (autorizada pelo usuário), de modo que quando se classifica um vídeo como favorito, automaticamente essa informação é disponibilizada no twitter. Portanto: "a Folksonomia demonstra ser uma estratégia viável para a classificação de informações em redes sociais, principalmente por sua flexibilidade em acomodar a diversidade cultural que tais redes se propõem a acolher" (AMSTEL, 2007).

Por outro lado, a tecnologia como dispositivo de divulgação da 'rede universal do conhecimento', também colabora para a padronização cultural (PEREIRA, 2007), ao mesmo tempo em que estimula a emergência de tribos e segmentos diversificados. Essa padronização cultural pode ser percebida porque os públicos se diluíram (não há separação entre autores e leitores) e mais do que 'centrais ou matrizes culturais' existem "circuitos culturais" (JENKINS, 2008), que sustentam comunidades virtuais.

A definição de um vocabulário depende de regras estabelecidas socialmente entre uma comunidade específica, e isso depende de valores culturais já estabelecidos. Como a *Web* viabiliza o relacionamento de comunidades com identidades bastante diversas, variados são os significados das marcações de módulo para módulo da informação. As linguagens de marcações passam a permitir anotações diversas dos conteúdos, promovendo certa interconexão semântica. Por outro lado, a *Web* Semântica não pode prover a interconexão perfeita entre sujeitos com valores cognitivos diferentes (CORY, 2008). Mas, segundo Cory, ações tecnológicas resultam em reações culturais. Por exemplo, a questão autoral, que se transforma diante da mobilidade e diversidade dos novos dispositivos; a necessidade de privacidade diante da vigilância das câmeras; as Wikipédias e comunidades de fãs, são acontecimentos que indicam a influência da tecnologia sobre a cultura.

2.1 INFLUÊNCIA DA *WEB* 2.0 NA FOLKSONOMIA

A *Web* 2.0 é um conjunto de práticas e princípios que definem padrões e modelos de negócios para uma nova geração de softwares na rede mundial de computadores (O´REILLY,

2005). Ela caracteriza-se por arquiteturas de informação participativas, com foco para a gestão colaborativa de grandes bases de dados.

Exemplos como o Del.icio.us (DELICIOUS, 2015), criado em 2003 e o Flickr (FLICKR, 2015), lançado em 2004, permitem a organização e compartilhamento de URLs e imagens, respectivamente. Eles são a materialização do que o fundador da internet, Tim Berners-Lee, idealizou como a função cognitiva da rede, explícito no artigo *"The Semantic Web"* (BERNERS-LEE, 2001). A *Web* semântica tem como objetivo estruturar dados, para que buscadores identifiquem conteúdos pertinentes aos assuntos procurados (BREITAM, 2005; FEITOSA, 2006; MIKA, 2006; PICKLER, 2007).

A característica de cooperação da *Web* 2.0 não se manifesta somente com a criação do termo. Em 1999, bem antes de ouvirmos falar em *Web* 2.0, foram criadas as primeiras ferramentas de construção de blogs, o Blogger e o Groksoup, lançados pela Pyra. Em 2001, foi lançada a Wikipédia, enciclopédia online construída pelos próprios usuários da *Web*, os quais têm plena liberdade de criar e editar os artigos.

Em 2004, O'Reilly cunhou o termo *Web* 2.0 em seu artigo onde define:

> "*Web* 2.0 é a mudança para uma internet como plataforma, e um entendimento das regras para obter sucesso nesta nova plataforma. Entre outras, a regra mais importante é desenvolver aplicativos que aproveitem os efeitos de rede para se tornarem melhores quanto mais são usados pelas pessoas, aproveitando a inteligência coletiva". (O`Reilly, 2004).

A Folksonomia tem como característica principal a participação intensificada do usuário nos processos de etiquetação das informações, o que é apenas possibilitado com o desenvolvimento da *Web* 2.0. Aquino (2007) chama de hipertexto 2.0 o novo hipertexto surgido pelos links formados pelas *tags*. A Folksonomia, portanto, é uma derivação, ou até uma evolução da Taxonomia. Sua origem, na língua inglesa, significa uma classificação dada pelos usuários (folks = pessoas + taxonomia).

O'Reilly (2005) diz que a *Web* 2.0 não tem limites rígidos, mas sim um núcleo gravitacional, já que é encarada como uma plataforma, na qual os próprios usuários controlam seus dados. O que caracteriza essa nova fase da *Web* é a existência de serviços e não de pacotes fechados de software. Sua arquitetura é edificada sob a cooperação, os dados possuem origem distintos e podem ser transformados. Com a *Web*, a publicação e o acesso à informação tornaram-se ações de fácil execução para quaisquer indivíduos. As pessoas, ao redor do mundo, passaram a ter em suas mãos a possibilidade de participar ativamente nesses processos. Desde a sua criação, a *Web* tem evoluído com a adição de novos serviços e

funcionalidades que, cada vez mais, permitem que os seus usuários participem de forma ativa na construção e organização dos conteúdos lá disponíveis.

Um exemplo clássico desta nova geração da *Web* 2.0 é a Wikipedia, uma enciclopédia dinâmica, na qual os próprios usuários disponibilizam e editam informações.

De acordo com Davis (2005), a *Web* 2.0 é uma atitude e não uma tecnologia. Uma atitude onde encoraja a participação dos internautas através de serviços e aplicações abertos. Para o autor, "abertos", no sentido técnico, refere-se a APIs (*Application Programming Interface*) apropriadas. Porém, o mais importante é o fato de ter a característica de ser socialmente aberta e suportada por tecnologias que proporcionam uma interoperabilidade de serviços nunca antes existente nas plataformas que antecederam a *Web* 2.0.

Em seu texto *"Talis, Web 2.0 and All That"*, faz considerações a respeito destas características e afirma que a *Web* sempre teve a finalidade de ser participativa, e sem esta característica não haveria a colaboração que acontece atualmente. O que ocorre, segundo o autor, é que nos últimos anos a *Web* perdeu um pouco o contato com suas raízes devido a interesses que influenciaram o seu desenvolvimento. Davis considera que a *Web* 2.0 seja, então, um conjunto de esforços para que a *Web* volte aos seus ideais originais, o que implica uma regeneração em uma nova versão. O que é importante é que há uma movimentação neste sentido, tecnologias que atentam para a face social da *Web* proporcionando a interação direta dos usuários.

Ainda há muito a ser estudado e desenvolvido nesta nova geração da *Web*: suas características, tecnologias e inovações. A dinamicidade das evoluções que ocorrem nos tempos de hoje pode ser considerada como um novo paradigma para a organização dos conteúdos dos recursos digitais na *Web*. A possibilidade de os próprios usuários participarem na organização desses conteúdos é, em especial, uma questão que vale ser pesquisada e implementada. Esta nova abordagem relativa à indexação dos recursos digitais da *Web* toma, de forma genérica, a designação de Folksonomia.

Logo, Folksonomia trata-se de um novo conceito que tem sido utilizado por diversos profissionais e estudiosos da área de informação. No entanto, parece não haver ainda um consenso na área, quer sobre a utilização deste termo, quer sobre o seu significado. Há os que preferem utilizar outros termos como, por exemplo, classificação social ou *social tagging*.

2.2 ASPECTOS COLABORATIVOS E SOCIAIS DA FOLKSONOMIA

A Folksonomia significa o uso da linguagem natural em arquitetura da informação pelas quais redes sociais alimentam bancos de dados (MIKA, 2006; SPITERI, 2007; CATARINO e BAPTISTA, 2006). O processo se dá quando as pessoas marcam com etiquetas (*tags*) e/ou com palavras-chaves (metadados) conteúdos a ser evidenciados, agregam valor formando nuvens de indicadores (*tag clouds*), em contextos sociais participativos. A indexação surge de maneira *bottom-up*, forma-se um "sistema de classificação de conteúdos *Web* gerado pelos usuários que permite aos mesmos atribuir *tags* aos seus recursos digitais favoritos através de palavras ou frases selecionadas de uma linguagem natural" (NORUZI, 2006).

A Folksonomia proporciona que as pessoas classifiquem conteúdo de diversas maneiras. A cada vídeo inserido no YouTube (www.youtube.com), por exemplo, pode ser colocada uma *tag* diferente, tornando-a específica. No momento da busca a *tag* servirá para aquele vídeo. Outro exemplo é o uso de *tags* para classificação de conteúdo dentro de um site de *bookmark* pessoal. A cada novo endereço inserido pode-se relacionar uma ou mais *tags*. Quando houver a busca, serão listados os conteúdos relacionados a uma *tag* específica ou a todas. Uma busca específica depende muito da organização que o sistema dispõe. Podem-se citar relações entre pessoas ou grupo de pessoas e um conteúdo específico. Ou entre pessoas e grupos de interesse.

Esses sistemas de classificação distribuída apresentam pontos positivos como a constante atualização da terminologia, a dinamicidade das marcações, flexibilidade e interatividade entre aplicações. Os pontos negativos são a imprecisão conceitual, o alto nível de subjetividade das marcações e a inconsistência no vocabulário (NORUZI, 2006).

A demarcação de um vocabulário está associada a uma série de regras estabelecidas socialmente entre uma comunidade específica, e isso depende de valores culturais. Como a *Web* põe em contato comunidades com identidades bastante diversas, são variados os significados das marcações de módulo para módulo da informação. As linguagens de marcações passam a permitir anotações diversas dos conteúdos, originando certa interconexão semântica. De acordo com CORY (2008) a *Web* Semântica não pode prover a interconexão perfeita entre sujeitos com valores cognitivos diferentes (CORY, 2008). O autor ainda afirma que as ações tecnológicas resultam em reações culturais. Por exemplo, a questão autoral, que se transforma diante da mobilidade e diversidade dos novos dispositivos; a necessidade de

privacidade diante da vigilância das câmeras; as Wikipédias e comunidades de fãs são fenômenos indicadores da influência da tecnologia sobre a Cultura.

Hoje em dia a *Web* permite o uso de linguagens mais flexíveis e de padrões cada vez mais aceitos de representação da informação. Isso a transforma em uma rede de conhecimento, e não apenas em um espaço onde coabitam dados sem conexão. Quando citado o termo *Web* Semântica, nesse sentido, não é somente como sinônimo de 'significado', mas sim uma arquitetura formal e compreensível por máquinas de representação. A *Web* 2.0 é um conjunto de práticas e princípios que definem padrões e modelos de negócios para uma nova geração de softwares na rede mundial de computadores (O´REILLY, 2005), caracterizada por arquiteturas de informação participativas, com foco para a gestão colaborativa de grandes bases de dados.

A arquitetura da Web Semântica é composta por mecanismos estruturados, rígidos, formais e complexos (como ontologias, RDF, RFF Schema, OWL, sistemas de inferência), que representam significados (conexão com as camadas culturais semânticas). O objetivo da Web Semântica é permitir que máquinas (na forma de agentes de software inteligentes) entendam o significado dos recursos descritos culturalmente. *Tags, posts, blogs, wikis* são instrumentos desta '*Web* Social', que possibilitam a comunicação e a organização coletiva dos dados, fazendo a interface com mecanismos complexos, controlados e formais de representação computacional da informação. *Web* Semântica é como o conjunto de instrumentos que fazem parte da *Web* 2.0.

A atribuição do termo *Web* Semântica para analisar o efeito social dos instrumentos semânticos (como *tags*) na comunicação ocorre porque a informação circula pela rede transformando as culturas no mundo, produzindo o que Manovich (2001) chama de Transcodificação Cultural, ou seja, não é apenas o computador que transforma a cultura, mas o processo inverso também ocorre. A cultura promove a personalização das interfaces digitais, que se tornam simultaneamente interfaces culturais. Temos acesso, através das interfaces culturais construídas pela tecnologia, aos eventos que ocorrem no mundo, com dinamicidade praticamente em tempo real. Essas interfaces sustentam a comunicação mediada pelo computador, fortalecendo as redes e comunidades virtuais.

Ao contextualizar a rede social no cenário virtual, Recuero (2009) observa que a Internet é responsável por ampliar em larga escala as possibilidades de conexões e de difusão de informações das redes, ressaltando que as redes sociais virtuais são a "teia de conexões que espalham informações, dão voz às pessoas, constroem valores diferentes e dão acesso a esse

tipo de valor [capital social]". Marteleto, Oliveira e Silva (2004) reforçam a contribuição da análise de redes sociais para compreensão do papel do capital social no desenvolvimento das comunidades e da inclusão social destas. Para estes últimos autores, capital social pode ser entendido como normas, valores, instituições e relacionamentos compartilhados que permitem a cooperação dentro ou entre os diferentes grupos sociais" (MARTELETO e OLIVEIRA e SILVA, 2004).

Nesse sentido, analisar as *tags* afetivas eleitas por usuários em redes sociais, significa analisar o comportamento e cultura comunicacional dos usuários presentes. Mais detalhadamente, o comportamento informacional envolve atos individuais e "se refere ao modo como os indivíduos lidam com a informação. Inclui a busca, o uso, a alteração, a troca, o acúmulo e até mesmo o ato de ignorar os informes" (DAVENPORT, 2000), enquanto cultura informacional ou comunicacional "abrange grupos ou organizações – em particular os valores e crenças de um grupo "Por cultura em relação à informação entendo o padrão de comportamentos e atitudes que expressam a orientação informacional" (DAVENPORT, 2000). Tais características apresentadas revelam que a classificação das informações através de *tags* é um processo de comunicação que deixa vestígios do comportamento informacional, que por sua vez consolidam práticas culturais e discursivas.

2.3 APLICAÇÕES DA FOLKSONOMIA

Segundo Catarino e Batista (2007), o uso da Folksonomia teve início como um método para organização de recursos digitais na *Web* (*World Wide Web*), mas hoje já foram desenvolvidos projetos para etiquetamento de vários outros tipos de coleções, como exemplos, sistemas para catalogação de acervos de bibliotecas, sistemas para catalogação do acervo museológico entre outros.

2.3.1 SISTEMAS BASEADOS EM FOLKSONOMIA

Os sistemas, baseados em Folksonomia, permitem que os usuários realizem a categorização de recursos na *Web* como forma de organização para a recuperação futura e compartilhem esses recursos já organizados com outros usuários que utilizam o sistema por meio das *tags* associadas a eles. Atualmente existem muitos serviços em funcionamento que utilizam esses sistemas, dos quais podem ser citados:

Flickr: Compartilhamento de fotos e imagens, onde os usuários inserem suas imagens pessoais e determinam *tags* para uma pesquisa posterior;

Del.icio.us: Bookmark pessoal, onde o usuário pode relacionar links de seu interesse e classificá-los por assunto;

YouTube: Compartilhamento de vídeos onde a busca é feita através de *tags* que os classificam.

Esses sistemas de *tags* acima citados proveem um ambiente social e colaborativo, e podem ser usados em ambientes organizacionais, em filtragem de spams e ferramentas de pesquisa acadêmica, entre outras finalidades. É importante ressaltar que *tagging* possibilita a modelagem de um ambiente onde a percepção e opiniões humanas valem como referência de busca e podem ser uma solução para os problemas de organização de informação. Deste modo, surge uma metodologia da construção do conhecimento, a 'etiquetagem social' (YUNTA, 2009; JIMENEZ 2009) para organizar a representação de conteúdos na *Web*.

2.3.2 CARACTERÍSTICAS

Nesse ambiente são considerados três elementos:

Pesquisa: Refere-se às relações entre informação e classificação entre links e suas *tags*. Critérios de relevância são usados, como por exemplo *tags* mais buscadas ou representação espacial como *clouds* (nuvens).

***Tags*:** Determinada palavra que reflete a classificação de determinado conteúdo. Após análise de uma informação é formada uma relação com palavras que façam sentido ao contexto descrito, sem pontos chaves ou classificação.

Usuários: São as pessoas que fornecem a informação e estabelecem a organização para ela. O usuário é a parte pensante do processo.

2.3.3 VANTAGENS E DESVANTAGENS DA FOLKSONOMIA

Gouvêa e Loh (2007) citam vantagens e desvantagens na utilização de *tags*. As vantagens são:

- Recuperação de documentos;
- Compartilhamento do conhecimento e forma de organização;
- Facilidade para associar classes já existentes, classificando conteúdos pertencentes a mais de um assunto;

- Alternativa para a *Web* semântica;
- Definição de conteúdo secundários;
- Criação descentralizada, causando uma estrutura menos rígida e mais flexível;
- Facilidade de uso, visto que as *tags* são criadas pelos usuários.

No entanto, as desvantagens são:

- Problemas nas relações de significados entre as *tags* e seus conteúdos;
- Problemas ortográficos variados, desde a má grafia até erros de sintaxe;
- Problemas nos usos das *tags* genéricas e específicas.

O fato do uso do vocabulário do usuário (Folksnomia) poder refletir as mudanças de linguagem e adaptar-se aos novos termos, custando relativamente pouco aos usuários e ao sistema que essa adaptação seja feita, essa característica traz também um elemento negativo: em um sistema de classificação (ou categorização) onde cada usuário pode classificar a informação compartilhada, e onde não há vocabulário controlado termina por surgir, evidentemente, problemas com o uso de acrônimos, controle de singular e plural, sinônimos, regionalismos, ambiguidade, etc. os problemas surgem quando, por exemplo, os usuários utilizam a mesma *tag* de formas completamente diferentes. A falta de controle de sinônimos faz por outro lado que diferentes *tags* sejam usadas para o mesmo conceito.

Uma vez que o sistema não possui controle de sinônimos, ocorre que diversas *tags* são utilizadas para marcar conteúdos de significados similares. Assim, "Mac", "Macintosh", "PowerMac", "Apple" e "g4" são utilizados para material relacionado ao computador Macintosh. Desta forma, alguns sistemas utilizam as *tags* relacionadas. Caso se realize a busca por apenas um termo, todos os outros termos utilizados para a descrição de conteúdos similares surgem listados na página.

O uso de acrônimos também representa uma área sujeita a ambiguidade. A *tag* "ISA", por exemplo, aparece tanto etiquetando a tecnologia de interface ISA nos hardwares de computadores (*Industry Stantard Architecture*), como também é utilizada para a página do Instituto Sócio Ambiental, além do serviço de suporte técnico IBM (*IBM Support Assistant*) ou ainda o site *International Studies Association*. Outros fatores que também prejudicam são o plural e singular. A lista das *tags* utilizadas no CiteUlike (*"most active tags")* apresenta tanto *"network"* como *"networks"*.

Uma variedade de termos inclui informações para a descrição do conteúdo (metadados) que possuem relação significativa para o usuário da informação. Sem um

controle de vocabulário, ocorre a falha de diversos termos a serem utilizados para designar um único conceito, e o inverso também ocorre, com um único termo para conceitos totalmente diferentes.

Podem ser citados os seguintes exemplos: as *tags* "bola", "gol", "jogador", "partida", "placar", "vitória" e "estádio" para identificar um site que traz informações sobre futebol ou qualquer outro tipo de esporte que possui os elementos citados. Por outro lado, a *tag* "música" é utilizada para imagens de instrumentos musicais, mas também para a imagem de uma partitura musical e ainda um grupo de pessoas tocando instrumentos ao ar livre. É notório que na hora de se utilizar o termo para buscar apenas um dos elementos apresentados, haverá dificuldade.

Outra falha que pode ser citada é um termo totalmente inadequado ao conteúdo, como procurar o termo "*soccer*" e ter como resultado informações sobre um parque localizado na divisa entre estados onde se realiza esporadicamente partidas de futebol, ou, não obter informação qualquer do porquê determinada informação recebeu uma *tag* que aparentemente não se encontra relacionada a ela.

Para minimizar as limitações dos sistemas, Merholz (2005) explica que a Folksonomia pode ser utilizada para o desenvolvimento de um vocabulário profissional controlado. O autor confronta o sistema a uma estrada em formação, onde múltiplos caminhos (múltiplas *tags*) seriam percorridos, mas somente os mais usados seriam "pavimentados". Ferramentas como o *Getty Thesaurus of Geographic Name*s *Online* podem auxiliar a estabelecer relações entre *tags* sinônimas e construir um vocabulário realmente utilizado pelo usuário. "Uma vez que se tenha um sistema estabelecido, você pode usar as *tags* mais comuns para desenvolver um vocabulário controlado que fala a língua do usuário" (MERHOLZ, 2005).

Uma característica a ser destacada é o fato de o usuário poder navegar entre os termos relacionados, revelando informação que não julgava existir a princípio. "É similar à diferença entre explorar um campo de problema para formular questões, ao contrário de realmente procurar por respostas para questões formuladas" (MATHES, 2004).

O usuário, nesse sistema, pode ter uma vaga ideia de sua necessidade de informação, a sua necessidade inicial, para ele uma dúvida ainda não formulada inteiramente, e ir modificando ou adaptando sua busca conforme navega pelas diferentes possibilidades que o sistema lhe oferece.

Analogicamente, pode-se imaginar que seria como vasculhar a estante da biblioteca, com uma dúvida ou necessidade ainda não formulada na mente. “Preciso de informações que tratem de automóveis antigos”. Entretanto, nas informações disponíveis pode-se encontrar documentos que se relacionem ao assunto “automóveis antigos”, mas de uma forma que inicialmente não se projetou: “Automóveis antigos, acidentes com automóveis, o trânsito da cidade de Londres, design e fabricação de modelos de automóveis”, entre outros.

A *Web* e os sistemas de Folksonomia trazem um volume tão significativo de informação que a questão passa a ser o esforço de concentração na objetividade em responder a necessidade inicial. É muito comum o usuário buscar determinada informação e terminar por encontrar uma quantidade de assuntos que ele não tinha mentalizado inicialmente. Isto é um fator que dificulta o alcance de uma resposta a uma questão determinada e formulada. Por outro lado, é inegável que projeta o alcance das informações a um nível que certamente não se imaginaria possível anteriormente.

Mesmo com os problemas inerentes ao sistema, alguns fatores levam ao êxito do emprego dessa estrutura de organização das informações. Entre eles, o pequeno esforço do sistema e de seus usuários para que seu vocabulário se adapte ao conteúdo, em comparação com a inviabilidade de sistemas complexos e hierárquicos para seu uso. A Folksonomia tem o seu foco na comunicação e no compartilhamento dessas informações, ao contrário de uma organização pessoal.

Para Butterfield (2004), um dos criadores do Flickr (FLICKR, 2015) é justamente a falta de hierarquia, de controle de sinônimos e de precisão semântica que faz com que ela (a Folksonomia) funcione.

Apesar de suas desvantagens, a utilização da etiquetagem feita pelos usuários tem sido adotada em sites com os mais variados conteúdos, com o intuito de organizar e separar as informações para sua possível busca no futuro. Essa busca pela organização das informações não é uma necessidade surgida com a quantidade de informações presente na rede. Ela vem desde os primórdios da humanidade, quando o homem sentiu a necessidade de identificar as coisas existentes no mundo. Com o tempo e a evolução dos estudos, foi criado um método para organizar e sistematizar informações, a Folksonomia.

2.4 DELICIOUS

O Delicious é um serviço *Web* 2.0 de gerenciamento de favoritos e rede social, foi desenvolvido por Joshua Schachter tendo realizado sua formação acadêmica em engenharia elétrica e da computação na Universidade Carnegie Mellon em Pittsburgh nos EUA. O site (DELICIOUS,2015) foi lançado no final do ano de 2003 e logo em seguida, já em 2005, sendo adquirido pelo Yahoo (2015). Em 2011 o site foi comercializado para os fundadores do Youtube (2015), Chad Hurley e Steve Chen e passou a fazer parte da empresa AVOS Systems. Recentemente, o Delicious esteve envolvido em uma nova negociação sendo comercializada para a empresa Science Inc que atualmente mantem a propriedade sobre o site. O Delicious caracteriza-se como um site de marcadores sociais (*Social Bookmarking*) utilizado para salvar e organizar URLs (*Uniform Resource Locator*) e permite descrevê-las através do uso de *tags* possibilitando ao usuário a encontra-las.

> "Sites de marcadores sociais visam armazenar, etiquetar, organizar e compartilhar de maneira colaborativa links considerados relevantes pelos usuários. Sendo assim, são planejados de maneira a valorizar o conhecimento compartilhado, dando maior relevância àqueles recursos mais populares" (COBO e PARDO, 2007).

O site apresenta em sua tela inicial, conforme apresentado na Figura 4, o conceito "Todas suas coisas em um lugar só. Acesse seus *bookmarks* a partir de qualquer computador, a qualquer momento, em qualquer lugar". Além disto, apresenta as *tags* relacionadas ao do endereço *Web* pesquisado.

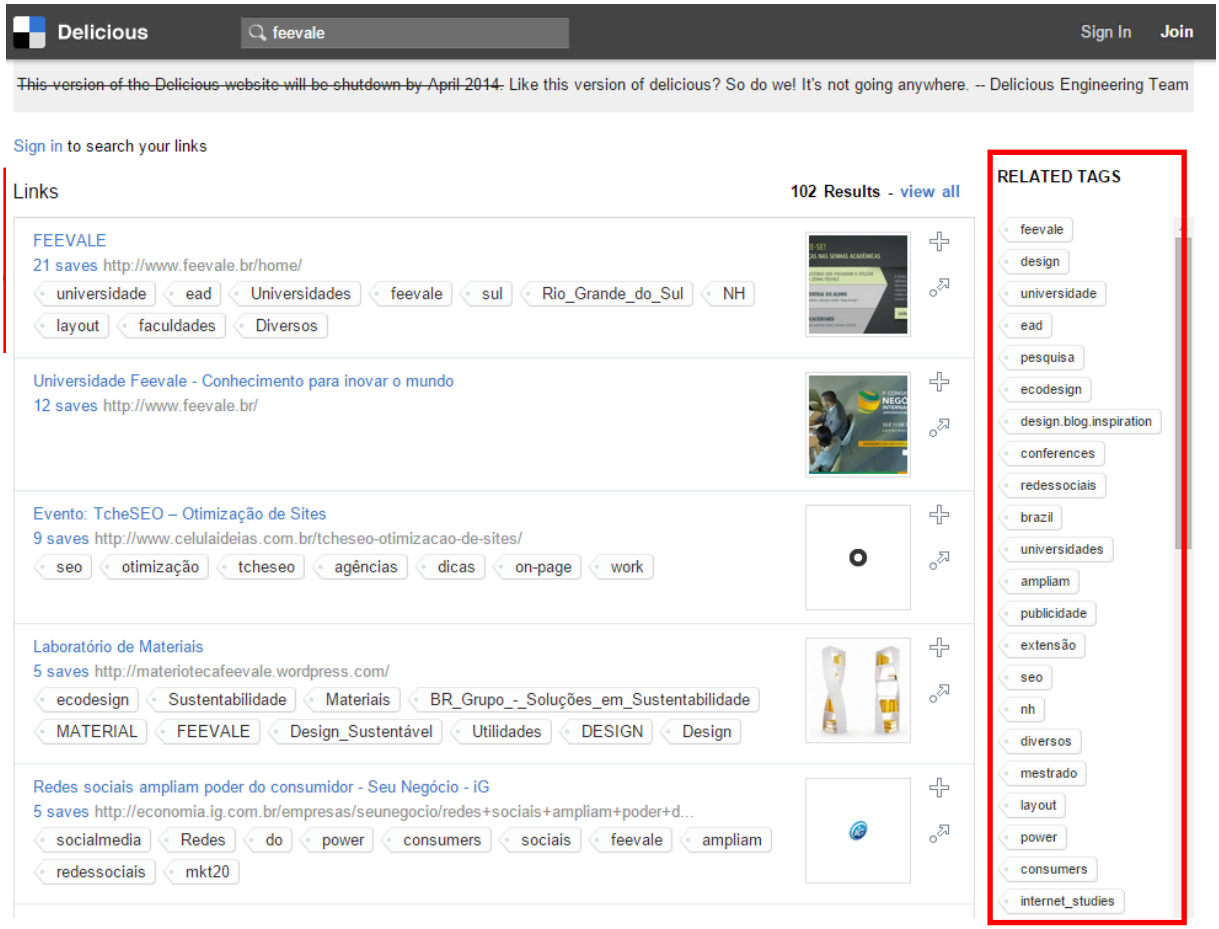


Figura 4 – Site Delicious (*Tags* relacionadas). Fonte: Delicious (2015).

Podemos observar nesta página alguns elementos importantes deste sistema.

1. *Related tags*: As *tags* mais utilizadas para o endereço.
2. *Links:* Endereços existentes para o domínio pesquisado (Endereço).
3. Lista de *tags* por endereço: grupo de *tags* por endereço.

2.4.1 API DELICIOUS

Na página *Web* do Delicious está disponível uma API (*Application Program Interface)* para que aplicações de terceiros possam usufruir do serviço. API pode ser encontrada através do enderenço (https://github.com/SciDevs/delicious-api). Através dela é possível inserir, consultar e salvar *tags* para os endereços *Web* desejado integrado em um sistema terceiro.

3 CLASSIFICADORES

A classificação é uma das possíveis tarefas de mineração de dados em que os algoritmos buscam por padrões que classifiquem os dados, ou seja, por classificadores (FAYYAD, 1996). No contexto de classificadores, um exemplo, instância ou registro, é um conjunto ordenado de variáveis, descrevendo um objeto de interesse. Cada variável, também chamada de atributo ou característica, possui valores de um conjunto pré-definido de valores que são dependentes do problema, e que descrevem algum aspecto do exemplo. Este conjunto pré-definido de valores é conhecido como domínio da variável. Existem dois tipos de domínios (GAMA, 1999):

a) Domínio nominal: também conhecido como domínio discreto, do qual pode-se enumerar todos os valores possíveis que uma variável pode ter. Exemplo: {vermelho, azul, amarelo};
b) Domínio contínuo: do qual nem sempre é possível enumerar todos os valores possíveis. Exemplo: temperaturas do corpo.

Cada exemplo possui uma variável especial, também conhecida como atributo categórico, rótulo ou classe, que descreve o fenômeno de interesse, sobre o qual será realizada a predição e, consequentemente, a classificação.

Um conjunto de dados usado na mineração de dados é constituído por um conjunto de exemplos, que possui **n** exemplos e ***m*** atributos. O conjunto de treinamento é um conjunto de exemplos rotulados e o conjunto de teste é um conjunto de exemplos não rotulados. O conjunto de teste também pode ser um conjunto de exemplos rotulados, proveniente do conjunto de exemplos, usado para estimar a qualidade do classificador (IBM, 2015).

Ao algoritmo de indução é fornecido um conjunto de treinamento. A tarefa do algoritmo é gerar um bom classificador a partir deste conjunto de instâncias classificadas. O classificador pode, então, ser usado para classificar instâncias não rotuladas, fazendo uma predição correta do rótulo de cada uma dessas novas instâncias.

Assim, pode-se considerar a classificação como uma tarefa de previsão, em que um conjunto de atributos previsores (características) é usado para prever um atributo meta (rótulo). A tarefa consiste em descobrir o relacionamento entre os atributos previsores e o atributo meta, usando um conjunto de exemplos cuja classe é previamente conhecida. Após o

aprendizado, o objetivo é utilizar o relacionamento descoberto para prever a classe de novos dados, a partir, apenas, de seus atributos previsores (GAMA, 1999).

O classificador também pode ser avaliado em relação à sua acurácia (precisão), compreensibilidade, tempo de aprendizado, requisitos de armazenamento, compatibilidade (simplicidade) e alguma outra propriedade que defina o quão bom ele é para esta determinada tarefa (BARANAUSKAS, 2001).

De maneira geral, o processo de classificação pode ser ilustrado pela Figura 5, em que o conhecimento do domínio, proveniente do especialista, pode ser utilizado ao escolher dados ou como entrada do indutor.

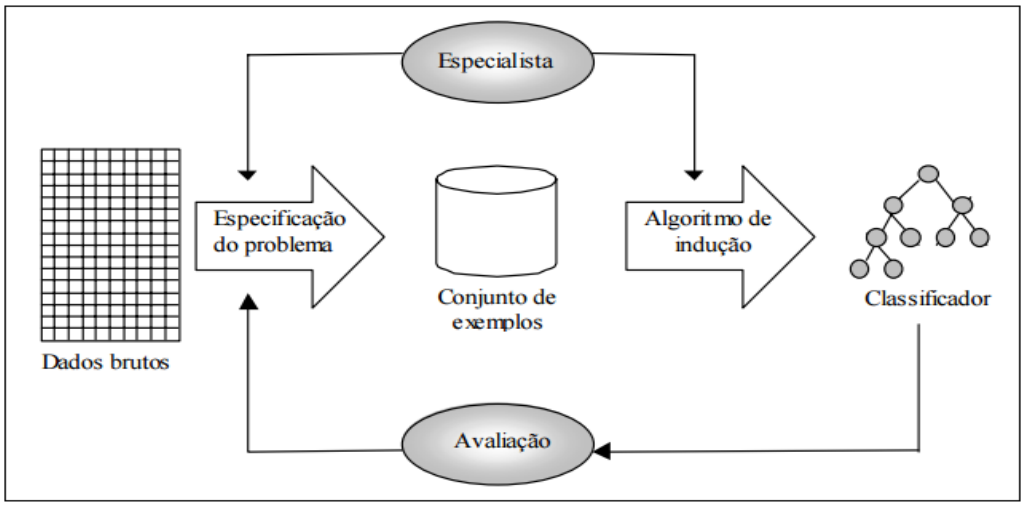


Figura 5 – Processo de classificação. Fonte: CLARK (1989).

Para exemplificar o processo de classificação, suponha que se queira aprender uma forma de predizer se um paciente tem problemas cardíacos. Para isso, é necessário verificar os históricos de pacientes com informações de idade, sexo, dor no peito, nível de colesterol, nível de sedentarismo, taxa máxima de batimentos cardíacos, entre outras características. Cada registro histórico é rotulado por um especialista médico como saudável ou doente. Após, o conjunto de exemplos composto pelos históricos é fornecido como entrada para o algoritmo de indução. A saída resultante consiste de algumas regras que permitem classificar novos pacientes, verificando se um novo paciente apresenta ou não problemas cardíacos.

Este exemplo mostra apenas uma aplicação da grande variedade de domínio em que os classificadores podem ser aplicados. Entretanto, segundo Clark (CLARK, 1989), existem alguns requisitos que um classificador deve possuir:

a) **Classificação precisa:** as regras induzidas deverão ser hábeis para classificar novo exemplos corretamente, permitindo um bom poder de predição;

b) **Regras simples:** devido a compreensibilidade, o indutor de regras deverá induzir regras tão pequenas quanto possível, permitindo a compreensão dos resultados;
c) **Eficiente geração de regras:** é importante que o indutor tenha capacidade de descobrir padrões também em situações complexas.

Há uma grande variedade de técnicas, ou métodos, que são utilizados na Mineração de Dados, com o objetivo de atingir a tarefa de classificação. Porém, ao selecionar uma destas técnicas, devem ser considerados vários aspectos decisivos para um bom desempenho da ferramenta de Descoberta de Conhecimento, pois algumas técnicas são mais adequadas para trabalhar com grandes volumes de dados ou com grande número de atributos, enquanto outras são boas no tratamento de atributos, quantitativos ou qualitativos. Outro aspecto importante é a qualidade de saída desejada, ou seja, se a técnica é capaz de derivar regras e também de aprender incrementalmente, acrescentando novos conhecimentos aos adquiridos anteriormente. Também a possibilidade de avaliar os resultados gerados do ponto de vista estatístico e a performance geral devem ser consideradas no momento de seleção da técnica de Mineração de Dados.

Entre as técnicas de Mineração de Dados usadas para classificação, as redes neurais possuem capacidade de classificação indiscutível (FONSECA, 1994). Já as árvores de decisão e as regras de classificação são as mais populares e utilizadas.

As árvores de decisão apresentam, como principal vantagem, estruturas simples e de grande legibilidade, expressando seus resultados de uma forma muito clara, que podem ser facilmente compreensíveis e usadas diretamente pelo usuário (ADRIANS, 1997). Elas podem ser aplicadas a grandes conjuntos de dados e são adequadas para qualquer tipo de dados, capazes de manipular atributos contínuos e discretos. Apesar de ser uma das técnicas mais simples e eficientes, elas são propensas ao *overfitting,* que ocorre quando o modelo aprende detalhadamente os padrões ao invés de generalizar (ESPOSITO, 1997).

Já as regras de classificação resolvem a maior desvantagem das árvores de decisão: a dificuldade de compreensão quando aplicadas em grandes bases de dados, pois as regras podem ser compreendidas sem que haja a necessidade de se referenciar a outras regras. Porém, a principal desvantagem das regras de classificação é que elas encontram muitas associações que são apenas ruídos, além de não trabalharem muito bem com atributos numéricos (ADRIAANS, 1997). Estas deficiências podem ser resolvidas, respectivamente, com a adição de um grau mínimo de suporte e de técnicas de discretização.

As representações através de regras de classificação e árvores de decisão são, geralmente, utilizadas em conjunto. Podem ser também equivalentes – se as regras forem

derivadas da árvore, sendo a eficiência da classificação de árvores de decisão mantida quando transformada em regras de classificação. Além dos aspectos citados acima, o conhecimento prévio das técnicas de árvores de decisão e regras de classificação motivaram o estudo das mesmas em sistemas de indução de classificadores.

Neste capítulo, são apresentados diversos aspectos básicos sobre Descoberta de Conhecimento em Base de Dados (KDD), Mineração de dados, Tarefas de Classificação e Árvores de Decisão.

3.1 DESCOBERTA DE CONHECIMENTO (KDD – KNOWLEDGE DISCOVERY DATABASE)

O processo de KDD *(Knowledge Discovery Database)* pode ser definido como um conjunto de atividades contínuas que compartilham o conhecimento descoberto a partir de bases de dados (Steiner et al., 2007). A definição KDD foi formalizado em 1989 em referência ao amplo conceito de procurar conhecimento a partir de bases de dados. Segundo Fayyad et al. (1996), o processo de KDD é um processo não trivial de descoberta de padrões válidos, novos, úteis e acessíveis. A principal vantagem do processo de descoberta é que não são necessárias hipóteses, sendo que o conhecimento é extraído dos dados sem conhecimento prévio.

Muitas vezes a expressão "Mineração de Dados" (do inglês "*Data Mining"* – DM*)* é usada como sinônimo do processo de KDD. *Data Mining* é um passo particular neste processo-aplicação de algoritmos específicos para extrair padrões (modelos) de dados. A descoberta de conhecimentos em bases de dados é caracterizada como um processo composto por várias etapas operacionais. A Figura ilustra as 5 etapas de KDD, de acordo Fayyad et al. (1996):

1. **Seleção dos dados:** etapa de preparação e seleção dos dados utilizados;
2. **Pré-processamento dos dados:** etapa de remoção ou atenuação de possíveis ruídos presentes nos dados selecionados;
3. **Transformação dos dados:** etapa em que são aplicados tratamentos e transformações sobre os dados para melhor adequá-los à extração de padrões;
4. **Mineração de dados *(data mining)*:** busca e extração de padrões nos dados por meio de algoritmos;
5. **Interpretação e avaliação dos resultados (Pós-processamento)**: análise de relevância e refinamento do conhecimento descoberto para o domínio em questão.

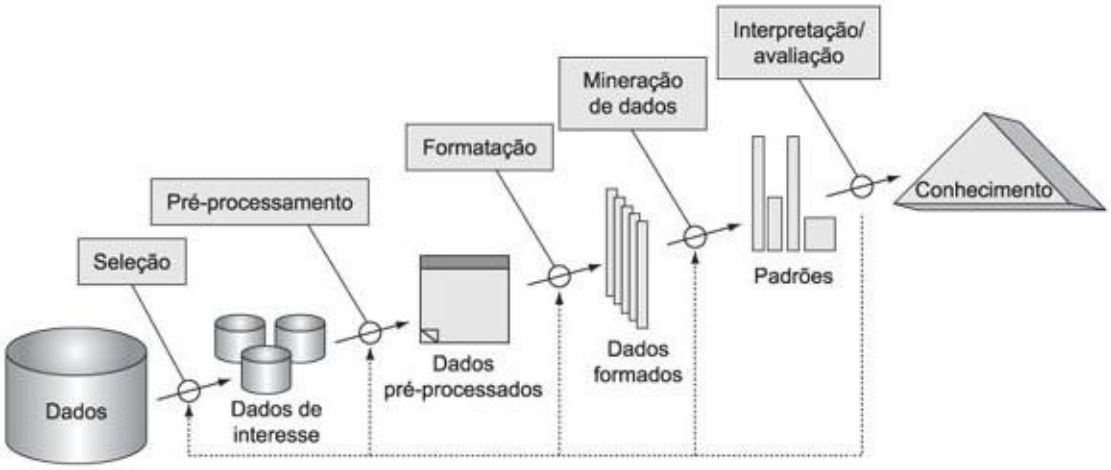


Figura 6 – Etapas KDD. Fonte: Fayyad et al. (1996).

Embora essas etapas devam ser executadas na ordem apresentada, o processo é interativo, pois o usuário pode continuamente intervir e controlar o curso das atividades caso necessite, e iterativo, por ser uma sequência finita de operações em que o resultado de cada uma é dependente dos resultados das que a precedem.

O principal objetivo do processo de KDD é extrair o conhecimento a partir de informações "escondidas" nos dados que sejam úteis nas tomadas de decisões, utilizando métodos, algoritmos e técnicas de diferentes áreas científicas, que segundo Tan, Steinbach e Kumar (2005) incluem Estatística, Inteligência Artificial, Aprendizagem de máquinas e Reconhecimento de Padrões.

3.2 SELEÇÃO DOS DADOS

Conforme apresentam ALMEIDA e DUMONTIER (2008), a tarefa de seleção de dados é crítica pois os dados podem não estar disponíveis em um formato apropriado para serem utilizados no processo de KDD. Ou ainda, mesmo se disponíveis, os dados podem precisar ser rotulados com auxílio de um especialista do domínio.

Existem problemas para a coleta de dados, porém, um dos principais problemas é descobrir onde encontra-los. A maioria dos sistemas de gerenciamento de dados que estão funcionando hoje são proprietários, mas existe uma tendência forte das empresas implementarem novos bancos de dados, que são direcionados para dar suporte às pessoas responsáveis por tomar decisões – chamados de repositórios de dados. O objetivo do

repositório de dados é integrá-los, de diversos sistemas transacionais, da forma mais confiável e seguro possível, contribuindo na seleção dos dados.

3.3 PRÉ-PROCESSAMENTO

O Pré-processamento é uma etapa em geral trabalhosa em função dos dados disponíveis não estarem de tal forma organizados a permitir a aplicação direta aos algoritmos de mineração.

Em resumo, no **Pré-processamento** são realizados os seguintes passos:

a) *Desenvolvimento de um entendimento do domínio da aplicação:* aquisição do conhecimento prévio e das metas do usuário final, identificando a informação requerida para uma ação específica, ou seja, o que o usuário quer conhecer e o que ele quer fazer com esse conhecimento. Isto significa que o especialista e o analista deverão trabalhar junto para especificar o problema a ser resolvido. O domínio da aplicação envolve também a elaboração de uma lista de requisitos; a avaliação do hardware, do software e da qualidade dos dados disponíveis; o desenvolvimento de um inventário das bases de dados disponíveis; a formulação do conhecimento necessário à organização no momento e no futuro; a identificação das pessoas que trabalham com o conhecimento, o tipo de decisões que elas costumam fazer, os padrões que elas utilizam e a funcionalidade que elas necessitam como apoio ao processo de decisão; e a identificação dos processos e transformações que as bases de dados terão que sofrer antes de serem utilizadas;

b) *Seleção dos dados:* criação de um conjunto de exemplos a partir de uma cópia dos dados operacionais armazenados nas bases de dados da organização, nos quais a descoberta será realizada. Isto envolve a definição dos atributos que serão importantes e utilizados para a descoberta, bem como a recuperação dos exemplos necessários. Entretanto, extrair e integrar dados nem sempre é uma tarefa fácil, pois pode envolver, além da conversão em baixo nível dos dados, a união de tabelas relacionais, pois o algoritmo de mineração não consegue trabalhar com múltiplas tabelas;

c) *Limpeza de dados:* definição de quais estratégias serão utilizadas para a manipulação de informações inconsistentes, coleta de informações necessárias ao modelo e realização de operações básicas de limpeza, como remoção de redundâncias, inconsistências e valores nulos. Entretanto, para efetuar a limpeza dos dados é

essencial e necessário, primeiramente, explorar os dados de forma a se familiarizar com os mesmos.

d) *Transformação dos dados:* redução e projeção dos dados, encontrando características comuns para representa-los e reduzindo, assim, o número efetivo de valores de variáveis em consideração. Isto pode significar expressar alguma informação da base de dados de uma forma diferente do original.

3.4 MINERAÇÃO DE DADOS (DATA MINING)

Data Mining ou Mineração de Dados é o processo de pesquisa em grandes quantidades de dados para extração de conhecimento. Utiliza técnicas de Inteligência Computacional para procurar relações de similaridade ou discordância entre dados, com o objetivo de encontrar padrões, irregularidades e regras. Desta forma, transforma os dados, aparentemente ocultos, em informações relevantes para a tomada de decisão e/ou avaliação de resultados.

Conforme QUONIAM (2002), todas as técnicas que permitem extrair conhecimentos de uma massa de dados que, de outra maneira, permaneceria escondido nas grandes bases de dados, chamam-se *data mining*.

A mineração de dados possibilita a busca em grandes bases de dados de informações desconhecidas, permitindo aos gestores uma maior agilidade nas tomadas de decisões. Uma empresa que utiliza *data mining* é capaz de criar parâmetros para entender o comportamento do consumidor, identificando afinidades entre as escolhas de produtos e serviços, prevendo hábitos de compras e analisando comportamentos que possam auxiliar na detecção de fraudes (PINTO, 2005).

Em outras palavras, a etapa de Mineração de Dados, têm-se os seguintes passos:

a) *Escolha da tarefa de mineração de dados:* decisão de qual tarefa será utilizada: classificação, associação, regressão, agrupamento, sumarização, dependência ou desvio;

b) *Escolha do algoritmo de mineração de dados:* seleção da (s) técnica (s) a ser (em) utilizada (s) na próxima etapa e decisão de quais modelos e parâmetros podem ser apropriados. As técnicas mais difundidas e utilizadas são: árvores de decisão, regras de classificação, redes neurais e algoritmos genéticos;

c) *Aplicação da mineração de dados:* busca por padrões de interesse particular em uma forma representacional ou em um conjunto de representações, a partir da base

customizada na etapa de pré-processamento. Pode ser utilizado, também, outro conjunto independente de dados para avaliação do modelo final.

3.5 TAREFAS DE *DATA MINING*

Em mineração de dados esse processo envolve o uso de diversas tarefas e técnicas. As tarefas são classes de problemas, que foram definidas através de estudos da área. As técnicas são grupos de soluções, ou seja, algoritmos para os problemas propostos nas tarefas. Cada tarefa pode apresentar várias técnicas e algumas técnicas podem ser utilizadas para solucionar tarefas diferentes. Cada classe de aplicação em *data mining* tem como base um conjunto de algoritmos que serão usados na extração de relações relevantes dentro de uma massa de dados, podendo exercer tarefas de classificação, estimativa, associação e segmentação, entre outras (HARMON, 1988).

Em geral, as tarefas de *data mining* podem ser classificadas em duas categorias:

a) Descritivas: descrevem o conjunto de dados de uma maneira concisa e resumida e descobrem propriedades gerais interessantes nestes dados;
b) Preditivas: constroem um modelo ou um conjunto de modelos, realizam inferências sobre o conjunto de dados disponíveis e tentam predizer o comportamento de novos conjuntos de dados.

As técnicas de DM possuem, por sua vez, vários algoritmos utilizados na solução dos problemas. Muitos desses algoritmos estão disponíveis em *softwares* matemáticos e estatísticos através de pacotes que possuem tarefas específicas de acordo com o problema que se deseja solucionar. As técnicas são apresentadas nas próximas sessões. De acordo com TAN et al. (2005), a classificação pode ser utilizada para os seguintes propósitos: modelagem descritiva e modelagem preditiva.

Na modelagem descritiva, um modelo de classificação é utilizado como uma ferramenta para distinguir exemplos de diferentes classes. Como exemplo, um médico pode utilizar um modelo de classificação para identificar quais são as principais causas (sintomas) de uma determinada doença.

Quando há o interesse em análise descritiva, é desejável que o modelo de classificação seja fácil de interpretar, ou seja, que fique claro ao usuário o porquê de um determinado exemplo pertencer a uma determinada classe.

Na modelagem preditiva, um modelo de classificação é utilizado para classificar exemplos cujas classes são desconhecidas, ou seja, exemplos que não foram utilizados na construção do modelo. Como exemplo, um médico pode utilizar um conjunto de dados históricos de seus pacientes, em que as classes (diagnóstico) já são conhecidas, para construir um modelo de classificação a ser utilizado para diagnosticar novos pacientes. Dependendo da área de aplicação, a interpretação do modelo de classificação não é essencial para a predição.

3.5.1 CLASSIFICAÇÃO

Classificação é uma tarefa de DM que consiste em levantar as características de um objetivo e associar essas características a classes pré-determinadas. Os algoritmos que constituem a tarefa de classificação se utilizam de árvores de decisão ou redes neurais, começando por um treinamento a partir de transações-exemplo (BARAZETTI, 2001).

Os exemplos existentes permitem determinar um conjunto de parâmetros, reunidos em um modelo que no decorrer do processo serão utilizados para discriminar o restante dos dados.

A imagem 7 possui a uma tabela que mostra um conjunto de exemplos sobre a escolha de linguagem de programação para desenvolvimento de aplicações, levando em consideração certas características do sistema.

Aplicação	Comercial	Distribuída	Cli/Ser	Internet	Matemática	Tempo Real	Linguagem
1	S	N	N	N	S	N	**Delphi**
2	S	S	S	N	S	S	**C++**
3	S	S	N	S	S	N	**Java**
4	N	N	N	S	N	S	**Java**
5	N	N	N	N	S	S	**C++**
6	N	S	N	S	N	N	**Java**

Figura 7 – Imagem com a tabela de exemplo para a tomada de decisão (Classificação). Fonte: BARAZETTI 2001.

Ao enviar os registros da Figura 7 na entrada de um classificador, por exemplo, um algoritmo de geração de árvore de decisão, seria gerada a árvore mostrada na Figura 8.

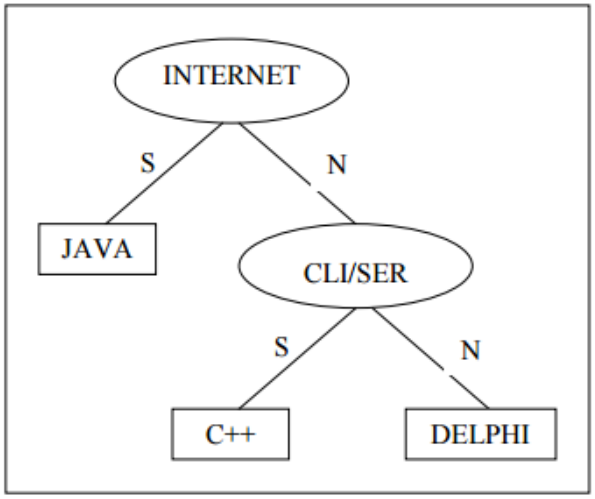


Figura 8 – Exemplo de um algoritmo utilizando árvore de decisão. Fonte: BARAZETTI 2001.

De posse destes dados, um desenvolvedor de software ao analisar a árvore, tomaria a decisão de escolher a linguagem de programação que mais se adaptaria ao propósito da aplicação.

3.5.1.1 ÁRVORES DE DECISÃO

Uma árvore de decisão é uma árvore onde cada nó não terminal representa um teste ou decisão sobre o item de dado considerado. O objetivo principal é separar as classes; tuplas de classes diferentes tendem a ser alocadas em subconjuntos diferentes, cada um descrito por regra simples em um ou mais itens de dados. Essas regras podem ser expressas como declarações lógicas, em uma linguagem como SQL, de modo que possam ser aplicadas diretamente a novas tuplas. O sucesso das árvores de decisão, deve-se ao fato de ser uma técnica extremamente simples, não necessita de parâmetros de configuração e geralmente tem um bom grau de assertividade.

Essa técnica será explanada de forma mais detalhada no capítulo 4, devido a escolha desta na geração do modelo de classificador usado neste trabalho.

3.5.1.2 REDES NEURAIS ARTIFICIAIS

As Redes Neurais Artificiais simbolizam um paradigma no campo da Inteligência Artificial (IA), que permite o desenvolvimento de sistemas computacionais capazes de imitar tarefas intelectuais complexas, tais como a resolução de problemas, o reconhecimento e classificação de padrões, os processos indutivos e dedutivos, dentre outras funções.

Diferentemente de sistemas chamados de heurísticos, já que procuram obter sistemas inteligentes baseados em lógica e em processamento simbólico (por exemplo, sistemas especialistas), as redes neurais artificiais se inspiram no modelo biológico da inteligência, isto é, na maneira como o cérebro é organizado em sua arquitetura elementar, e em como o mesmo é capaz de executar tarefas computacionais (TAFNER, 1998).

Com base nesta inspiração, foi desenvolvido o modelo de neurônio artificial, que procura ser similar ao neurônio biológico, apresentando funções e arquitetura cognitivas parecidas, apesar de serem construídos de matérias distintas, um virtual (produzido em computador) e outro orgânico (FREEMAN, 1992).

Redes Neurais Artificiais (RNAs) são técnicas computacionais que apresentam um modelo matemático para imitar o comportamento do cérebro humano: aprender, errar e fazer descobertas. Ou ainda, são técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimentos através da experiência (HAYKIN, 1999).

Apesar das RNAs tentarem se aproximar ao máximo da estrutura biológica, existe uma grande diferença: enquanto que grandes RNAs podem ter centenas ou milhares de unidades de processamento, o cérebro de um mamífero pode ter muitos bilhões de neurônios. Um sistema de rede neural artificial é composto por várias unidades de processamento, onde, geralmente, são interligadas por canais de comunicação que estão associados a um determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede.

Para executar tarefas, uma rede neural não precisa guardar instruções de comando e executá-las de forma lógica, como em um computador tradicional. Ao invés disso, a rede aprende o que é preciso ser feito e executa a função. Dessa forma, uma mesma rede, se ela for capacitada com os neurônios necessários para tal, é capaz de executar várias funções diferentes, independente de espaço de memória.

As pesquisas em redes neurais artificiais são motivadas pela ideia de que o cérebro humano é um computador, ou seja, um sistema de processamento de informações, altamente complexo e paralelo. Ele tem a capacidade de organizar sua estrutura funcional para a execução das tarefas de processamento muito mais rapidamente que o mais rápido computador digital já construído.

As redes neurais possuem a característica de serem muito apropriadas ao reconhecimento de padrões. As redes neurais podem sofrer um aprendizado, modificando seu comportamento frente a um conjunto de estímulos de entrada (padrão de entrada). Isto será obtido através da alteração dos pesos de atuação das entradas. Logo, as redes neurais são muito adequadas para o reconhecimento e classificação de padrões, pois podem se adaptar para responder a um padrão específico (OSÓRIO, 1991). Elas possuem também uma boa velocidade de processamento, devido ao seu paralelismo interno, possibilitando o desenvolvimento de operações que devem operar em tempo real.

Haykin (2001, p.28) define uma rede neural como:

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

É importante deixar claro que, na prática, vários empecilhos podem ocorrer durante a implementação de uma rede neural artificial que provocam até uma impossibilidade de sua aplicação em alguns problemas. Percebe-se também que as redes neurais artificiais normalmente não conseguem fornecer soluções precisas quando aplicadas isoladamente; precisam ser integradas a outros tipos de paradigmas de sistemas. Ainda há um longo caminho de pesquisas a ser percorrido para que se consiga chegar a uma arquitetura computacional que reproduza fielmente a capacidade do cérebro humano.

As RNAs têm resultados muito eficientes na solução de problemas onde os métodos tradicionais da computação convencional não apresentam soluções satisfatórias. Uma das áreas de maior potencial de aplicação são justamente problemas ligados ao reconhecimento de padrões. Isso se deve principalmente ao fato que imitam, ou pelo menos tentam imitar, o raciocínio humano usado para resolver problemas.

3.5.2 ESTIMATIVA (REGRESSÃO)

Enquanto o problema de classificação lida com valores discretos (sim ou não), o problema de estimativa consiste na determinação de valores contínuos. Através de dados de

entrada, utiliza-se a estimativa para predizer o valor de alguma variável contínua, como a renda, altura ou saldo de cartão de crédito de um indivíduo (Carvalho, 2001).

Como exemplos de tarefas estimativa têm-se: prever a demanda de um consumidor para um novo produto; estimar a renda total de uma família; estimar o valor em tempo de vida de um cliente; estimar a probabilidade de que um paciente morrerá baseando-se nos resultados de um conjunto de diagnósticos médicos; estimar o número de filhos em uma família.

A vantagem dos modelos de estimativa está na possibilidade de ordenar os dados. Com os dados ordenados pela probabilidade da prática de esqui, a empresa enviaria o material promocional somente para os clientes com maior probabilidade de praticar o esporte, obtendo resultados satisfatórios com menores gastos.

Algumas aplicações da técnica de estimativa são:

a) Estimativa do número de crianças em uma família;
b) Estimativa do total de pessoas em uma família;
c) Estimativa da probabilidade de resposta para mala direta;

3.5.3 ASSOCIAÇÃO

A técnica de associação tem por objetivo encontrar relacionamentos ou padrões frequentes entre conjuntos de dados, tais como “90% de pessoas que compram pão também compram leite”. Atualmente, grandes empresas de varejo utilizam-se das regras de associação, pois permitem que o processo de *marketing* seja dirigido, além de servirem para reorganização de *layout* das lojas, onde os itens podem ser agrupados de modo a induzir a venda de artigos relacionados (CERVO, 2002).

Um exemplo de entrada de dados para descoberta de regras de associação pode ser visto na Figura 9. Como se pode observar, a relação entre os itens café, pão e manteiga é de 100%, ou seja, nesta amostra de vendas efetuadas, todos os clientes que compraram café também compraram pão e manteiga.

ID	LEITE	CAFÉ	CERVEJA	PÃO	MANTEIGA	ARROZ	FEIJÃO
1	Não	**Sim**	Não	**Sim**	**Sim**	Não	Não
2	Sim	Não	Sim	Sim	Sim	Não	Não
3	Não	**Sim**	Não	**Sim**	**Sim**	Não	Não
4	Sim	**Sim**	Não	**Sim**	**Sim**	Não	Não

Figura 9 - Entrada de dados para a descoberta de regras de associação. Fonte: RODRIGUES (2000).

A técnica de descoberta de regras de associação é apropriada à tarefa de associação.

3.5.4 SEGMENTAÇÃO (CLUSTERIZAÇÃO)

Nesta técnica, o algoritmo deve criar conjuntos de dados com valores semelhantes a partir de um banco de dados heterogêneo. Uma vez criados os conjuntos, pode-se aplicar um algoritmo de classificação, criando assim regras para os mesmos. Na classificação, os dados são subdivididos e colocados em alguma das classes definidas. A técnica de segmentação, muitas vezes, é uma das primeiras etapas dentro de um processo de DM, já que identifica grupos de registros correlatos que serão usados como ponto de partida para futuras explorações (RODRIQUES, 2000).

Como aplicações de segmentação, pode-se citar:

a) Determinação do número de região de vendas;
b) Determinação de grupos de consumidores potenciais;
c) Agrupamento das casas de uma área de acordo com sua categoria, área construída e localização geográfica.

A Figura 10 mostra quatro segmentos extraídos de um banco de dados de clientes. No primeiro segmento, pode-se perceber um cliente que possui alto salário e carro de luxo, diferentemente do segmento 2 que possui um salário baixo e carro popular. Os segmentos 3 e 4 são quase um mesmo segmento, diferenciado apenas pelo atributo filho.

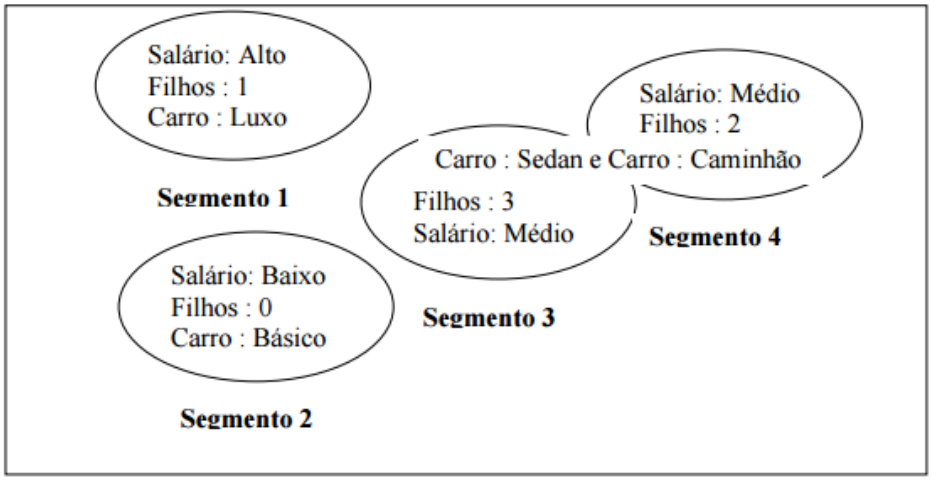


Figura 10 - Clientes divididos em segmentos que possuem semelhanças. Fonte: RODRIGUES (2000).

3.5.5 SUMARIZAÇÃO

É uma tarefa descritiva, que consiste em definir um conjunto mínimo de características que seja capaz de identificar um subconjunto de objetos, em outras palavras, essa tarefa tem

como objetivo procurar, identificar e indicar características comuns entre conjuntos de dados. Ainda, segundo Fayyad (1996), a tarefa de sumarização envolve métodos para encontrar uma descrição compacta para um subconjunto de dados. Um exemplo simples desta tarefa poderia ser tabular o significado e desvios padrão para todos os itens de dados. Métodos mais sofisticados envolvem a derivação de regras de sumarização.

As técnicas de sumarização são comumente aplicadas para análise exploratória de dados e geração de relatórios automatizados. As tarefas de *Data Mining*, descritas acima, são representadas de forma resumida na Figura 11.

TAREFA	DESCRIÇÃO	EXEMPLOS
Classificação	Constrói um modelo de algum tipo que possa ser aplicado a dados não classificados a fim de categorizá-los em classes	?? Classificar pedidos de crédito ?? Esclarecer pedidos de seguros fraudulentos ?? Identificar a melhor forma de tratamento de um paciente
Estimativa (ou Regressão)	Usada para definir um valor para alguma variável contínua desconhecida	?? Estimar o número de filhos ou a renda total de uma família ?? Estimar o valor em tempo de vida de um cliente ?? Estimar a probabilidade de que um paciente morrerá baseando-se nos resultados de diagnósticos médicos ?? Prever a demanda de um consumidor para um novo produto
Associação	Usada para determinar quais itens tendem a co-ocorrerem (serem adquiridos juntos) em uma mesma transação	?? Determinar quais os produtos costumam ser colocados juntos em um carrinho de supermercado
Segmentação (ou *Clustering*)	Processo de partição de uma população heterogênea em vários subgrupos ou grupos mais homogêneos	?? Agrupar clientes por região do país ?? Agrupar clientes com comportamento de compra similar ?? Agrupar seções de usuários Web para prever comportamento futuro de usuário
Sumarização	Envolve métodos para encontrar uma descrição compacta para um subconjunto de dados	?? Tabular o significado e desvios padrão para todos os itens de dados ?? Derivar regras de síntese

Figura 11 – Tarefas de *data mining* de forma resumida. Fonte: Fayyad (1996).

3.6 PÓS-PROCESSAMENTO

A etapa de Pós-processamento compreende o processo de tratamento de conhecimento adquirido por meio do *Data* Mining, com o intuito de facilitar a interpretação e avaliação deste, para priorizar o conhecimento descoberto. Dentre as principais funções desta etapa estão na elaboração e organização, podendo incluir a simplificação de gráficos, diagramas e outros tipos de relatórios demonstrativos, além da conversão da forma de representação do conhecimento extraído. O Pós-processamento tem como principal objetivo apoiar na verificação de até que ponto estes padrões contribuem na solução do problema inicialmente identificado. O Pós-processamento possui as seguintes etapas:

a) ***Interpretação dos padrões enumerados:*** avaliação e interpretação dos padrões encontrados para verificar o que constitui e o que não constitui conhecimento útil. Nesta etapa pode ocorrer a necessidade de retorno a uma das etapas anteriores;
b) ***Consolidação da descoberta de conhecimento:*** incorporação do conhecimento adquirido à organização, documentando-o e relatando-o aos usuários através de técnicas de visualização de dados e implementando alguma estratégia baseada na

descoberta. Nesta etapa deve-se verificar e resolver eventuais conflitos com o conhecimento extraído.

Após a etapa de Pós-processamento, o conhecimento extraído depois de avaliado e validado é consolidado na fase de utilização do conhecimento, sendo incorporado a um sistema para apoio a ao processo de tomada de decisão.

4 ÁRVORES DE DECISÃO

Uma árvore de decisão é um modelo de classificação/regressão cuja estrutura consiste em um determinado número de nós e arcos (ramos), segundo (FURNKRANZ; GAMBERGER; LAVRAC, 2012). Este modelo composto por uma estrutura no formato de uma árvore, aonde cada nó interno representa um determinado teste em uma característica de uma instância, e os arcos representam o resultado do teste realizado. Os nós externos da árvore (nós terminais, ou nós-folha) são referentes às classes de classificação. Para classificar uma determinada instância, a Árvore de Decisão é percorrida de cima para baixo, seguindo pelos arcos nós aonde as características das instâncias satisfazem os testes realizados em cada nó até alcançar um nó-folha, que contém a nova classificação da instância. Na Figura 12 é apresentada uma árvore de decisão simples, e como a mesma realiza a classificação de uma determinada instância.

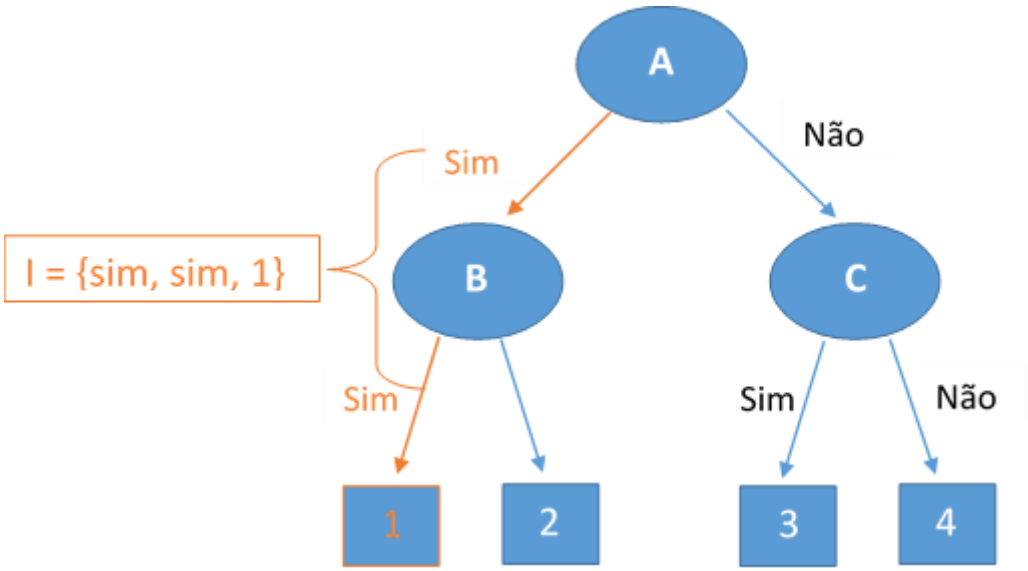


Figura 12 – Árvore de decisão (Simples). Fonte: desenvolvido pelo autor.

O exemplo fornecido pela Figura 12 é bem rudimentar, mas é facilmente compreendido. A instância *I*, contém duas decisões: sim e sim. A árvore então executa o teste em cada nó, e compara com a característica presente na instância. A terceira característica é a classe que a instância pertence: a classe 1. Ao verificar o resultado dos dois testes, a árvore rotula a instância de acordo com a classe presente no nó-folha.

Árvore de Decisão é, provavelmente, o algoritmo de AM mais estudado e aplicado para aplicações em *Data Mining* (WITTEN; FRANK, 2000). Os motivos são variados: é um modelo que possui suporte a diversos tipos de características (categóricas e numéricas), sua

representação de conhecimento adquirido é facilmente compreendida, e o seu processo de aprendizado e treinamento é relativamente rápido, comparado a outros algoritmos.

4.1 CLASSIFICADORES BASEADOS EM ÁRVORES DE DECISÃO

Um classificador tem, como base, um algoritmo de aprendizagem, e este tem como entrada um conjunto de exemplos, constituído de valores de atributos, sendo seu objetivo a saída de um esquema de classificação, que irá predizer classes baseadas nos valores dos atributos. Um meio eficiente para representar classificadores a partir de dados, é o uso de árvores de decisão. Esta técnica tem, como ponto forte, a sua eficiência em termos de tempo de processamento e o fornecimento de um meio intuitivo de analisar os resultados, por apresentar como estrutura final do classificador, uma forma de representação simbólica simples e normalmente bastante compreensível.

Os classificadores baseados em árvores de decisão surgiram no final dos anos 50, tendo, como principal referência, o trabalho de Hunt (QUILAN, 1993) (FONSECA, 1994), que apresenta vários experimentos de indução. Posteriormente, Friedman, trabalhando em novas pesquisas, desenvolveu o algoritmo CART (QUILAN, 1993) (FONSECA, 1994). Também o algoritmo ID3, desenvolvido por Quilan, teve grande influência na área de pesquisa em aprendizado de máquina, tendo, como sucessores, os algoritmos C4.5 (QUILAN, 1993) e C5.0 (NOGUEZ, 2000).

Os fundamentos dos classificadores baseados em árvore de decisão são idênticos, embora sejam muitas as possibilidades existentes para a sua construção. Os algoritmos baseiam-se na sucessiva divisão de um problema em vários subproblemas de menores dimensões, até que a solução para cada um dos subproblemas tenha sido encontrada. Desta maneira, os classificadores baseados em árvores de decisão buscam meios de dividir um problema (conjunto de exemplos) em vários subproblemas (nodos). Esta divisão ocorre até cada um destes subconjuntos conter apenas uma classe, ou até uma das classes demonstrar ser majoritária, não necessitando mais divisões, gerando, em qualquer uma das duas situações, um nodo folha.

A estrutura de uma árvore de decisão é formada por nodos que representam os atributos, por ramos (ligações) provenientes dos nodos, que recebem os possíveis valores do atributo em questão, e de nodos folha, que representam as diferentes classes de um conjunto de exemplos (dados). A Figura 13 ilustra a representação da estrutura de uma árvore de decisão, onde são apresentados seus componentes: nodos, arcos e nodos folha.

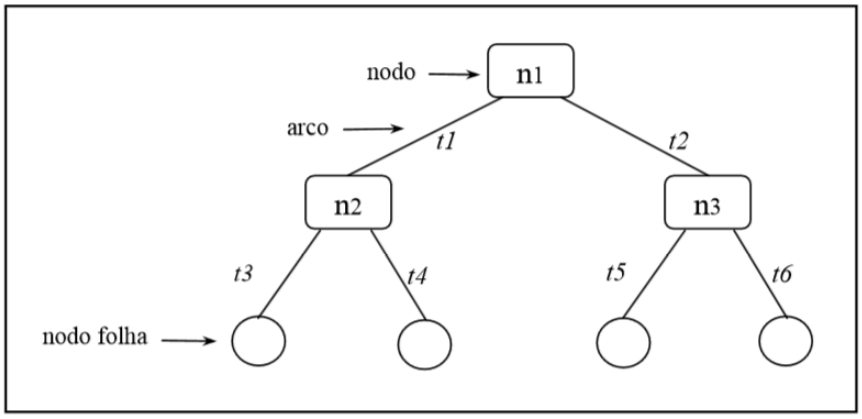


Figura 13 – Árvore de decisão. Fonte: desenvolvido pelo autor.

Dentro da filosofia de árvores de decisão, a classificação consiste em seguir o caminho determinado pelos sucessivos nodos dispostos ao longo de uma árvore, até ser alcançada uma folha que contém a classe a ser atribuída a um novo exemplo de um conjunto de exemplos.

4.2 PRINCIPAIS CONCEITOS VINCULADOS À INDUÇÃO DE ÁRVORES DE DECISÃO

Uma vez construída uma árvore de decisão, seu uso é imediato e muito rápido computacionalmente. Além disso, a interpretação da árvore de decisão é um de seus pontos fortes. No entanto, a construção de uma árvore de decisão, chamado de processo de indução, pode ser uma tarefa de alta demanda computacional. Embora a indução de uma árvore de decisão possa ser realizada de forma manual, usando uma abordagem *top-down*, as principais demandas por árvores de decisão geralmente recorrem a processos automáticos de indução, em uma abordagem *bottom-up*, a partir de dados disponíveis.

Todo processo de indução possui um *bias* indutivo, associado à preferência de uma hipótese sobre outras, supondo a existência de hipóteses que são igualmente consistentes. Há muitas maneiras de uma árvore de decisão ser estruturada a partir de um conjunto de atributos. De forma exaustiva, o número de árvores de decisão possíveis cresce fatorialmente à medida que o número de atributos aumenta. Logo, torna-se impraticável definir a estrutura da árvore de decisão ótima para um determinado problema, devido ao elevado custo computacional envolvido nessa busca.

Nesse sentido, algoritmos baseados em heurísticas têm sido desenvolvidos para a indução de árvores de decisão. Mesmo que eles não garantam uma solução ótima, apresentam resultados satisfatórios em tempo factível.

4.3 TOP-DOWN INDUCTION OF DECISION TREE (TDIDT)

O *Top-Down Induction of Decision Tree* (TDIDT) é um algoritmo muito popular e é utilizado como referência para muitos algoritmos de indução de árvores de decisão, dentre eles os mais conhecidos como ID3 (QUINLAN, 1986), C4.5 (QUINLAN, 1993) e CART (BREIMAN et al., 1984).

O TDIDT elabora regras de decisão de forma implícita em uma árvore de decisão, a qual é construída por sucessivas divisões dos exemplos de acordo com os valores de seus atributos preditivos. De acordo com BRAMER (2007), esse processo é conhecido como particionamento recursivo. O esqueleto do algoritmo de TDIDT é baseado em três possibilidades sobre um conjunto de treinamento T contendo classes C1, C2, ..., Ck:

- T contém um ou mais objetos, sendo todos da classe C_j. Assim, a árvore de decisão para T é um nó folha que identifica a classe C_j.
- T não contém objetos. A árvore de decisão também é um nó folha, mas a classe associada deve ser determinada por uma informação externa. Por exemplo, pode-se utilizar o conhecimento do domínio do problema.
- T contém exemplos pertencentes a mais de uma classe. Neste caso, a ideia é dividir T em subconjuntos que são, ou tendem a dirigir-se para, coleções de exemplos com classes únicas. Para isso, é escolhido um atributo preditivo A, que possui um ou mais possíveis valores O_1, O_2, ..., O_n. T é particionado em n subconjuntos exclusivos T_1, T_2, ..., T_n, onde T_i contém todos os exemplos de T que têm valor O_i para o atributo A. A árvore de decisão para T consiste de um nó de decisão identificando o teste sobre o atributo A, e uma aresta para cada possível resultado, ou seja, n arestas. No lugar de um único atributo A, pode também ser considerado um subconjunto de atributos.

O mesmo algoritmo de indução de árvores de decisão (passos 1, 2 e 3) é aplicado recursivamente para cada subconjunto de exemplos T, com i variando de 1 até n. Basicamente, o algoritmo TDIDT é um algoritmo recursivo de busca gulosa que procura, sobre um conjunto de atributos, aqueles que "melhor" dividem o conjunto de exemplos em subconjuntos.

4.4 ESCOLHA DOS ATRIBUTOS PREDITIVOS PARA OS NÓS DA ÁRVORE

O critério de seleção decide qual atributo preditivo é utilizado em cada nó da árvore. Existem diferentes tipos de critérios de seleção, sendo esta uma das variações entre os

diversos algoritmos de indução de árvores de decisão. Esses critérios são definidos em termos da distribuição de classe dos exemplos antes e depois da divisão (TAN et al.,2005).

A maioria dos algoritmos de indução de árvores de decisão trabalha com funções de divisão univariável, ou seja, cada nó interno da árvore é dividido de acordo com um único atributo. Nesse caso, o algoritmo tenta encontrar o melhor atributo para realizar essa divisão.

Os critérios de seleção para a melhor divisão são baseados em diferentes medidas, tais como impureza, distância e dependência. A maior parte dos algoritmos de indução busca dividir os dados de um nó-pai de forma a minimizar o grau de impureza dos nós-filhos. Quanto menor o grau de impureza, mais desbalanceada é a distribuição de classes. Em um determinado nó, a impureza é nula se todos os exemplos nele pertencerem à mesma classe. Analogamente, o grau de impureza é máximo no nó se houver o mesmo número de exemplos para cada classe possível.

4.5 CONSTRUÇÃO DE UMA ÁRVORE DE DECISÃO

Árvores de decisão são exemplos simbólicos de mineração de dados cuja estrutura apresenta-se no formato de uma árvore. Cada nó interno da árvore indica um teste sobre um atributo, cada ramo representa um resultado do teste, e os nós terminais (folhas) correspondem a classes ou distribuições de classes na sua estrutura. A profundidade de uma árvore é definida pela maior distância entre uma folha e a raiz (primeiro nó). Com isso, tem-se uma técnica que constrói regras de classificação passíveis de avaliação, interpretação e posterior aplicação. Em função destas características, as árvores de decisão tornam-se interessantes para os problemas de reconhecimento de padrões.

Com já explanado anteriormente, algumas das vantagens apresentadas pelas árvores de decisão são sua flexibilidade, pois não assumem uma distribuição única dos dados, sendo métodos não-paramétricos; robustez, uma vez a seleção interna de características produz árvores que tendem a ser bastante robustas mesmo com a adição de variáveis irrelevantes; interpretabilidade, já que todas as decisões são baseadas nos valores (conhecidos) dos atributos usados para descrever o problema; e velocidade, pois a maioria dos algoritmos constrói rapidamente as árvores de decisão (GAMA, 1999).

Geralmente, o procedimento de uma árvore de decisão consiste em apresentar um conjunto de dados ao nó raiz da árvore e avaliá-lo segundo um teste lógico. Dependendo do resultado, a árvore ramifica-se para um dos nós descendentes e este procedimento é repetido até que uma folha conceda a classificação dos dados. A Figura 14, logo abaixo, exemplifica a estrutura de uma árvore de decisão de resposta binária.

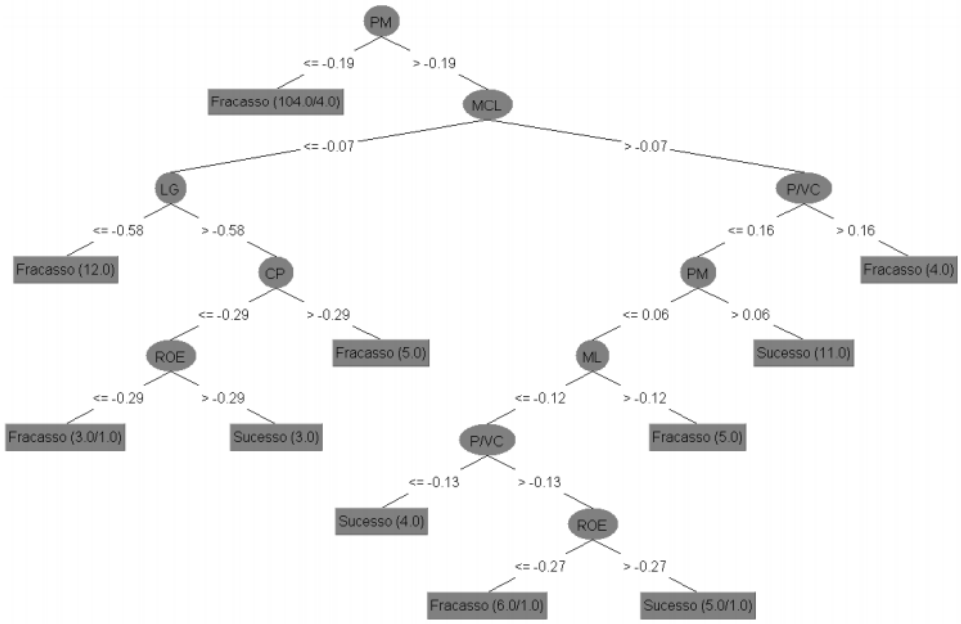


Figura 14 - Exemplo de árvore de decisão com resposta binária. Fonte: Gama (1999).

O aprendizado de uma árvore de decisão é normalmente supervisionado, ou seja, o método aproxima funções-alvo de valor discreto, na qual a função aprendida é representada por uma árvore de decisão. As árvores treinadas podem ser representadas como um conjunto de regras *if-then* para melhor compreensão e interpretação.

As árvores de decisão são construídas usando um algoritmo de partição recursiva. Uma das possibilidades é este algoritmo construir uma árvore por divisões recursivas binárias que começa no nó raiz e desce até os nós folhas. Neste caso, têm-se dois fatores principais no algoritmo de partição: a forma de selecionar uma divisão para cada nó intermediário (crescimento) e uma regra para determinar quando um nó é terminal (poda).

Grande parte dos algoritmos de construção (também chamado de indução) de árvores de decisão corresponde a um procedimento “guloso” (técnica para problemas de otimização que sempre faz uma escolha com a intenção de atingir uma solução ótima global) que recursivamente constrói a árvore do nó raiz em direção aos nós terminais. Em cada iteração, a partir da base de dados de treinamento, os algoritmos procuram pelo atributo que melhor

separa as classes para realizarem a ramificação da árvore, e recursivamente processam os subproblemas resultantes das ramificações.

O aprendizado indutivo de árvores de decisão pode ser dividido em aprendizado supervisionado e não-supervisionado, embora o aprendizado semi-supervisionado também tem sido referenciado ao longo dos últimos anos (CHAPELLE et al., 2006).

• Critério para a escolha da característica a utilizar em cada nó;

• Como calcular a partição do conjunto de exemplos;

• Quando decidir que um nó é uma folha;

• Qual o critério para selecionar a classe a atribuir a cada folha;

Algumas vantagens podem ser apontadas em relação as árvores de decisão, nomeadamente:

• Podem ser aplicadas a qualquer tipo de dados;

• A estrutura final do classificador é bastante simples podendo ser guardada e manipulada de uma forma bastante eficiente;

• Manipula de uma forma muito eficiente informação condicional subdividindo o espaço em subespaços que são manipulados individualmente;

• Revelam-se normalmente robustos e insensíveis a erros de classificação no conjunto de treino;

• As árvores resultantes são normalmente bastante compreensíveis podendo ser facilmente utilizadas para se obter uma melhor compreensão do fenômeno em causa. Esta é talvez a mais importante de todas as vantagens enunciadas;

Na Figura 15 é ilustrado um diagrama do processo de indução de um classificador e, posteriormente, a sua utilização. Primeiro, o conjunto de treinamento, no qual os rótulos das classes dos exemplos são conhecidos, é utilizado por um algoritmo de aprendizado para construir um modelo. Após a construção, esse classificador pode ser aplicado para predizer os rótulos das classes dos exemplos do conjunto de teste, ou seja, exemplos cujas classes são desconhecidas.

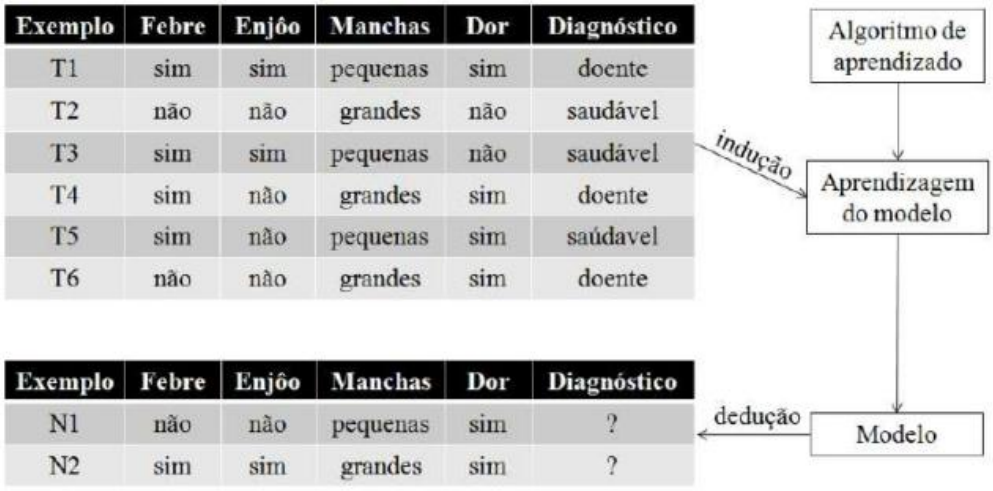


Figura 15 – Diagrama do processo de indução de um classificador. Fonte: CHAPELLE (2006).

Na construção da árvore de decisão, procura-se associar a cada nó de decisão o atributo "mais informativo" entre aqueles ainda não utilizados no caminho desde a raiz da árvore. No entanto, cada algoritmo tem a sua própria metodologia para distinguir o atributo mais informativo, fazendo com que a topologia da árvore e a qualidade da árvore variem em função do algoritmo utilizado.

Um dos aspectos negativos que existe na construção da árvore de decisão é o risco de crescimento exponencial. O processo de crescimento da árvore ocorre até não ser possível criar mais sub árvores, ou, alternativamente, até atingir uma condição de paragem previamente definida. Normalmente são definidas condições de paragem, estas condições podem ser a profundidade da árvore ou a significância mínima que um novo corte que cria uma nova sub árvore implicará no modelo preditivo. A definição de condições de paragem do processo de crescimento de uma árvore é um aspecto crucial para o bom desempenho do processo, pois caso não seja definido nenhuma condição, a árvore iria crescer até ao máximo possível, originando uma "árvore maximal" que contemplaria todas as observações do *data-set* de treino, ou seja, para além de ser seria demasiado moldada para os dados de treino, originando assim um problema grave de *overfitting*.

Pruning é o processo que consiste na simplificação da árvore de decisão gerada previamente, esta simplificação é atingida pelo corte sucessivo dos nodos que representam uma relevância preditiva pouco acentuada. O *pruning* permite ao mesmo tempo diminuir o tamanho e a complexidade das árvores de decisão, diminuindo o risco de *overfitting*. Como mencionado anteriormente, a decisão de paragem do crescimento de uma árvore é um assunto a que se deve dar elevada importância, no entanto é impossível determinar quando se deve parar o processo de crescimento pois é impossível calcular se a adição de um nodo extra irá

melhorar drasticamente a capacidade preditiva da árvore. Por estas razões, é frequentemente realizado o crescimento da árvore até um determinado tamanho que implique a inclusão de um pequeno número de casos em cada nodo terminal, e posteriormente é realizado o *pruning*, de forma a remover os nodos que não representam uma mais-valia para o modelo. O *pruning* pode ser feito mediante diferentes algoritmos, sendo o objetivo, atingir a melhor relação possível entre a simplicidade e o poder de previsão da árvore, o que não é trivial.

4.6 ALGORITMO PARA A INDUÇÃO DA ÁRVORE

O algoritmo para a construção de uma árvore de decisão seleciona um atributo de particionamento e divide o conjunto de exemplos, criando um ramo para cada valor deste atributo. A cada ramo é criado um nodo e novamente selecionado um novo atributo de particionamento para o subconjunto de exemplos atribuído ao novo nodo e, assim, sucessivamente. Este processo tem, como objetivo, separar os exemplos em classes, de modo que os exemplos de distintas classes tendam a serem atribuídos a partições diferentes.

Essa abordagem de divisão e conquistas adotada pelos algoritmos de indução de árvores de decisão foi desenvolvida e refinada ao longo de vários anos por John Ross Quinlan. Sua contribuição inicial foi o algoritmo ID3 (QUINLAN, 1986). Várias melhorias foram realizadas nesse algoritmo, culminando no surgimento de algoritmo C4.5 (QUINLAN, 1993), muito utilizado em aplicações práticas e pesquisas acadêmicas.

Em ambos, a escolha do atributo que geram as ramificações é feita a partir de uma medida conhecida como Ganho da Informação, que nada mais é do que a informação mútua entre a variável escolhida e a variável resposta condicionada às escolhas feitas na iteração i. O atributo que proporcionar maior ganho de informação é selecionado como atributo teste do nó corrente, em outras palavras, é escolhida a variável que promove a maior minimização da entropia.

O algoritmo 3.1 mostra um algoritmo genérico para construir árvores de decisão, em que S representa o conjunto de exemplos aplicado à árvore, sendo que inicialmente S contém todos os exemplos de treino. Para melhor ilustração, a Figura 16 apresenta o algoritmo genérico para a construção de árvores.

(1) *Se* todos os exemplos no atual conjunto de exemplos S satisfazem um critério de parada
então
(2) cria um nodo folha com algum nome da classe e pára;
senão
(3) seleciona um atributo A para ser utilizado como um atributo de particionamento e cria um nodo com o nome do atributo de particionamento;
(4) escolhe um teste sobre os valores de A, com resultados mutualmente exclusivos e coletivamente exaustivos $R_1,...,R_k$, e cria um ramo, a partir do nodo recentemente criado, para cada teste;
(5) particiona S nos subconjuntos $S_1,...S_k$, tal que cada S_i, i=1..,k, contenha todos os exemplos em S com resultado R_i do teste escolhido;
(6) aplica este algoritmo recursivamente para cada subconjunto S_i, i=1,...k;
fim_senão
fim_se

Figura 16 – Algoritmo genérico para construção de árvores de decisão. Fonte: o autor.

Cada passo do algoritmo é identificado por um número sequencial entre parênteses à sua esquerda. Os passos (1) e do (3) são passos mais relevantes, por requererem acesso aos dados para avaliar uma regra candidata. Estes são os passos que consomem mais tempo do algoritmo quando a descoberta de conhecimento se dá em grandes bases de dados.

O passo (1) do algoritmo 3.1 consiste em decidir o momento de parar o particionamento recursivo. O processo é parado se todos os exemplos no nodo atual possuem a mesma classe. No entanto, se esta condição não for verdadeira, pode ser interessante para o particionamento no atual nodo, para evitar que a árvore se expanda muito. Esta forma de parar o particionamento é conhecida como pré-poda e é discutida com mais detalhes a seguir.

No passo (2) se todos os exemplos no recém-criado nodo folha têm a mesma classe, o algoritmo dá ao nodo o nome da classe; contudo, o algoritmo pode dar ao nodo folha o nome da classe mais frequente ocorrida neste nodo.

No passo (3) é computado o melhor atributo candidato à partição, avaliando como, efetivamente, os valores do atributo candidato discriminam as classes dos exemplos.

No passo (4) é criado um arco para cada valor distinto, resultante do particionamento do atributo selecionado no passo (3).

O passo (5) consiste em designar cada exemplo a um dos arcos criados, de acordo com o valor do atributo de particionamento.

Por último, no passo (6), o algoritmo é aplicado recursivamente para cada subconjunto do conjunto de exemplos S.

Neste capítulo foi demonstrado a aplicabilidade a importância dos classificadores, da Descoberta de Conhecimento, e consequentemente, das técnicas de Mineração de Dados.

4.7 WEKA (Waikato Environment for Knowledge Analysis)

O Weka é uma ferramenta de mineração de dados com vasta coleção de algoritmos de aprendizado em máquina e ferramentas de pré-processamento de dados. Foi projetado com intuito de que seus métodos sejam executados em conjunto de dados na forma mais flexível possível. Inclui dezenas de ferramentas com interface fácil e de rápida comparação entre seus métodos, com facilidade na geração e comparação dos modelos mais adequados para a solução do problema. No algoritmo J48, é possível plotar a árvore gerada com suas regras, o que foi de grande ajuda para validação dos modelos junto a um especialista da área.

Seu desenvolvimento pela Universidade de Waikato, na Nova Zelândia, seguiu os termos da GNU (*General Public License*), com implementação da linguagem Java, o que facilita a implementação de softwares de terceiros nesta linguagem utilizando suas bibliotecas.

A aprendizagem de máquina é realizada com a aplicação de algoritmos e técnicas que permitem ao computador aperfeiçoar seu desempenho na execução de tarefas, como, por exemplo, a busca de padrões em um conjunto de dados de treinamento.

O WEKA possui, basicamente, quatro sistemas de aprendizagem de máquina:

a) Aprendizagem supervisionada (classificação): onde a aprendizagem é realizada a partir de um conjunto de dados amostrais, denominado conjunto de dados de treinamento, podendo ser aplicada em novos conjuntos de dados;
b) Aprendizagem por associação: busca, em um conjunto de dados, qualquer associação entre as características de um outro conjunto de dados amostrais;
c) Agrupamentos (clusters): grupos de dados similares são procurados em um conjunto;
d) Precisão numérica: o resultado a ser previsto não é uma classe discreta, mas uma quantidade numérica.

Os algoritmos que trabalham com a aprendizagem supervisionada são preditivos, e suas pesquisas, no conjunto de dados de amostra, desempenham inferências nos dados com o intuito de fornecer previsões ou tendências, obtendo informações não disponíveis a partir dos dados disponíveis. A forma mais comum de representação destes algoritmos são as regras de indução por árvores de decisão (SILVA, 2004).

Muitos são os algoritmos de classificação que utilizam a representação sob o formato de Árvore de Decisão (WITTEN; FRANK, 2005). No WEKA, o algoritmo J48, proposto por Quinlan (1993), é considerado o algoritmo que representa o melhor resultado na montagem de árvores de decisão a partir de um conjunto de dados de treinamento. Para a montagem da árvore, o algoritmo J48 utiliza a abordagem de dividir-para-conquistar, onde um problema complexo é decomposto em subproblemas mais simples, aplicando recursivamente a mesma estratégia a cada subproblema, dividindo o espaço definido pelos atributos em subespaços, associando-se a eles uma classe (WITTEN; FRANK, 2005).

A abordagem dividir-para-conquistar tem início com um conjunto de dados de treinamento (amostras) e cria uma estrutura em forma de árvore contendo as regras com os atributos que definem as classes. Se os atributos são suficientes, é sempre possível construir uma árvore de decisão que classifica corretamente cada objeto do conjunto de treinamento e estas regras podem ser aplicadas na classificação de novos conjuntos de dados (WITTEN; FRANK, 2005).

Conforme já apresentado na sessão 3.4, o processo de aprendizagem e montagem da estrutura de uma árvore de decisão é conhecido como indução de árvores de decisão. A indução busca padrões em informações disponíveis com o propósito de inferir conclusões racionais sobre a informação. A tarefa de indução é o processo de desenvolver regras de classificação que podem determinar a classe de um objeto através de seus atributos. Os objetos são descritos em termos de uma coleção de atributos.

Uma árvore pode ser induzida pela sucessiva seleção e subdivisão dos atributos. O modo de seleção de um atributo é determinado pela verificação de cada atributo, avaliando-se a qualidade ou representatividade do mesmo, de forma a aumentar o desempenho de decisão da árvore. A escolha de atributos representativos é realizada por meio de funções de avaliação. O conceito básico da avaliação no processo de divisão é selecionar um atributo que produza a melhor árvore. Porém, a função de avaliação prediz somente um único nó. Desde modo, a função avaliação precisa descobrir o caminho na árvore que realiza boas decisões, mesmo com informações incompletas. A função de avaliação mais utilizada para a escolha do atributo trabalha pela redução do grau de incerteza no nó corrente (PAULA, 2002).

4.7.1 TREINAMENTO E EXECUÇÃO DO ALGORITMO DE MINERAÇÃO DE DADOS

Como já apresentado neste capítulo, árvore de decisão é uma forma simples de classificar amostras em um número finito de classes. Elas consistem na determinação de regras para divisão de um conjunto de dados em grupos, de forma que uma determinada amostra seja caracterizada pelo comportamento do grupo onde está inserida. Essas regras variam de acordo com o algoritmo escolhido, sendo essa escolha dependente do objetivo da análise. Neste trabalho foi utilizado o algoritmo J48 disponível no software WEKA versão 3.6.12.

Esse algoritmo é uma implementação de código aberto do algoritmo C.45 para o software WEKA. Ele gera o modelo de árvore de decisão a partir da análise de dados de treinamento construindo uma árvore do topo (nó raiz) para a base (folhas), onde a variável definida como o topo representa o elemento mais integrador da análise, portanto, a variável definida como o topo representa o elemento mais integrador da análise, assim sendo, a variável mais significativa da classificação. O funcionamento do algoritmo é semelhante ao de um fluxograma em forma de árvore criando sub-árvores até chegar às folhas (categoria final da classificação), o que implica numa sequência hierárquica de divisões. Após a árvore de decisão montada, para classificar uma nova amostra, basta seguir o fluxo na árvore começando do nó raiz até chegar a uma folha.

O algoritmo J48 utiliza como função de avaliação o conceito de entropia da informação, que de acordo com Shannon e Weaver (1949), é a medida da incerteza e do seu complemento associada a uma variável aleatória.

No WEKA a extração do conhecimento, obtido através da aprendizagem da árvore de decisão gerada a partir do conjunto de dados de treinamento com a finalidade específica de classificar objetos utilizando o algoritmo J48, segue os seguintes passos: (a) leitura do arquivo ARFF, com apresentação quantitativa e estatística dos atributos e classes existentes no conjunto de dados de treinamento; (b) escolha do método de classificação, no caso deste trabalho, a árvore de decisão; (c) validação dos dados com apresentação do número total de instâncias (exemplos), percentual de instâncias classificadas corretamente e incorretamente, valor do kappa estatístico, erro médio absoluto, erro médio quadrático, percentual relativo do erro absoluto, detalhes da precisão por classe, o peso médio encontrado e a matriz de confusão das classes; (d) montagem da árvore de decisão; e (e) obtenção do conhecimento por meio da

exportação do arquivo texto, denominado IF-THEN, que contêm as regras de decisão da aprendizagem da classificação do WEKA.

O algoritmo J48 no WEKA produz uma árvore de decisão a partir de atributos numéricos e dados incompletos, gerando regras de decisão. Ele trabalha com a produção com conjunto de regras denominado de “lista de decisão” que é usada na formação de árvores de decisão. O algoritmo programa a sequência mínima de otimização (SMO) utilizada na aprendizagem de máquina para extrair a máxima quantidade de informação de um conjunto de exemplos dada à condição de que somente um atributo será utilizado para realização do teste (EIBE et.al., 1999).

O algoritmo J48 aprende com os dados com elevada acuidade, entretanto para a solução de problemas mais complexos, com a mesma acuidade, é necessária uma quantidade considerável de dados (Souza, 1998).

A utilização de árvores de decisão como técnica de mineração dados apresenta vantagens sobre outras técnicas que também buscam a classificação dos dados. Souza (1998) indica que na árvore de decisão o tempo de elaboração do modelo é consideravelmente menor apresentando acuidade similar a outras técnicas, além disso, ela permite a aplicação em grandes conjuntos de dados, possibilitando uma visão real da natureza do processo de divisão.

5 TRABALHOS RELACIONADOS

Foi realizada uma pesquisa bibliográfica de trabalhos relacionados a fim de compreender com mais propriedade sobre como utilizar Folksonomia no contexto de filtragem de conteúdo. Este capítulo, portanto, apresenta quatro trabalhos encontrados. Dois (2) destes trabalhos com enfoque em "classificação e filtragem de conteúdo", enquanto os outros dois (2) abordam "Folksomonia".

5.1 O USO DE FILTROS DE CONTROLE NA INTERNET: O CASO DO SISTEMA EDUCACIONAL

Barros, Bittencourt, Pimentel (2010) apresentam um estudo de caso, usando uma pesquisa qualitativa, trazendo uma análise de dados coletados na Universidade Federal de Alagoas junto aos alunos de vários cursos e também de professores, das áreas das Tecnologias da Informação e Comunicação.

O avanço da Internet nos meios sociais e educacionais têm gerado uma gama de estudos e pesquisa sobre as possibilidades e os desafios que o uso das mídias e tecnologias pode promover. No meio educacional um embate é realizado no sentido de liberar ou não a navegação em sites de relacionamento, blogs, microblogs ou em chat. Na maioria das escolas e até mesmo nas universidades, o comum é o uso de "filtros" que impedem a navegação, restringindo o uso dos navegantes.

Barros, Bittencourt, Pimentel (2010), tendo como base os pressupostos teóricos que fundamentam o uso das TIC na educação, analisam em uma pesquisa qualitativa, a partir de um estudo de caso descritivo, o uso de mecanismos para o controle de acesso a conteúdo na Internet, suas contribuições e possíveis consequências para aprendizagem dos alunos. Os autores desenvolveram ambientes virtuais para a coleta de dados que foram visitados por alunos e professores. O que motivou o desenvolvimento do trabalho foram as dificuldades que a maioria dos alunos apresentaram ao tentar usar alguns sites, inclusive sites de relacionamento, como possibilidade para sua aprendizagem. Outros autores como, Mercado (2009), Moran (2000) Almeida (2009) e Silva (2003) apontam para a integração das mídias digitais nos espaços de aprendizagem, indicando possibilidades pedagógicas com o uso de suas interfaces. Desta maneira, as TICs na educação podem ser pensadas como recursos que potencializam as dimensões do processo de ensino aprendizagem, tendo em vista que estamos

inseridos no contexto da Cibercultura (LÉVY, 1999) e que os alunos do século XXI, nasceram meio a esta avalanche de tecnologias (PRENSKY, 2001).

Como é possível observar nos dias de hoje, a utilização de filtros de controle para acesso à Internet tem se tornado uma ferramenta necessária no meio acadêmico e corporativo. Com a instalação destas ferramentas é possível gerenciar e/ou monitorar o uso da Internet no ambiente, bloqueando o acesso a sites e aplicativos considerados inúteis, não produtivos e até mesmo perigosos. Grande parte das instituições e empresas adotam políticas de bloqueio à Internet que podem incorporar filtros IP, bloqueio de URLs por palavras-chave e bloqueio de domínios. Sítios de chat, comunicadores instantâneos, redes sociais, blogs, palavras-chave que sugerem conteúdo sexual, humor, mídias são bloqueados com o argumento de que elas não condizem com um ambiente produtivo de trabalho. Segundo os autores, a eficiência do bloqueio de conteúdos em algumas empresas é muito mais rígida do que qualquer política de censura à Internet de vários países.

Visando melhorar a comunicação de pessoas, autores como Prado (2001), Mercado (2005) e Cruz (2006) incentivam o uso de Chat no ambiente educacional. Mercado (2008) defende o uso de ferramentas online como parte do processo de ensino aprendizagem, para captar conhecimentos e aprofundar conteúdos que tenham sido trabalhados em sala de aula.

Barros, Bittencourt, Pimentel (2010) criaram uma comunidade autônoma e aberta em um site de relacionamento para tratar de assuntos de interesse daqueles que se utilizam dos serviços e benefícios que a universidade oferece. E como a rede social escolhida possibilita a criação de enquetes e fóruns de discussão, foi criado um fórum para discutir a utilização de mecanismos de bloqueio na Internet interna da UFAL (Universidade Federal de Alagoas). A enquete do fórum criada foi “O que você acha da utilização pela universidade de mecanismos como filtros de conteúdo para impedir o acesso a sites de relacionamento, sites de vídeos, etc.?”.

Os autores concluem que no tocante a visão dos alunos, como foco da pesquisa, alguns alunos sentem-se cerceados de empreender seus projetos nos espaços públicos de acesso à Internet. Outros desafiam o sistema e constantemente buscam mecanismos de burlar este controle. Ainda outros argumentam que isso é a maneira de cercear e manter o controle social privando alunos de alguns conteúdos que poderiam empreender projetos revolucionários, e isto caminha de encontro à ordem social.

As atuais instituições de ensino têm um bom arsenal de aparatos tecnológicos. A concepção acerca da escola do futuro é aquela que oferece acesso as tecnologias, informática, aulas à distância, exposição de aulas em slides com projetores multimídia, etc. Entretanto, nos

laboratórios de informática dessas instituições vigoram regras de uso da rede, com mais proibições do que possibilidades. São comuns as seguintes restrições: acesso e uso de determinados correios eletrônicos, sites de mensagens instantâneas, sites de relacionamento, redes sociais, sites especializados em downloads, sites de jogos, sites de armazenamento de vídeos, sites de armazenamento de músicas e seus *streamings*, etc. Enfim, o uso dos computadores é estritamente regulamentado nestes laboratórios e, muitas vezes, na própria imagem de fundo da área de trabalho do sistema operacional, dizeres advertindo as punições para aqueles que estão dispostos a descumprir tais regras.

Os autores consideram que uma educação, que utilize tais meios, propicia, tanto para professores, como para alunos, elementos pedagógicos indispensáveis para que sejam utilizados a favorecer o ensino-aprendizagem. Na visão da maioria dos alunos que participaram da pesquisa criada pelos autores, os sites de relacionamento, podem ser utilizados, para desenvolver alguma atividade de estudo. Já para a maioria dos professores ainda é cedo para que tudo seja liberado, mas que uma nova educação deve incluir o bom uso das TICs e o uso de filtros ainda é um recurso que precisa ser utilizado enquanto a sociedade se adapta a cultura do online.

5.2 PROTÓTIPO SISBLOQUE: TÉCNICA DE FILTRAGEM E BLOQUEIO DE CONTEÚDO WEB

Os tempos modernos requerem maiores cuidados com os serviços e dados trafegados pela Internet. A criminalidade digital tem evoluído e como consequência gerando grandes prejuízos tanto financeiros, morais e materiais aos usuários. Este cenário tem motivado o crescimento de novas ferramentas de segurança de acesso à Internet. Os sistemas de bloqueio e filtragem de conteúdo *Web* encontram-se, em sua maioria, associados a regimes políticos opressivos, cujo principal objetivo na sua utilização é a censura. Infelizmente, este tipo de aplicação não só limita os direitos, como revoga o princípio daquilo que é uma rede global de partilha de informação pública, a Internet. Não obstante a filtragem e bloqueio de conteúdos *Web* é uma área em proliferação, cuja sua correta utilização tem vindo a demonstrar-se extremamente benéfica em determinadas áreas como a detecção e bloqueio de conteúdos relacionados a pedofilia.

O artigo apresenta o protótipo Sisbloque, um sistema de filtragem e bloqueio de conteúdos *Web* projetado para ser implementado em ISPs (*Internet Service Providers*) e grandes companhias. Ele propõe não só um mecanismo de filtragem de conteúdo com novas

técnicas aperfeiçoadas e inovadoras, como a intersecção de conteúdos relacionados, bem como a garantia de execução transparente suportada por um mecanismo de manipulação de erros.

A Internet no mundo tem evoluído em escalas significativas nos últimos anos, tanto do ponto de vista de novos usuários, como também das novas tecnologias que diariamente são introduzidas. Organizações e instituições disponibilizam estes recursos tecnológicos como ferramentas de apoio aos seus clientes / usuários, com objetivo de viabilizar novos sistemas e serviços. Atualmente, além destas entidades, a criminalidade também tem usado de forma consistente as novas tecnologias e produzido métodos mais sofisticados para lesar usuários. Em decorrência deste fato, a Internet torna-se num ambiente inseguro para o compartilhamento de informação, ou ainda para qualquer outra atividade.

Contudo, a correta utilização de sistemas de filtragem e bloqueio de conteúdos *Web* apresenta-se como uma solução eficaz, não só em face de problemas como a publicação on-line de conteúdos pedófilos, como também na restrição de acesso a conteúdos improdutivos por parte dos colaboradores em uma organização. A maioria dos sistemas de filtragem e bloqueio de conteúdos *Web* são soluções proprietárias ou produtos comerciais, afastando o conhecimento da comunidade científica da maioria dos detalhes de implementação destes sistemas. Tais constatações motivaram os autores para o desenvolvimento e concepção de um sistema aberto de filtragem e bloqueio de conteúdos *Web*, designado Sisbloque. Este sistema encontra-se em fase de desenvolvimento e a sua concepção tem como principal objetivo o potencial uso em ambientes ISPs, grandes companhias ou instituições que necessitem deste tipo de serviço.

O sistema Sisbloque propõe um mecanismo de filtragem de conteúdo mais conciso e confiável, proveniente do melhoramento de métodos existentes como a filtragem baseada na origem e introdução de novos conceitos nos métodos de avaliação de conteúdo. Os recursos utilizados para o desenvolvimento do Sisbloque consistem em softwares de código aberto como um servidor *HTTP* de tecnologia Apache [APACHE 2015] que faculta ao Sisbloque a capacidade de combinar determinados parâmetros do protocolo *HTTP* durante a atividade de bloqueio do tráfego *HTTP*. A gestão de acessos a endereços *URL*/IP é implementada com base na integração do Web proxy Squid [SQUID 2015] e do respectivo *plugin* SquidGuard [SQYUDGUARD 2015]. Todo este conjunto de componentes trabalha sobre o sistema operacional Linux, mais especificamente a distribuição Fedora Core 8 [FEDORA 2015] de Kernel 2.6.25.9, a qual confere suporte a um *framework* de filtragem de pacotes designado Netfilter [NETFILTER, 2015].

Conforme apresentado pelos autores, o sistema Sisbloque possui dois mecanismos de filtragem, o filtro transparente e filtro de conteúdo. Estes dois filtros são, na realidade, dois módulos físicos distintos que contêm técnicas de filtragem diferentes e adequadas à sua função e posição física na topologia de rede onde o sistema se encontra em atividade.

Para evitar a latência no acesso ao conteúdo *Web*, o filtro transparente é composto por métodos de filtragem de conteúdo, que não requerem tempo de processamento expressivo. Estes métodos consistem na filtragem de endereços *URL* do protocolo *HTTP* e de endereços IP contidos nos pacotes de dados. Após esta filtragem, os endereços *URL*/IP são comparados com endereços *URL*/IP contidos nas listas de inclusão e exclusão do sistema. Caso se confirme veracidade na comparação do endereço com um dos endereços contidos na lista de exclusão, a respectiva ligação ao servidor de conteúdo é interrompida e redirecionada para o mecanismo de manipulação de erros. Já o filtro de conteúdo é composto por duas técnicas que abordam aspectos diferentes daquela que se encontra implementada no filtro previamente descrito, estas técnicas designam-se de filtragem por avaliação e intersecção de conteúdos relativos.

A filtragem por avaliação provém de melhoria na existente técnica de filtragem por palavras [GREENFIELD et al. 2001]. Deste modo o conteúdo de um endereço *URL*/IP não é avaliado pelas palavras que possui, mas sim pelo valor resultante da soma de todas as cotações atribuídas a cada palavra. Desta forma, a soma das cotações pode ser positiva ou negativa, caso a palavra seja ou não considerada ofensiva. Após avaliação do conteúdo referente ao endereço *URL*/IP, é realizada a comparação entre o resultado da soma das cotações obtido e o valor predefinido pelo administrador do sistema, o qual é o limiar do que é ou não considerado conteúdo malicioso. Por conseguinte, o endereço *URL*/IP do conteúdo avaliado é ou não adicionado às listas de exclusão do filtro transparente. Os endereços *URL*/IP que são avaliados por esta técnica provêm do filtro transparente, e consistem em todos os endereços desconhecidos ao sistema, ou seja, são endereços que não estão cadastrados nas listas de exclusão, nem na lista de inclusão.

Conforme descreve Pires, Fonte e Soares [2009] o sistema Sisbloque é constituído basicamente por três módulos físicos distintos: o módulo de filtro transparente, o módulo do filtro de conteúdo e o módulo de mecanismo de manipulação de erros. Os módulos dos filtros têm a principal função avaliar os endereços e conteúdos solicitados pelos usuários, já o módulo do mecanismo de manipulação de erros efetua a geração de códigos de erro aleatórios da gama 5xx do protocolo *HTTP*. Para tal efeito é utilizado neste módulo o servidor *HTTP* Apache.

Os autores chegam a conclusão que é possível desenvolver um sistema com uma arquitetura distribuída e utilizando sistemas de código aberto, altamente eficaz e de baixo custo de implementação e manutenção. Alguns pontos são importantes e merecem destaque, como as técnicas utilizadas que possuem conceito inovador no que diz respeito a filtragem e bloqueio de conteúdo. Os dois artigos comentados ilustram a necessidade de elaboração de sistemas de filtragem de conteúdos eficientes, a fim de evitar a má utilização da Internet em determinados contextos.

5.3 ESTUDO SOBRE A DOAÇÃO DA FOLKSONOMIA EM SISTEMA DE INFORMAÇÃO: UMA PROPOSTA DE HIBRIDISMO

O trabalho de Silva e Miranda (2013) consiste em um estudo sobre a aplicação da Folksonomia em sistemas de informação. Estudos mostram que a necessidade de organizar o que está a sua volta é uma ação humana que ocorre há muito tempo. Para isto, a representação e classificação são métodos de organização. Diversos debates são realizados para definição dos pontos que precisam ser considerados na implementação de Folksonomia em Sistema de Informação (SI). Trata da mudança de paradigma da *Web* 1.0, ambiente rígido da Taxonomia, para a *Web* 2.0, ambiente flexível da Folksonomia.

O objetivo do trabalho de Silva e Miranda (2013) é buscar por estruturas híbridas (Taxonomia e Folksonomia), formalizada em uma pesquisa bibliográfica e embasada em periódicos científicos, bibliografia especializada e websites que fazem uso destas formas de representação, e consequentemente analisar questões para utilização da Folksonomia em SI, como também apresentar modelos de sistemas na *Web* que utilizam o hibridismo e apresentar uma proposta de hibridismo para SI. O trabalho elenca como pontos de análise: a flexibilidade, identificação de padrões, colaboração social e a forma anárquica resultante da marcação da *tag*. Os autores apresentam como resultado da pesquisa uma estrutura de representação e organização da informação a partir dos modelos de hibridismo. O website de e-commerce da Livraria Cultura é um exemplo, pois faz uso concomitante de uma nuvem de *tags* com sua Taxonomia, a identificação de padrões adotada pelo website de e-commerce Americanas e o modelo de Folksonomia influenciado a Taxonomia pelo website de e-commerce Submarino.

Os autores apresentam o resultado de uma pesquisa realizada em 2010 para a compreensão da Folksonomia quanto à determinação de quatro questões mínimas que devem ser consideradas para a sua implementação, são elas: a “flexibilidade”, a “identificação de

padrões", a "colaboração social" e a "forma anárquica que nasce a informação". Destaca-se também a possibilidade de geração de problemas na sua forma de classificar a informação, além de fazer com que os sistemas considerados rígidos (Taxonomia) ganhem novo valor. Neste contexto específico de "anarquia" na marcação das *tags*, cinco pontos devem ser analisados: plurais, polissemia, sinonímia, palavras consideradas egoístas e palavras escritas de modo incorreto.

Conforme descreve Silva e Miranda (2013): "foram consideradas a "flexibilidade", sendo esta considerada a vantagem mais clara do uso da Folksonomia; a "identificação de padrões" já que possibilita a identificação de padrões de organização, conceitos compartilhados e de colaboração entre as pessoas; a "colaboração social", considerada a mais relevante para o uso da Folksonomia, devido à possibilidade de colaboração em torno da captura e organização da informação; e a "forma anárquica que nasce a marcação da *tag*", pontuada nas características plurais, polissemia, sinonímia, palavras consideradas egoístas e escrita incorreta.

Conforme apresentado no trabalho de Silva e Miranda (2013), foram considerados os pontos de análise supracitados para a implementação da Folksonomia no SI da empresa Design. Como inspiração para a implementação a empresa Design baseou-se nos websites de e-commerce da Livraria Cultura, com o "uso concomitante de uma nuvem de *tags* com sua Taxonomia"; a "identificação de padrões" adotada pelo website de *e-commerce* Americanas; e o modelo de Folksonomia influenciado a Taxonomia pelo *website* de *e-commerce* Submarino. Diante das características apresentadas, acredita-se que a adoção de ambientes híbridos permitirá à equipe usufruir da interatividade proporcionada pela *Web* 2.0, como também navegar em um ambiente minimamente organizado, assim como prevê a *Web* 1.0.

A Folksonomia pode ser um benefício ao SI que busca a participação coletiva, reforçando que a sua ideologia não deve ser mais vista como caos da atual Internet. Dessa forma, visualiza-se, neste contexto, que a citação da estrutura de representação e organização da informação, estará em constante mudança conforme o "tagueamento" dos usuários, sendo necessário atenção nestas mudanças pelos profissionais da informação.

Conclui-se que o atual momento da Internet está delineado pela interação e, consequentemente, os sistemas precisam acompanhar essa realidade, mudando conforme as ações de seus usuários. O hibridismo pode oferecer este acompanhamento sem perder a flexibilidade e a interatividade, como também proporcionar um ambiente taxonômico, respeitando o "tagueamento" realizado.

5.4 GERENCIAMENTO DO VOCABULÁRIO DO USUÁRIO EM SISTEMAS BASEADOS EM *TAGGING*

Ao falar do trabalho de Silva (2009) devemos considerar a fundamentação teórica no qual descreve ao aumento do uso de rótulos de texto para categorizar informação na Web a partir de 2002. Nesse período, grande parte dos novos websites, permitiram que seus usuários utilizassem palavras simples (etiquetas, ou *tags*) na forma de um texto não-estruturado em um processo denominado *tagging*. Nos dias de hoje, diversos sistemas baseados em *tagging*, cada um centrado em um tipo de objeto ou recurso, obriga os usuários a manterem seus vocabulários espalhados por diversos sistemas que adotam essa abordagem, o que prejudica a manutenção de um vocabulário de termos consistente e dificulta os processos de categorização e de recuperação de informação. Por essa razão, o trabalho do autor tem como principal objetivo desenvolver um repositório central e único para o vocabulário de *tags* que o usuário utiliza para categorizar e recuperar conteúdos em sistemas baseado em *tagging*.

O gerenciador de *tags* auxiliará o usuário na organização e no uso do seu vocabulário pessoal, reduzindo os problemas e a complexibilidade em lidar com diversos sistemas e vocabulários simultaneamente. A intenção é melhorar a qualidade das categorizações e, consequentemente, da recuperação de informação em sistemas baseados em *tagging*. O trabalho é exposto às discussões teóricas e práticas sobre a ferramenta de gerenciamento de *tags* proposta, a arquitetura desse sistema e exemplos de testes que indicam a sua viabilidade.

Para ilustrar a estrutura necessária do sistema desenvolvido por Silva (2009) para o gerenciamento de *tags*, abaixo está a Figura 17. A interface *Web* (visão de dados pelo usuário) para acesso é basicamente formada por dados estruturados e prevê a utilização da linguagem XML e JSON. O sistema *Tag Manager* é o componente que gerencia o acesso externos, processa respostas dos sistemas baseados em Folksonomia e prepara os dados a serem mostrados para o usuário. Duas conexões HTTP (*Hypertext Transfer Protocol*) são exigidas, sendo uma de interação do usuário com o sistema e outra de interação do sistema com as personomias do usuário.

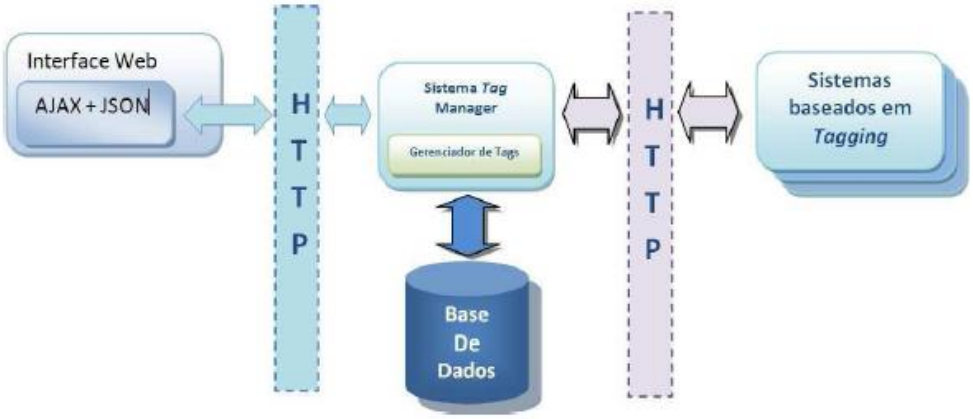


Figura 17 - Estrutura necessária do sistema desenvolvido. Fonte: SILVA (2009).

Conforme descreve o autor, Silva (2009), o sistema de gerenciamento de *tags* foi dividido em cinco camadas acopladas pelas interfaces de programação que trabalham juntas para satisfazer as necessidades do usuário:

1. Camada de interação com o usuário
2. Camada de controle
3. Camada de negócios
4. Camada de persistência de dados
5. Camada de dados

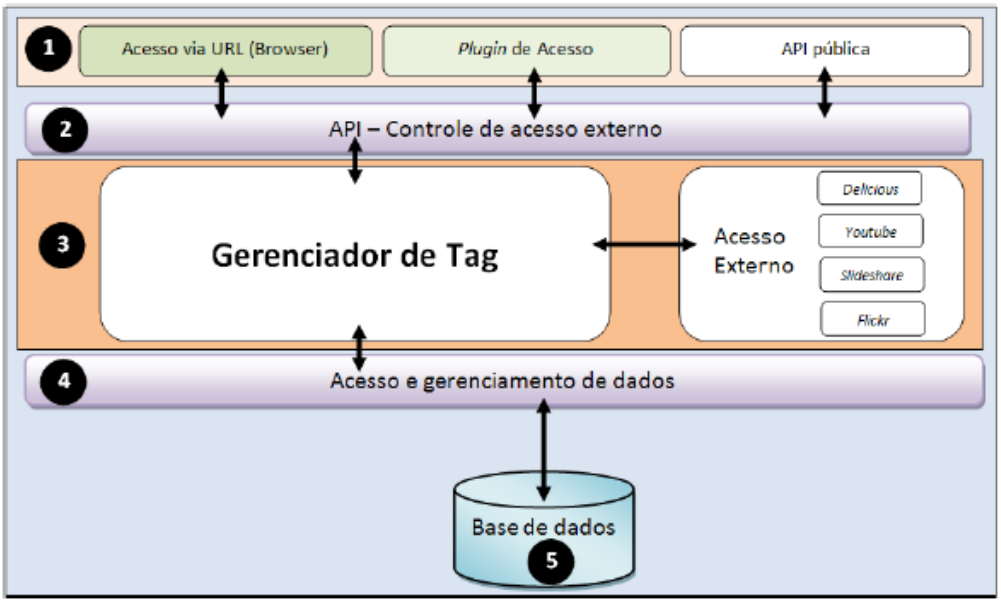


Figura 18 – Arquitetura em camadas do sistema. Fonte: SILVA (2009).

Com a arquitetura ilustrada na Figura 18, o objetivo é agrupar as *tags* de diversos sistemas baseados em Folksonomia, visando minimizar o esforço do usuário no uso do seu vocabulário utilizado para categorização e para a recuperação e recursos, além de facilitar a

manutenção dos termos que compõem esse vocabulário. Outro aspecto considerado é o controle do ciclo de vida de um termo utilizado por um usuário, desde o momento de sua criação ou apropriação até o momento em que ele deixa de ser utilizado para novas categorizações, bem como para as buscas.

As tecnologias utilizadas para a implementação foram as seguintes: Java, em função de algumas APIs disponibilizadas pelos sistemas baseados em Folksonomia (Delicious, Youtube, Flicker e SlideShare) serem implementadas nesta linguagem; Framework HttpClient provê as funcionalidades HTTP para serem utilizadas pelas aplicações; Jericho (Parse HTML) é uma biblioteca Java que permite analisar e manipular documento HTML; API do Delicious, versão 1.14, utilizando a linguagem Java que permite: buscar, nomear ou excluir *tags*, buscar todas as categorizações para um usuário, buscar categorizações para uma *tag* e, também, alterar categorizações; PostgreSQL, é o sistema de base de dados utilizado no projeto.

Por fim, como o principal objetivo de um usuário deste tipo de sistema é a organização da informação de seu interesse, o autor acredita que o sistema desenvolvido (*Tag Manager*) possa auxiliar o trabalho de categorização, de recuperação e de manutenção dos termos utilizados pelo usuário, possibilitando a centralização das personomias e o gerenciamento das mesmas de uma forma unificada. Esse sistema auxilia no controle e gerenciamento do vocabulário do usuário (personomia global), que colabora para a criação de uma camada de *tags* sobre os dados do usuário refletindo na melhor recuperação da informação do objeto.

A apresentação destes trabalhos fortalece a ideia de integração do uso da técnica de etiquetagem chamada Folksonomia em SI com o propósito de filtragem de conteúdo.

Os dois primeiros artigos situam a importância da filtragem de conteúdo nos dias atuais, tanto em companhias, quanto em ambientes domésticos. Os dois últimos artigos ilustram dois trabalhos de integração de Folksonomia em Sistemas de Informação. Portanto, enxergou-se como espaço de pesquisa, a realização deste trabalho de conclusão que utiliza a Folksonomia para criação de um modelo de classificador, a fim de bloquear ou permitir (filtrar conteúdo) o acesso a determinadas URLs, dentro de uma companhia, ou ainda em um ambiente doméstico.

6 ESTUDO DE CASO

Nos dias de hoje é necessário o aumento de controle nas restrições de acessos dentro das corporações aplicando políticas cada vez mais rígidas para minimizar possíveis vulnerabilidades de segurança. Mecanismos de defesa para a proteção do ambiente computacional devem ser planejados e realizados com base no conhecimento das ameaças e dos riscos existentes e nos que possam vir a existir como forma de prevenção e proteção ao sistema em que se está a trabalhar. Neste capítulo é apresentada uma proposta de modelo de classificador que tem como objetivo identificar o acesso aos sites de uma organização ou mesmo de um ambiente doméstico, classificando tais URLs em apropriadas ou bloqueadas para o acesso, a partir da técnica de Folksonomia, utilizando o serviço do Delicious (2015) na captação de *tags*.

Para geração deste modelo foi utilizada a tarefa de classificação de Árvores de Decisão. No entanto, a fim de encontrar um modelo de classificador, foram necessárias diversas etapas de pré-processamento, assim como definido no fluxo de KDD (demonstrado na Figura 6). Após a etapa de pré-processamento, o modelo foi aplicado sobre um conjunto de URLs de treinamento e na etapa de pós-processamento, foi verificada a acurácia do modelo proposto, suas vantagens e deficiências. A Figura 19 ilustra as 11 etapas aplicadas neste trabalho, etapas estas relacionadas ao pré-processamento, descoberta do modelo (*Data*

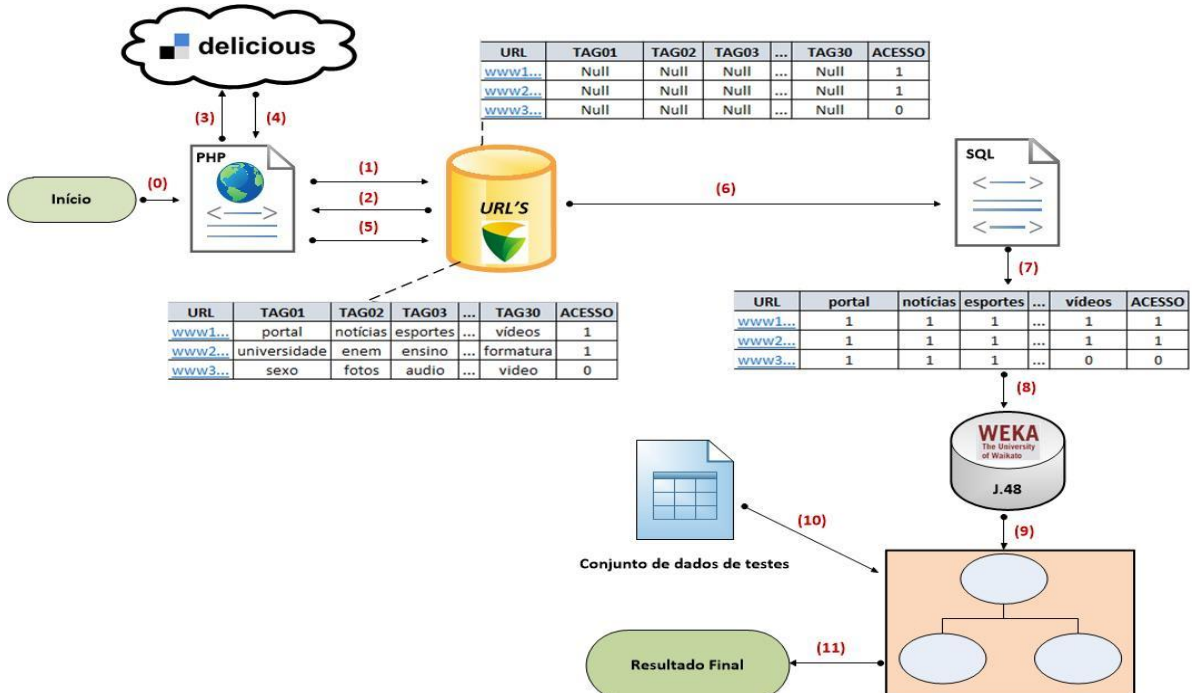


Figura 19 – Fluxo da geração do modelo de classificador. Fonte: desenvolvido pelo autor.

Mining) e pós-processamento.

6.1 PRÉ-PROCESSAMENTO

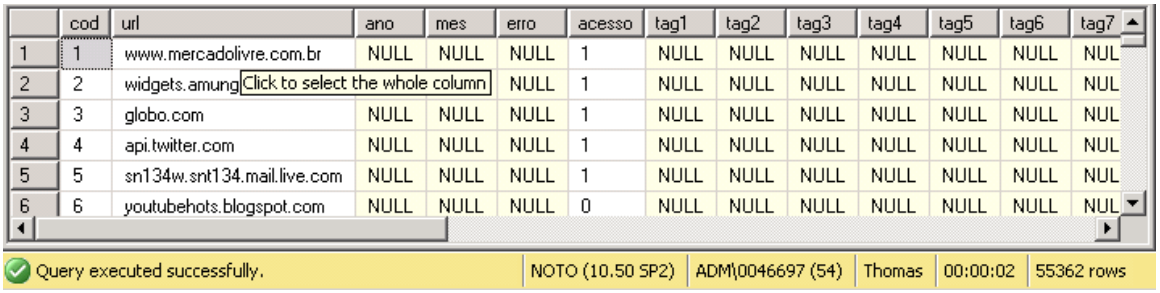
Conforme explanado no capítulo 3, o pré-processamento é uma etapa trabalhosa em função dos dados, na maioria das vezes, não estarem de tal forma organizados a permitir a aplicação direta dos algoritmos de mineração. Desta forma, é necessário selecionar, limpar e transformar os dados de modo que possam servir de entrada ao algoritmo escolhido durante a etapa da mineração de dados.

Observa-se, na Figura 19, que há uma base de dados de URLs. Esta base de dados está armazenada em um SGBD (Sistema Gerenciador de Banco de Dados) SQL SERVER 2008 R2 e contém uma tabela denominada "FolkUrl", responsável por manter 55.362 URLs distintas. Para cada URL registrada, armazenam-se os seguintes atributos:

FolkUrl				
Nome	**Tipo de Dados**	**PK**	**FK**	**Descrição**
cod	INT	SIM		Código do endereço *Web*
url	NVARCHAR(Max)			Endereço *Web*
ano	INT			Ano do cadastro
mês	INT			Mês do cadastro
erro	INT			Informa se o endereço é válido
acesso	INT			Flag indicando se o conteúdo do endereço é acessível ou não conforme os critérios de acesso. Consideramos as seguintes categorias: Material adulto (pornografia) e segurança (Vírus): 0 - Bloqueado 1 - Acesso permitido
tag1	NVARCHAR(Max)			*Tag* coletada no site Delicious
tag2	NVARCHAR(Max)			*Tag* coletada no site Delicious
...	...	...	...	...
tag30	NVARCHAR(Max)			*Tag* coletada no site Delicious

Figura 20 – Tabela "FolkUrl – Estrutura e atributos. Fonte: desenvolvido pelo autor.

Os registros de URLs da tabela "FolkUrl" foram coletados a partir de acessos realizados pelos usuários de uma universidade localizada no Vale do Rio dos Sinos, durante o período de 30 dias. A partir destes acessos registrados, foi possível classificar cada URL como acessível ou não, por meio do atributo "acesso". O valor "1" identifica que a URL teve o acesso permitido e o valor "0" informa que a URL foi bloqueada, conforme é possível observar na Figura 21.



	cod	url	ano	mes	erro	acesso	tag1	tag2	tag3	tag4	tag5	tag6	tag7
1	1	www.mercadolivre.com.br	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NUL
2	2	widgets.amung	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NUL
3	3	globo.com	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NUL
4	4	api.twitter.com	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NUL
5	5	sn134w.snt134.mail.live.com	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NUL
6	6	youtubehots.blogspot.com	NULL	NULL	NULL	0	NULL	NULL	NULL	NULL	NULL	NULL	NUL

Click to select the whole column

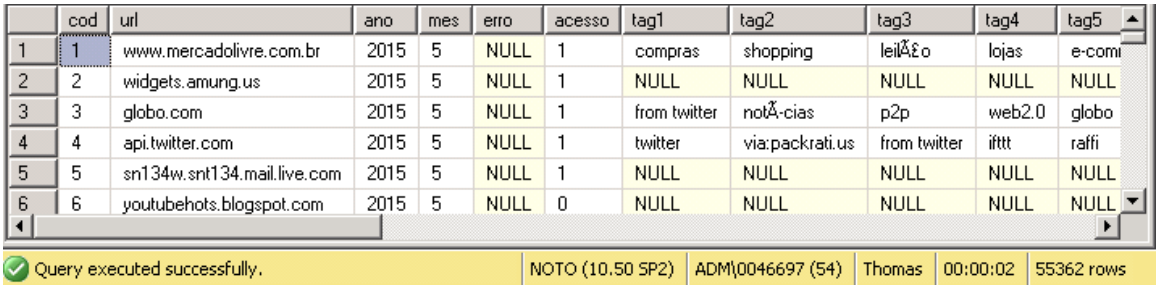
Query executed successfully. | NOTO (10.50 SP2) | ADM\0046697 (54) | Thomas | 00:00:02 | 55362 rows

Figura 21 – Tabela FolkUrl com atributo "acesso" preenchido. Fonte: desenvolvido pelo autor.

Para a inserção dos valores no atributo "acesso", foram considerados as categorias "pornografia" e "vírus" da ferramenta WEBSENSE (2015). Desta maneira, todas as URLs que estavam classificadas como uma destas duas categorias quando processadas pela ferramenta da instituição, foram parametrizadas com o acesso negado (0 - NÃO).

Etapa 1: Execução do script load_tags.php

De posse das 55.362 URLs, foi necessário criar um *script* para atualizar cada URL com as *tags* que estão associadas a ela no site Delicious. Este *script*, denominado "load_tags.php", é responsável por capturar todas as URLs da tabela "FolkUrl" (etapas 0, 1 e 2 da Figura 19). Posteriormente, este *script* acessa o site Delicious e recupera as *tags* de cada URL lida da tabela "FolkUrl" (etapas 3 e 4 da Figura 19). Por fim, o *script* salva na mesma tabela "FolkUrl" as *tags* recuperadas para cada *URL*, bem como o ano e o mês em que foi realizada esta operação (etapa 5 da Figura 19). Após a aplicação do *script*, a tabela "FolkUrl"



	cod	url	ano	mes	erro	acesso	tag1	tag2	tag3	tag4	tag5
1	1	www.mercadolivre.com.br	2015	5	NULL	1	compras	shopping	leilÃ£o	lojas	e-comi
2	2	widgets.amung.us	2015	5	NULL	1	NULL	NULL	NULL	NULL	NULL
3	3	globo.com	2015	5	NULL	1	from twitter	notÃ-cias	p2p	web2.0	globo
4	4	api.twitter.com	2015	5	NULL	1	twitter	via:packrati.us	from twitter	ifttt	raffi
5	5	sn134w.snt134.mail.live.com	2015	5	NULL	1	NULL	NULL	NULL	NULL	NULL
6	6	youtubehots.blogspot.com	2015	5	NULL	0	NULL	NULL	NULL	NULL	NULL

Query executed successfully. | NOTO (10.50 SP2) | ADM\0046697 (54) | Thomas | 00:00:02 | 55362 rows

Figura 22 – Tabela FolkUrl com *tags* do Delicious. Fonte: desenvolvido pelo autor.

apresenta-se conforme a Figura 22.

Desta forma, o *script* "load_tags.php" foi utilizado para varrer cada registro (URL) da tabela "FolkUrl", e buscar as *tags* associadas a esta URL. Posteriormente, foram preenchidos os atributos "*tag1*", "*tag2*", "*tag3*", ..., "*tag30*" com as respectivas *tags* "Populares" e

“Recomendadas” encontradas. Este *script* foi implementado utilizando a linguagem PHP, versão 5.6.7, conforme o código da Figura 23:

```
include "conexao.php";
include "UrlDao.class.php";
include "Url.class.php";
include "Delicious.class.php"

$listaUrls = UrlDao::getUrls($conn);

foreach ($listaUrls as $url) {

      $xmlDelicious = UrlDao::getXmlByUrlFromDelicious($url-
>getUrl());

$listaTagsPopulares =
UrlDao::getTagsPopularesByXmlFromDelicious($xmlDelicious);

      $listaTagsRecomendadas =
UrlDao::getTagsRecomendadasByXmlFromDelicious($xmlDelicious);

      $listaTags =
array_merge($listaTagsPopulares,$listaTagsRecomendadas);

      UrlDao::saveTags($conn, $url->getCodUrl(), $url-
>getUrl(),$listaTags);

}
```

Figura 23 - *Script* de conexão “load_tags.php”. Fonte: desenvolvido pelo autor.

Para que o *script* “load_tags.php” funcione corretamente foram criados alguns arquivos auxiliares e algumas classes, descritas abaixo. Estes arquivos e classes têm seus códigos apresentados no Apêndice A.

* arquivo “conexao.php”: instruções que permitem a conexão com o SGBD SQL Server, local onde estão armazenadas as URLs.

* Delicious.class.php: classe necessária para conectar-se no site Delicious. Para utilização do site Delicious, foi necessário cadastrar-se previamente no mesmo, a fim de obter um usuário e senha. Logo, esta classe permitiu o envio de uma URL ao site Delicious e a recuperação dos dados enviados pelo site em formato XML. Esta classe foi adaptada de GEEK-ALPHA (http://www.geek-alpha.com/php/using-php-connect-delicious-api-classe) com base na API Delicious V1.

* Url.class.php: classe desenvolvida para armazenar os objetos retornados no XML da classe “Delicious.class.php". Basicamente, é uma classe com dois atributos $codUrl e $linkUrl, com métodos *set* e *get*.

* UrlDao.class.php: classe responsável por acessar o banco de dados e operar sobre cada registro da tabela "FolkUrl". Esta classe transforma o XML resgatado para uma determinada URL e, salva as *tags* de cada URL dentro da tabela "FolkUrl".

Etapa 2: Transformação da tabela “FolkUrl”

Após o preenchimento das *tags* para cada URL na tabela “FolkUrl”, foi verificado, a partir do comando SQL descrito abaixo, que 38933 urls não tem *tags* associadas.

```
select COUNT(*) from FolkUrl where tag1 is null
```

Devido a característica de etiquetagem social/colaborativa do site Delicious (2015) as URLs consultadas e retornadas com nenhuma *tag* para o endereço em questão, tem como motivo que os usuários do site ainda não a etiquetaram.

Para a melhor compreensão das informações coletadas no site Delicious e transferidas para dentro da tabela “FolkUrl”, foram executados os comandos SQL, da Figura 24, a fim de analisar a quantidade de *tags* associadas por atributo (*tag*1, *tag*2, ..., *tag*30), conforme a estrutura da tabela “FolkUrl”.

```
Select count(*) from FolkUrl where tag1 is not null
Select count(*) from FolkUrl where tag2 is not null
...
Select count(*) from FolkUrl where tag30 is not null
```

Figura 24 - Comandos SQL para analisar as *tags* associadas. Fonte: desenvolvido pelo autor.

Com os resultados dos comandos acima foi desenvolvido um gráfico no qual representa a quantidade de *tags* associadas por URL. Foi possível observar que 16429 URLs possuem no mínimo 3 *tags* cadastradas. Conclui-se, portanto, que os usuários associam em sua maioria 3 *tags* por endereço dentro do site Delicious. Diante dos números é possível concluir que a quantidade de *tags* associadas tem uma redução a partir da 4ª *tag* (atributo *tag*4), chegando à zero (0) na 11ª *tag* (atributo *tag*11).

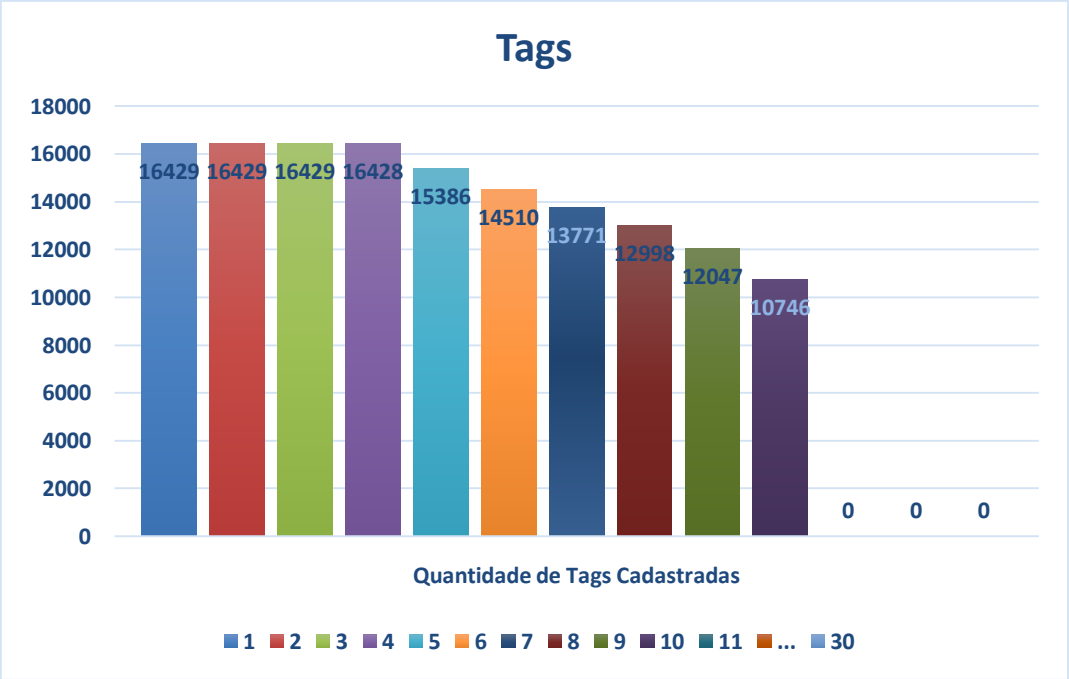


Figura 25 – Número de *tags* cadastradas por atributos *tag1, tag2, ..., tag10*. Fonte: desenvolvido pelo autor.

Na etapa de limpeza de dados, foram eliminadas da tabela "FolkUrl" as URLs que não possuíam nenhuma *tag* cadastrada. Abaixo, segue o comando SQL executado:

```
delete from FolkUrl where tag1 is null
```

A partir deste procedimento, restaram 16429 linhas (URLs). Destas, 1243 URLs que tiveram seu acesso bloqueado e 15186 dos registros com acesso permitido, conforme pode ser visualizado na Figura 26.

```
select acesso, COUNT(*) as "Número de Registros"
from FolkUrlWork
group by acesso
```

	acesso	Número de Registros
1	0	1243
2	1	15186

Figura 26 - Número de registros de solicitações de acesso. Fonte: desenvolvido pelo autor.

Para melhor representação dos indicadores acima, o gráfico abaixo apresenta que 92% das solicitações tiveram seu acesso permitido e 8% bloqueado, evidenciando nesta tabela um grande desbalanceamento em relação ao atributo "acesso".

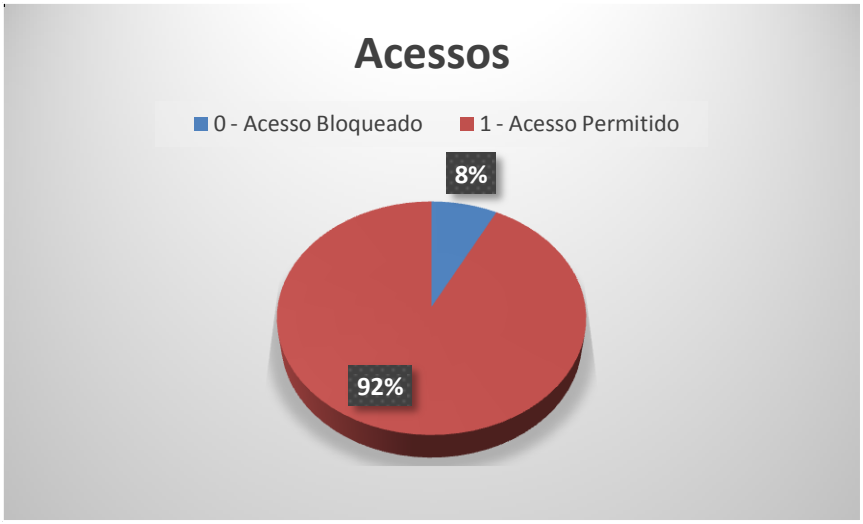


Figura 27 - Percentual de acessos bloqueados e permitidos da tabela "FolkUrl". Fonte: desenvolvido pelo autor.

A partir da tabela "FolkUrl", foi criado o arquivo "FolkUrl.csv" (etapas 6 e 7 da Figura 19), a fim de permitir sua entrada no programa WEKA (conforme etapa 8 da Figura 19).

Etapa 3: Limpeza do arquivo "FolkUrl.csv"

Para fazer com que o arquivo "FolkUrl.csv" pudesse ser lido pela ferramenta WEKA e consequentemente aplicada alguma técnica de *Data Mining*, foram necessárias diversas tarefas para limpeza dos dados. Inicialmente foram retirados os atributos que não possuíam qualquer conteúdo para as *tags* associadas, ou seja, foram eliminados os atributos correspondentes às colunas *tag*11, *tag*12, *tag*13, ..., *tag*30.

Como as *tags* são escritas em diversos idiomas foi necessário trocar determinadas codificações pelo seu símbolo adequado. Foram retirados os símbolos de ";" existentes em alguns conteúdos de *tags*. A Figura 28 ilustra algumas transformações dos dados após a limpeza.

Pré Limpeza	Pós Limpeza
www;youtube.com	www.youtube.com
Programming & IT	Programming & IT
educação;	educação
Casts&Blogs	Casts Blogs

Figura 28 - Limpeza relacionada aos símbolos ";" e transformações de conteúdos em idiomas diferentes. Fonte: desenvolvido pelo autor.

Também identificou-se que alguns conteúdos foram cadastrados pelos usuários com aspas, conforme Figura 29. Portanto, os elementos que tinham padrões de aspas foram transformados, eliminando-se as aspas. A Figura 29, ilustra esta transformação dos dados.

Pré Limpeza	Pós Limpeza
aerea"	aerea
catarina"	catarina
example"	example

Figura 29 - Limpeza de aspas. Fonte: desenvolvido pelo autor.

Conforme explanado no capítulo 3, o pré-processamento é uma etapa trabalhosa em função dos dados não estarem de tal forma organizados a permitir a aplicação direta dos algoritmos de mineração. Após a realização das tarefas de limpeza mencionadas, foi criado um arquivo denominado "FolkUrl_Limpo.csv", conforme ilustrado na Figura 30.

cod	url	ano	mes	erro	acesso	tag1	tag2	tag3	tag4	tag5	tag6	tag7	tag8	tag9	tag10
1	www.sebrae.com.br	2015	5	NULL	SIM	empreendedorismo	sebrae	negócios	...	...	...	...	...	carreira	pesquisa
2	www.popmatters.com	2015	5	NULL	SIM	music	culture	reviews	...	...	...	...	...	blog	books
3	www.carlosmartins.com.br	2015	5	NULL	SIM	marketing	empreendedorismo	planodenegócios	...	...	...	...	...	NULL	NULL
4	www.youhide.com	2015	5	NULL	NÃO	proxy	anonymous	privacy	...	...	...	...	...	proxys	Barra de favoritos
5	mowser.com	2015	5	NULL	SIM	mobile	browser	web	...	...	...	...	...	conversion	lifehacks
6	www.machadodeassis.org.br	2015	5	NULL	SIM	literatura	machado	biblioteca	...	...	...	...	...	escritor	Portuguesa
7	kilobyte.com.br	2015	5	NULL	SIM	blog	Proxy	revistas	...	...	...	...	...	NULL	NULL
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
16428	www.netburner.com	2015	5	NULL	SIM	embedded	electronics	hardware	...	...	...	...	...	robotics	project
16429	www.netcomputadores.com.br	2015	5	NULL	SIM	loja	componentes	suprimentos	...	...	...	...	...	compras	Tecnologia_-_vendas

Figura 30 - Arquivo "FolkUrl_Limpo.csv" gerado após a limpeza do conteúdo das *tags*. Fonte: desenvolvido pelo autor.

6.2 TREINAMENTO E TESTE PRELIMINAR

Após a etapa de limpeza dos dados, foi possível carregar o arquivo "FolkUrl_Limpo.csv" dentro da ferramenta WEKA. Desta forma, foi realizada uma primeira avaliação preliminar, a fim de verificar a qualidade do modelo de árvore de decisão a ser gerado com base neste arquivo. Para esta primeira avaliação os atributos *cod, url, ano, mes e erro* foram eliminados, restando somente os atributos: "acesso, *tag*1, *tag*2, *tag*3, ..., *tag*10", conforme ilustrado na Figura 31. Estes atributos foram removidos, pois não são necessários na geração do modelo, haja vista que pretende-se encontrar um modelo que relacione as *tags* existentes com o atributo "acesso": permitido ou bloqueado (SIM ou NÃO).

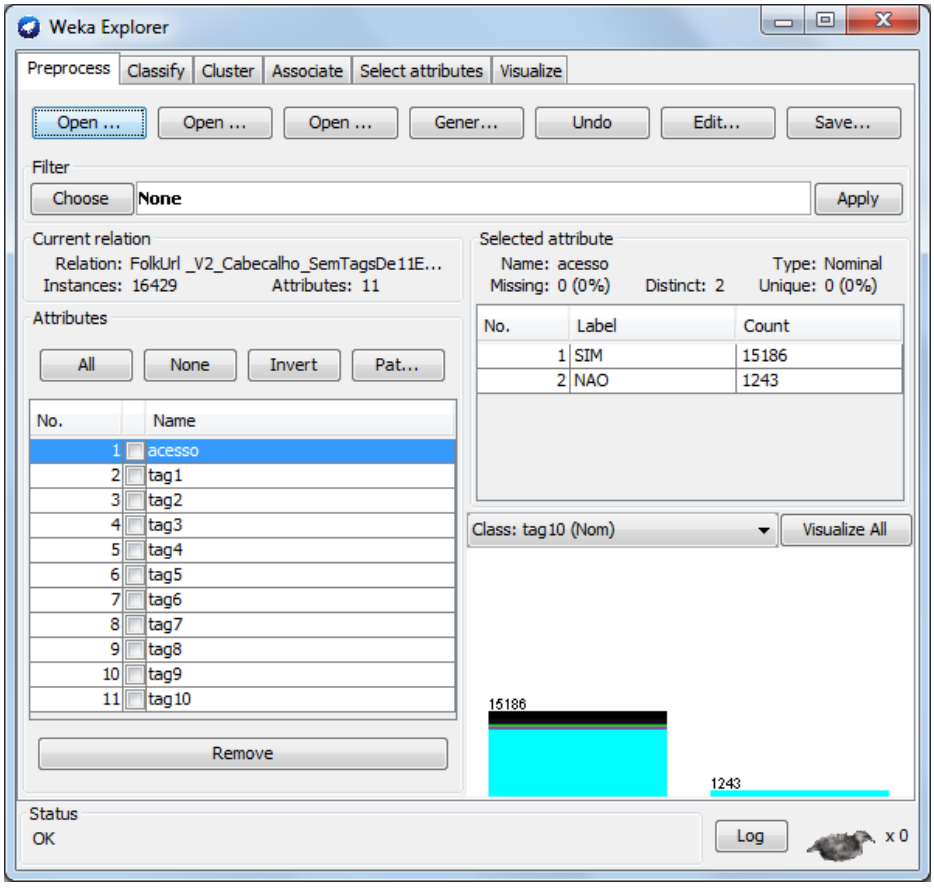


Figura 31 - Remoção de atributos dentro da ferramenta WEKA. Fonte: desenvolvido pelo autor.

Posteriormente, foi aplicado o algoritmo J48 sobre os dados inseridos na ferramenta WEKA. Configurou-se o atributo “acesso” como o atributo alvo e foi utilizado como configuração de treinamento e teste: *cross-validation* com 10 *folds*. Como pode ser observado na Figura 32, foi gerado um modelo ineficiente com uma árvore com somente 1 folha, tendo classificado todos os registros como acesso permitido: SIM, conforme pode ser visualizado na Figura 31.

Desta forma, foram classificados corretamente 92.43% dos registros (instâncias), que são justamente todos os 15186 registros existentes com “acesso” SIM e 7.56% (1243) dos registros foram classificados incorretamente. Estas 1243 instâncias têm o valor original do atributo “acesso” igual a NÃO e também foram classificados como “acesso” SIM. A Figura 33, que ilustra a matriz de confusão, deixa clara a ineficiência do modelo gerado.

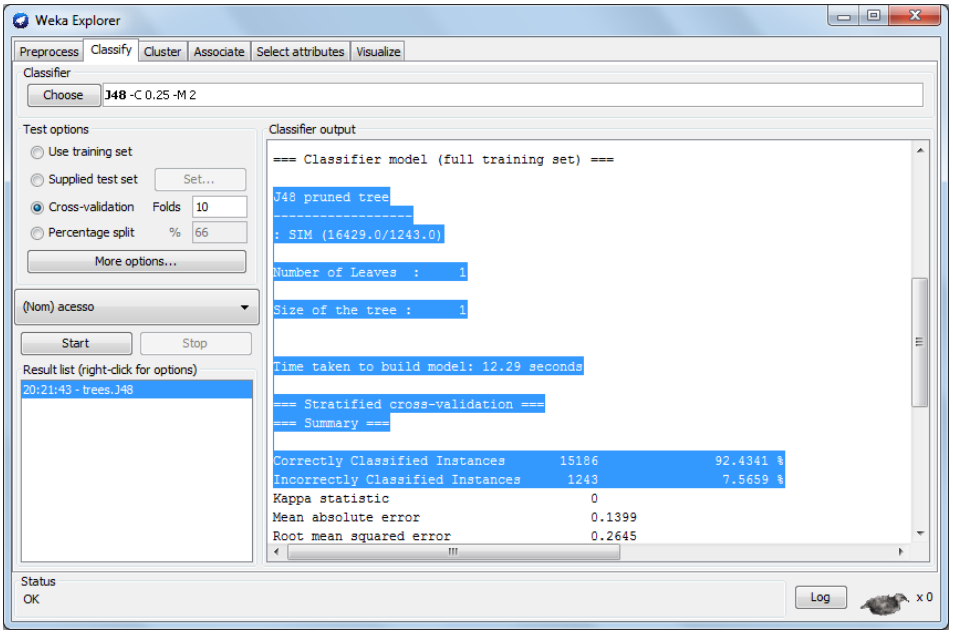


Figura 32 - Treinamento e teste preliminar aplicando o algoritmo J48 sobre os dados. Fonte: desenvolvido pelo autor.

```
=== Confusion Matrix ===

     a     b   <-- classified as
 15186     0 |     a = SIM
  1243     0 |     b = NAO
```

Figura 33 - Matriz de confusão gerada a partir da aplicação do algoritmo J48. Fonte: desenvolvido pelo autor.

6.3 NOVO PRÉ-PROCESSAMENTO

Como o teste preliminar do modelo gerado tem um resultado ruim, foram realizadas diversas transformações no arquivo “FolkUrl_Limpo.csv”, a fim de buscar um modelo de classificador mais adequado. Estas transformações tiveram o objetivo de gerar uma planilha semelhante àquela representada na Figura 19, etapa 7. Ou seja, ao invés de ter como atributos *tag*1, *tag*2, *tag*3, ..., *tag*10, os conteúdos destes 10 atributos foram varridos e transformados em atributos. Exemplificando, se o atributo *tag*1, por exemplo, tiver como conteúdo blog, wiki e sexo, estes três valores serão representados como três atributos na planilha construída.

A fim de compreender melhor o conteúdo dos atributos *tag*1, *tag*2, *tag*3, ..., *tag*10, da tabela “FolkUrl”, foram realizadas diversas consultas para descobrir o número de valores distintos em cada um deles (Figura 34). Desta forma, foi possível visualizar e compreender a variedade de termos (futuros atributos) que os usuários utilizam para classificar as URLs.

```
select count(distinct tag1) from folkurl     3515
select count(distinct tag2) from folkurl     4257
select count(distinct tag3) from folkurl     4735
select count(distinct tag4) from folkurl     5103
select count(distinct tag5) from folkurl     4968
select count(distinct tag6) from folkurl     4921
select count(distinct tag7) from folkurl     4831
select count(distinct tag8) from folkurl     4649
select count(distinct tag9) from folkurl     4512
select count(distinct tag10) from folkurl    4511
```

Figura 34 – Consulta SQL para visualizar a variedade de termos nos atributos *tag1, tag2, ..., tag10*. Fonte: desenvolvido pelo autor.

Para facilitar a geração desta planilha final (Figura 19, etapa7), utilizou-se algumas tabelas auxiliares. Inicialmente foi criada uma tabela denominada "*folkurltodastags",* cujo objetivo é armazenar todas as *tags* dos atributos *tag1, tag2, ..., tag10*. Desta forma, foram executadas as consultas detalhadas na Figura 35.

```
create table folkurltodastags (
cod int primary key auto_increment,
tag varchar(100));

insert into folkurltodastags (tag)
select tag1 from folkurl

insert into folkurltodastags (tag)
select tag2 from folkurl

insert into folkurltodastags (tag)
select tag3 from folkurl

...

insert into folkurltodastags (tag)
Select tag10 from folkurl
```

Figura 35 – Consulta SQL para armazenamento das *tags* distintas. Fonte: desenvolvido pelo autor.

A tabela "folkurltodastags" ficou com 164250 registros, resultado dos 16425 registros da tabela "folkurl" multiplicando pelos 10 atributos (*tag1, tag2, ..., tag10*). Todos os valores de *tags* existentes nos atributos *tag1, tag2, ..., tag10* da tabela "folkurl" foram inseridos na tabela "folkurltodastags", mais precisamente no atributo *tag*. Executando-se o comando SQL

da Figura 36 foi possível compreender e visualizar o número de repetições de cada uma das *tags*. A consulta exibida na Figura 36 mostra que existem 25861 *tags* distintas com o número de repetições variados para cada uma delas.

```
select tag, count(tag)
from folkurltodastags
group by tag
order by count(tag) desc
```

tag	count(tag)
blog	3375
design	2184
blogs	1848
tools	1427
news	1409
web	1239
Brasil	1237

Figura 36 - Número de repetições de valores das *tags* na tabela "folkurltodastags". Fonte: desenvolvido pelo autor.

O número de *tags* distintas é muito grande e, de acordo com a transformação que deseja-se fazer, o arquivo final (etapa 7 da Figura 19), teria 25861 atributos. No entanto, existem muitas *tags* que são repetidas poucas vezes. Por exemplo, há 17681 atributos que se repetiram apenas 1 vez. Ou seja, os usuários classificaram no site Delicious, 17681 atributos somente 1 vez.

Logo, a fim de reduzir a dimensionalidade do arquivo e dar preferência aos atributos que tiveram um número de repetições maior, definiu-se coletar somente as *tags* que se repetiram um número maior ou igual a 50 vezes. A Figura 37 ilustra os comandos utilizados para criar a tabela "folkurltodastags_50vezesoumais", responsável por armazenar todas as *tags* que se repetiram com uma frequência maior ou igual a 50 vezes. Ao final restaram 415 *tags*.

```
create table folkurltodastags_50vezesoumais (
cod int primary key auto_increment,
tag varchar(100),
vezes int);

insert into folkurltodastags_50vezesoumais (tag, vezes)
select tag, count(tag)
from folkurltodastags
group by tag
having count(tag) >=50
order by count(tag) desc
```

Figura 37 – Comandos SQL para criação da tabela "folkurltodastags_50". Fonte: desenvolvido pelo autor.

A partir da tabela “folkurltodastags_50vezesoumais”, foi possível gerar o *script* “transpose_tags.php”, ilustrado na Figura 38. Este *script* busca todas as *tags* da tabela “folkurltodastags_50vezesoumais” e as insere como atributos do arquivo “tags.csv”, gerado ao final do *script*. Utilizando-se a tabela original “folkurl” é verificado se para cada linha desta tabela, cada uma das *tags* existentes em “folkurltodastags_50vezesoumais” existe em algum dos atributos *tag1*, *tag2*, ..., *tag10*. Se existir, esta *tag* receberá o valor SIM, caso contrário, receberá o valor NÃO.

```
<?php
$conexao = mysql_connect("localhost","root","12345");
mysql_select_db("test", $conexao);

$sql50 =  "select tag from folkurltodastags_50vezesoumais order by vezes desc";
$tags50 = mysql_query($sql50);
$cabecalho_acesso = "acesso";
$cabecalho_tags = "";
$arrTags = array();
while ($dados = mysql_fetch_array($tags50))
      $cabecalho_tags .= ";" . trim($dados["tag"]);
$cabecalho = $cabecalho_acesso . $cabecalho_tags . "\n";

$arr_tags = split(";",$cabecalho_tags);
$sql = "select acesso, tag1, tag2, tag3, tag4, tag5, tag6, tag7, tag8, tag9, tag10
from folkurl";
$tags = mysql_query($sql);
$linha = "";
while ($dados = mysql_fetch_array($tags)) {
      $linha .= $dados[0];
      for ($i=1; $i<sizeof($arr_tags); $i++) {

            if ($arr_tags[$i] == $dados[1] ||
                  $arr_tags[$i] == $dados[2] ||
                  $arr_tags[$i] == $dados[3] ||
                  $arr_tags[$i] == $dados[4] ||
                  $arr_tags[$i] == $dados[5] ||
                  $arr_tags[$i] == $dados[6] ||
                  $arr_tags[$i] == $dados[7] ||
                  $arr_tags[$i] == $dados[8] ||
                  $arr_tags[$i] == $dados[9] ||
                  $arr_tags[$i] == $dados[10])
                        $linha .= ";SIM";
            else
                        $linha .= ";NAO";
      }
      $linha .= "\n";
}

$arquivo = fopen("tags.csv", "w");
fwrite($arquivo,$cabecalho);
fwrite($arquivo,$linha);
fclose($arquivo);
?>
```

Figura 38 – *Script* PHP “transpose_tags.php”. Fonte: desenvolvido pelo autor.

Ao final da execução do *script* “transpose_tags.php” é gerado um arquivo (“tags.csv”), conforme representado na Figura 39. Nesta figura estão ilustrados 10 registros com os 10 primeiros atributos (*tags*). Ao total o arquivo tem 16425 registros e 415 atributos. Além dos 415 atributos que representam as *tags* foi acrescentado também neste arquivo o atributo “acesso”, que na tabela original (“FolkUrl”) estava associado ao acesso ou não obtido pela URL. O arquivo “tags.csv” portanto, representado na Figura 39, será a entrada para o algoritmo J48.

	A	B	C	D	E	F	G	H	I	J	K
1	acesso	blog	web	Brasil	software	fashion	notícias	download	shopping	music	programming
2	SIM	NAO	NAO	NAO	NAO	NAO	NAO	NAO	SIM	NAO	NAO
3	SIM	NAO	NAO	NAO	NAO	NAO	SIM	NAO	NAO	NAO	NAO
4	SIM	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO
5	NAO	NAO	NAO	NAO	NAO	NAO	NAO	SIM	NAO	NAO	NAO
6	SIM	NAO	NAO	NAO	NAO	NAO	SIM	NAO	NAO	NAO	NAO
7	SIM	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO
8	SIM	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO
9	SIM	NAO	NAO	NAO	NAO	SIM	NAO	NAO	SIM	NAO	NAO
10	SIM	NAO	SIM	NAO	NAO	NAO	NAO	NAO	NAO	NAO	NAO

Figura 39 – Porção do arquivo “tags.csv”. Fonte: desenvolvido pelo autor.

Alguns atributos apareceram repetidamente nesta planilha, pois quando foram recuperados no site Delicious, vieram com alguma “sujeira”. Por exemplo, o atributo “web2.0”, apareceu em 2 colunas, isto porque ele foi recuperado tanto como “web20”, quanto como “web2;0”. Por este motivo, os atributos que apareceram de modo repetido sofreram algumas transformações.

Sobre o atributo “web2.0”, que aparece na coluna “L” da planilha e o atributo “web2.0 ” que está presente na coluna “GW” foi realizada a seguinte transformação: (a) foi criada uma outra coluna; (b) nesta coluna aplicou-se a seguinte fórmula: =IF(OR(L2="SIM";GW2="SIM");"SIM";"NAO") na linha 2; (c) a fórmula foi aplicada para todas as linhas da planilha. Desta forma, os dois atributos que têm o mesmo significado viraram um atributo somente. A mesma ideia foi aplicada em outros atributos que também estavam repetidos.

6.4 NOVO TREINAMENTO E TESTE

Usando o arquivo “tags.csv” foi aplicado sobre ele o algoritmo J48. Configurou-se o atributo “acesso” como o atributo alvo no modelo de dados e foi utilizado como configuração de treinamento e teste: *cross-validation* com 10 *folds*. O algoritmo gerou um modelo com 67 folhas e com tamanho da árvore igual a 133. Este modelo classificou corretamente 94.02%

dos registros, conforme pode ser visualizado na Figura 40, superando o modelo preliminar que havia sido gerado.

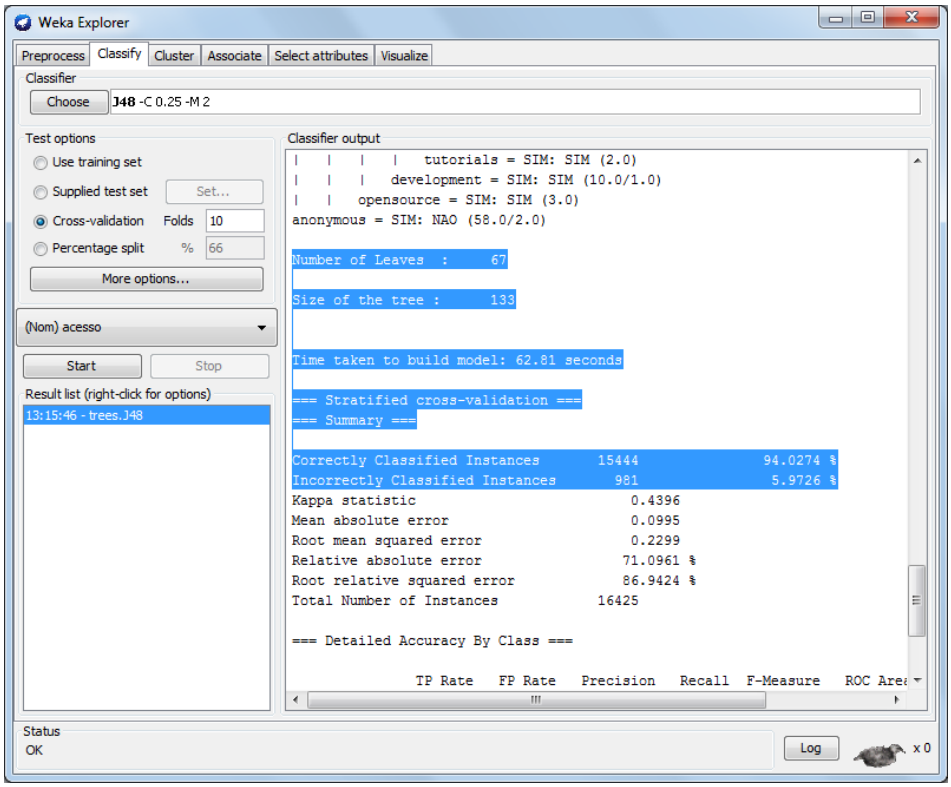


Figura 40 – Treinamento WEKA J48 com arquivo "tags.csv". Fonte: desenvolvido pelo autor.

A Figura 41 exibe a matriz de confusão resultante após a execução do algoritmo J48. Nela está descrito que das 15.182 instâncias que tem originalmente o valor SIM no atributo "acesso", 15014 foram classificadas corretamente, e 168 instâncias classificadas incorretamente. Das 1243 instâncias que tem o valor original NÃO no atributo "acesso", 813 foram classificadas incorretamente, e 430 classificadas de modo correto. O resultado do classificador provavelmente teria um melhor resultado na classificação do atributo "acesso" como NÃO, se houvesse um número maior de instâncias de URLs que tivessem sido bloqueadas na tabela "FolkUrl".

```
=== Confusion Matrix ===

     a     b   <-- classified as
 15014   168 |     a = SIM
   813   430 |     b = NAO
```

Figura 41 - Resultado matriz de confusão algoritmo J48. Fonte: elaborado pelo autor.

Por fim, após um árduo trabalho na restruturação no arquivo de entrada do WEKA com a definição das tags como atributos principais no arquivo "tags.csv", obteve-se um

resultado animador onde fortalece o uso da Folksonomia em conjunto com o algoritmo de árvores de decisão em um classificador de restrição de acesso à sites da Internet.

7 CONCLUSÃO

O objetivo neste trabalho foi apresentar um modelo de indução de classificador com ênfase no processo de mineração de dados, que leva ao descobrimento de conhecimento no modelo decisório para o acesso ao conteúdo da Internet.

Sistemas de restrição de conteúdo baseados somente em endereços da Internet têm perdido sua eficácia em virtude da dinamicidade e capacidade que sites hospedem seus conteúdos de forma distribuída. Desta forma, o classificador gerado apresentou bons resultados a fim de auxiliar sistemas de restrição de conteúdo.

Para desenvolver o processo de mineração de dados, utilizou-se neste modelo a técnica de árvores de decisão, com o algoritmo C4.5 (J48), a fim de gerar um classificador a partir da extração de uma base de dados contendo 55.362 URLs de uma universidade localizada no Vale do Rio dos Sinos. Conforme apresentado no capítulo 2, o site Delicious, organiza suas informações por meio da etiquetagem social de seus usuários. Observou-se pelas *tags* analisadas que a etiquetagem do usuário, embora não possa ser considerada perfeita, pode ser aproveitada como uma ferramenta de apoio para sistemas de restrição de conteúdo.

Pelas características do site e das *tags* coletadas, é possível perceber que os usuários usam seu próprio vocabulário, e, assim, contribuem com a organização das informações, pois suas *tags* são absorvidas pelo sistema e passam a fazer parte do coletivo. Diante da Folksonomia, a preocupação do usuário não é de organizar a informação e nem contribuir com o ambiente *Web*; sua preocupação é etiquetar para si e para o seu grupo social. A consequência desse ato, de etiquetar, é uma das maneiras de organização da informação nas redes sociais.

Como visto no capítulo 5, os trabalhos relacionados, fortalecem a ideia de integração do uso da técnica de etiquetagem chamada Folksonomia em SI. Os trabalhos relacionados, estabelecem a importância da filtragem de conteúdo atualmente, tanto no âmbito corporativo, quanto em ambientes domésticos. Deste modo, fortaleceu a realização deste trabalho, a fim de desenvolver um modelo de classificador.

Durante o desenvolvimento do classificador um dos maiores desafios foi a etapa do pré-processamento, onde se fez necessária a limpeza do arquivo a ser utilizado no software WEKA, alterando diversos símbolos para suas codificações adequadas. Verificou-se que no

primeiro processamento, com a primeira ideia de atributos (*tag1, tag2, ..., tag10*) foi gerado um modelo ineficiente. Para obter melhor resultado foi realizada a alteração no arquivo de entrada coletando somente os atributos (*tags*) que foram repetidos um número maior ou igual a 50 vezes. A partir desta transformação, o arquivo de entrada ficou com um número grande de atributos e, gerou um modelo que indica um bom potencial da utilização de Folksonomia neste contexto.

Como possíveis trabalhos futuros entende-se como necessário agrupar atributos. Há uma variação sintática de alguns atributos que têm semânticas iguais e prejudicam a construção do modelo, em virtude das características da Folksonomia. Isto porque os usuários da *Web* utilizam os termos de maneira livre e aleatória para etiquetar as URLs.

Outro trabalho futuro possível é resgatar uma base de dados mais homogênea da universidade. A base de dados utilizada neste trabalho estava muito desbalanceada, pois ela possuía muitos registros com o valor do atributo "acesso" SIM e poucos com o valor do atributo "acesso" NÃO. Uma amostra com mais URLs com valor de atributo acesso “NÃO” provavelmente traria um melhor desempenho do classificador.

.

8 REFERÊNCIAS BIBLIOGRÁFICAS

A.C.MONOGRAFIA. Estudo de caso. Disponível em:< http://www.monografiaac.com.br/estudo-caso.html> Acesso em: 21 de fev. 2015.

ADRIAANS, Pieter; ZANTINGE, Dolf. Data Mining. Harlow: Addison-Wesley, 1997. 158p.

ALMEIDA, F.C.; DUMONTIER, P. O Uso de Redes Neurais em Avaliação de Riscos de Inadimplência, Revista de Administração FEA/USP, São Paulo, jan/mar 1996.

AMSTEL, Frederick van. Folksonomia: Vocabulário descontrolado, Anarquitetura da informação ou samba do crioulo doido? São Paulo, p. 15, 2007. Disponível em: < http://www.encontroai.org/viewabstract.php?id=34&cf=1 >. Acesso em: 17 fev. 2015.

AQUINO, Maria Clara. Hipertexto 2.0, Folksonomia e memória coletiva: um estudo das tags na web. E-Compós (Brasília), n. 9, nov. 2007.

BARANAUSKAS, José Augusto; MONARD, Maria Carolina. Extração automática de conhecimento por múltiplos indutores. 2001. 181p. Tese (Doutorado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos.

BARAZETTI, Mônica Cristina. Data Mining, (2001). Disponível em: <http://www.pr.gov.br/batebyte/edicoes/1999/bb90/estagiario.htm>. Acesso em: 21 mar. 2015.

BARRACUDA. Barracuda Web Filter. Disponível em: <http://www.barracudanetworks.com/ns/products/spyware_overview.php> Acesso em: 12 fev. 2015.

BARRACUDA. Disponível em: <http://barracudanetworks.com> Acesso em: 12 fev. 2015.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. Nature Publishing Group, Maio 2001. Disponível em: <http://www.scientificamerican.com/article.cfm?id=the-semantic-web&print=true> Acesso

em: 17 jan. 2015.

BRAMER, M. (2007). Principles of data mining. Springer, London.

BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A., & STONE, C. J. (1984). Classification and Regression Trees. Wadsworth.

BREITMAN, Karin Koogan. Web semântica: a Internet do futuro. Rio de Janeiro :LTC, 2005.

CATARINO, M.; BAPTISTA, A. Folksonomia: um novo conceito para a organização dos recursos digitais na Web. DatagramaZero, Rio de Janeiro, v. 8, n. 3, 2007. Disponível em: <http://dgz.org.br/jun07/Art_04.htm>. Acesso em: 20 jan. 2015.

CERVO, Amado L.; BERVIAN, Pedro A. Metodologia científica. São Paulo: Prentice Hall, 2002.

CHAPELLE, O., SCHÖLKOPF, B., & ZIEN, A. (2006). Semi-supervised learning. MIT Press.

CLARK, P.; Niblett, T. The CN2 Induction algorithm. Machine Learning, p.261-283. 1989.

COBO Romaní, Cristóbal;Pardo Kuklinski, Hugo. 2007. Planeta Web 2.0. Inteligencia colectiva o medios fast food. Grup de Recerca d'Interaccions Digitals, Universitat de Vic.Flacso México. Barcelona / México DF.

CORY, Doctorow. 2008. Content: Selected Essays on Technology, Creativity, Copyright, and the Future of the Future. Disponível em <http://craphound.com/content/>. Acessado em: 01 mar. 2015.

DAVENPORT, Thomaz H. Ecologia da informação: porque só a tecnologia não basta para o sucesso da era da informação. São Paulo: Futura, 2000.

DAVIS, Ian. Talis, Web 2.0 and All That. Disponível em: <http://iandavis.com/blog/2005/07/ talis-*Web*-20-and-all-that>. Acesso em: 18 jan. 2015.

DELICIOUS. Disponível em:<http://delicious.com> Acesso em: 10 jan. 2015.

EIBE, Ian Witten et al. Weka: Practical Machine Learning Tools and Techniques with Java Implementations. In: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING, 6., 1999, Perth, Australia. Proceedings... . Piscataway, NJ, USA: Morgan Kaufmann, 1999. p. 192-196.

ESPOSITO, Floriana; MALERBA, Donato; SEMERARO, Giovanni. A comparative analysis of methods for pruning decision trees. IEEE Transactions on Pattern Analysis and Machine Intelligence, New Yourk, v. 19, n. 5, p. 476-491, May 1997.

FAYYAD, Usama; Piatetski-Shapiro, Gregory; Smyth, Padhraic (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data. In: Communications of the ACM, pp.27-34, nov. 1996.

FEITOSA, Ailton. Organização da informação na Web: das *tag*s à *Web* semântica. Brasília: Thesaurus, 2006. 131p.

FLICKR: photo sharing. Disponível em: <http://www.flickr.com>. Acesso em: 25 jan. 2015.

FONSECA, José Manuel Matos Ribeiro da. Indução de árvores de decisão: HistClass - proposta de um algoritmo não paramétrico. 1994. 140p. Dissertação (Mestrado em ngenharia Informática) - Departamento de Informática, Universidade Nova de Lisboa, Lisboa.

FREEMAN, James A; SKAPURA, David M.. Neural Networks Algorithms, Applications and Programmming Techniques. In:. : , 1992. .

FURNKRANZ, J.; GAMBERGER, D.; LAVRAC, N. Foundations of Rule Learning.[S.l.]: Springer-Verlag Berlin, 2012.

GAMA, J. M. P. (1999). Combining Classication Algorithms. PhD thesis, Faculdade de Ciências da Universidade do Porto, Departamento de Ciências de Computadores.

GOUVÊA, Cleber e LOH, Stanley. Folksonomias: identificação de padrões na seleção de *tag*s para descrever conteúdos. RESI – Revista Eletrônica de Sistemas de Informação.Ed. 11, n. 2, 2007. Disponível em <http://revistas.facecla.com.br/index.php/reinfo/article/view/214/118>. Acessado em 12 de fev.2015.

HARMON, P.; KING, D. Sistemas especialistas. Tradução Antônio Fernandes Carpinteiro. Rio de Janeiro: Campus, 1988. 304 p

HAYKIN, S. Neural networks, a comprehensive foundation. New Jersey: Englewood Cliffs, 1999. 842p.

HAYKIN, Simon. Redes neurais: princípios e prática. 2.ed. Porto Alegre: Bookman, 2001.

HUBNER, Jomi Fred. Árvores de Decisão. (2002). Disponível em <http://dsc.inf.furb.br/arquivos/tccs/monografias/2003-1emersontenfenvf.pdf>. Acesso em: 21 mar. 2015.

IBM. Mineração de dados com o WEKA, Parte 2: Classificação e armazenamento em cluster. Disponível em http://www.ibm.com/developerworks/br/opensource/library/os-weka2/. Acesso em: maio, 2015.

JENKINS, Henry. Cultura da Convergência. São Paulo, Aleph, 2008.

JIMENEZ, Antonio Garcia et al. Web social y organización del conocimiento: algunas claves. In: 9th Congress of the Spanish International Society for Knowledge Organization, 2009, Valência. p. 1028-1042

M86. Site Oficial. Disponível em: <http://m86security.com> Acesso em: 18 fev. 2015.

M86. Web Filtering and Reporting Suite. Disponível em: <http://www.m86security.com/products/web_security/m86-web-filtering-reporting-suite.asp> Acesso em: 18 fev. 2015.

MANOVICH, L. The Language of the New Media. Cambridge, Massachusetts: MIT Press, 2001. Uma experiência de tradução desta obra em português está disponível em http://www.fafich.ufmg.br/~novamidia.

MARTELETO, Regina Maria; SILVA, Antônio Braz de Oliveira e. Redes e capital social: o enfoque da informação para o desenvolvimento local. Ciência da Informação, Brasília, v. 33, n. 3, p. 41-49, set.dez. 2004. Disponível em: <http://revista.ibict.br/index.php/ciinf/article/view/518/472>. Acesso em: 22 fev. 2015.

MATHES, Adam. Folksonomies: cooperative classification and communication through shared metadata. 2004. Disponível em: <http://adammathes.com/academic/computer-mediated-communication/folksonomies.html> Acesso em: 26 jan. 2015.

MERHOLZ, Peter. Metadata for the masses. Disponível em: <http://www.adaptivepath.com/publications/essays/archives/000361.php>. Acesso em: 25 fev. 2015.

MIKA, Peer. Social Networks and the Semantic Web. Disponível em <wrg.upf.edu/WRG/seminarios/260407_aMika.pdf> Acesso em: 18 abr. 2015.

MOURA, Maria Aparecida. Folksonomias, redes sociais e a formação para o *tagging* literacy: desafios para a organização da informação em ambientes colaborativos virtuais. Informação e Informação, Londrina, v.14, n. especial, 2009.

NASCIMENTO, Geysa F.C. De Lima. Folksonomia como Estratégia de Indexação dos Bibliotecários no Del.icio.us. João Pessoa: 2008. Projeto de Trabalho de Conclusão de Curso de Pós-Graduação em Ciência da Informação.

NOGUEZ, J. H. S. Um estudo sobre a implementação de Árvore de Decisão com Múltiplos Classificadores. Porto Alegre, 2000. Trabalho apresentado na Semana Acadêmica 2000, Universidade Federal do Rio Grande do Sul.

NORUZI, Alireza. Folksonomy: Why do we need controlled vocabulary? E- prints in Library and Information Science. p.7, 2006. Disponível em: <http://eprints.rclis.org/archive/00011286/ . > Acesso em: 28 fev. 2015.

O´REILLY, Tim. What is web 2.0?\2005. Disponível em: <http://oreilly.com/web2/archive/what-is-web-20.html> Acesso em: 19 jan. 2015.

OSÓRIO, Fernando Santos (1991), "Um estudo sobre o reconhecimento visual de caracteres através de redes neurais", In: Dissertação submetida à Universidade Federal do Rio Grande do Sul para obtenção do Grau de Mestre em Computação, Brasil.

PAULA, M. B. Indução automática de árvore de decisão. 85 f. Dissertação da Universidade Federal de Santa Catarina – UFSC, Florianópolis - SC, 2002.

PEREIRA, Débora de Carvalho. Amazônia transnacional: as redes ambientais na internet e a padronização da natureza. Dissertação de Mestrado em Extensão Rural da Universidade Federal de Viçosa, março de 2007.

PEREIRA, Henrique C. Folksonomia e a maneira com que nós colocamos ordem nas coisas. Disponível em: <http://www.revolucao.etc.br/archives/folksnomia-e-a-maneira-com-que-nos-colocamos-ordem-nas-coisas/>. Acesso em: 05 de março de 2015.

PEREIRA, Jonathan. Surge a folksnomia, uma nova forma de classificação virtual. Disponível em: <http://listas.ibict.br/pipermail/bib_virtual/2006-May/002832.html>. Acesso em: 05 de março de 2015.

PICKLER, Maria Elisa Valentim. Web semântica: ontologias como ferramentas de representação do conhecimento. Perspect. ciênc. inf. 2007, vol.12, n.1, p. 65-83. Disponível em: <http://www.scielo.br/pdf/pci/v12n1/05.pdf>. Acesso em: 15 fev. 2015.

PINTO, F.; Santos M. F. "Descoberta de Conhecimento em Bases de Dados em Atividades de CRM"; Datagadgets 2005; 1º Congresso Espanhol de Informática CEDI 2005; Granada.

QUILAN, J. R. Comparing connectionnist and symbolic methods. Australia. University od Sydney, 1994. Disponível em: <http://citeseer.nj.nec.com/quinlan94comparing.html>. Acesso em: 03 mar. 2015.

QUILAN, J. Ross. C4.5: programs for Machine Learning. San Mateo: Morgan Kaufmann, 1993. 302p.

QUINTARELLI, Emanuele. Folksonomies: power to the people. In: INCONTRO ISKO ITALIA - UNIMIB, Milão, 2005. Papers...Milan: Universitá di Milano, 2005. Disponível em: <http://www.iskoi.org/doc/folksonomies.htm>. Acesso em: 10 fev. 2015.

RECUERO, Raquel da Cunha. Comunidades virtuais: uma abordagem teórica. In: SEMINÁRIO INTERNACIONAL DE COMUNICAÇÃO, 5, 2001. Disponível em: <http://www.bocc.ubi.pt/pag/recuero-raquel-comunidades-virtuais.pdf>. Acesso em: 13 fev. 2015.

RIGO, Sandro José. Introdução à educação a distância. São Leopoldo: Unisinos, 2010.

RODRIGUES, A. M. Escavando Dados no Varejo. Centro de Estudos em Logística - COPPEAD - Universidade Federal do Rio de Janeiro, 2000.

SANDHU, R. S; SAMARATI, P. Acecess Control: principles and pratice. IEEE Communications Magazine, p. 40-48, set. 1994.

SHANON, C.; WEAVER, W. The mathematical theory of communication. Urbana: University of Illinois Press, 1949.

SILVA, M. P. S. Mineração de Dados: Conceitos, Aplicações e Experimentos com Weka. Livro da Escola Regional de Informática Rio de Janeiro - Espírito Santo. Porto Alegre: Sociedade Brasileira de Computação, v. 1, p. 1-20, 2004.

SOUSA, M. S. R. Mineração de Dados: Uma Implementação Fortemente Acoplada a um Sistema Gerenciador de Banco de Dados Paralelo. Rio de Janeiro, 1998, 67 p. Dissertação (Mestrado em Engenharia de Sistemas e Computação). Universidade Federal do Rio de Janeiro.

SPITERI, Louise F. Structure and form of folksonomy *tag*s. The road to the public library cataloque. Webology, v. 4, n. 2, 2007. Disponível em: < http://webology.ir/2007/v4n2/a41.html >. Acesso em: 25 fev. 2015

STEINER, M. T. A., Nievola, J. C., Soma, N. Y., Shimizu, T., Steiner Neto, P. J. Extração de regras de classificação a partir de redes neurais para auxílio à tomada de decisão na concessão de crédito bancário. Pesquisa Operacional, v.

27, n. 3, p. 407-426, 2007.

TAFNET, M.A. Redes Neurais Artificiais: Aprendizado e Parilidade. Revista Cérebro e Mente, Universidade Estadual de Campinas. mar/maio 1998.

TAN, P.-N., STEINBACH, M., & KUMAR, V. (2005). Introduction to Data Mining. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

VANDER WAL, Thomas. Folksonomia coinage and definition. 2007.Disponível em: <http://vanderwal.net/folksonomy.html> Acesso em: 24 jan. 2015.

WAL, Thomas Vander. Folksonomy definition and wikipedia. Disponível em: <http://www.vanderwal.net/random/entrysel.php?blog=1750>. Acesso em: 24 jan. 2015.

WEBSENSE. Disponível em: <http://websense.com > Acesso em: 20 jan. 2015.

WEBSENSE. Websense Web Filter. Disponível em: <http://www.websense.com/content/WebFilter.aspx> Acesso em: 20 jan. 2015.

WITTEN, I. H.; FRANK, E. Data mining: practical machine learning tools and techniques. 2 ed. San Francisco: Morgan Kaufmann Publishers, 2005.

WITTEN, I. H.; FRANK, E. Data Mining: Pratical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers. San Francisco, California, 2000.

YAHOO. Disponível em:<http://yahoo.com> Acesso em: 26 jan. 2015.

YOUTUBE. Disponível em:<http://youtube.com> Acesso em: 26 jan. 2015.

YUNTA, Luis Rodríguez. Etiquetado libre frente a lenguajes documentales: aportaciones en el ámbito de Biblioteconomía y Documentación. In: 9th Congress of the Spanish International Society for Knowledge Organization, 2009, Valencia. p. 832-845.

APÊNDICE A

Para o desenvolvimento deste trabalho foram desenvolvidos cinco *scripts* com o objetivo de executar determinadas tarefas. A seguir, são apresentados os códigos fontes dos respectivos *scripts*.

1. **Conexao.php**

```
<?php
$host = "NOTO";
$db = "Thomas";
//$user = "adm\thomas;
//$password = "teste";

$conninfo = array("Database"=>$db);
$conn = sqlsrv_connect($host, $conninfo) or
die(print_r(sqlsrv_errors(),true));

/*$sql = "select COUNT(*) as total from FolkUrl";
$result = sqlsrv_query($conn, $sql);
$dados = sqlsrv_fetch_array($result);
echo "Total: " . $dados["total"];*/
?>
```

2. **Load_tags.php**

```
<?php
header("Content-Type: text/html; charset=utf-8");
include "conexao.php";
include "UrlDao.class.php";
include "Url.class.php";
include "Delicious.class.php";

# Lendo as URLs enviadas do Banco de Dados
//$urls = $_POST["urls"];

$listaUrls = UrlDao::getUrls($conn);
foreach ($listaUrls as $url) {
        $xmlDelicious = UrlDao::getXmlByUrlFromDelicious($url->getUrl());
        $listaTagsPopulares =
UrlDao::getTagsPopularesByXmlFromDelicious($xmlDelicious);
        $listaTagsRecomendadas =
UrlDao::getTagsRecomendadasByXmlFromDelicious($xmlDelicious);
        $listaTags = array_merge($listaTagsPopulares,$listaTagsRecomendadas);

        UrlDao::saveTags($conn, $url->getCodUrl(), $url->getUrl(),$listaTags);
```

```
}
UrlDao::getTags($conn);

//print_r($listaUrls);
# Para cada url é necessário descobrir as tags no Delicious
#
?>
```

3. Delicious.class.php

```
<?php
#    Fonte:    http://www.geek-alpha.com/php/using-php-connect-delicious-api-
classes
class Delicious {
     private $curl;
    private $curl_options; //options that will be used for curl
    private $username; //delicious username
    private $password; //delicious password

public function __construct($username, $password){
     $this->username = $username;
    $this->password = $password;
    $this->curl = curl_init();
}

public function set_options(){
    $this->curl_options = array(
        CURLOPT_RETURNTRANSFER => 1, //return the response from the delicious
API.
        CURLOPT_SSL_VERIFYPEER => false, //disable verification of the peer's
certificate
        CURLOPT_USERPWD => $this->username . ':' . $this->password, //set the
username and password to be used for authentication
      //CURLOPT_HTTPHEADER      =>      array("Content-Type:      text/xml;
charset=utf8","Expect: 100-continue")
    );
}
public function execute(){
    curl_setopt_array($this->curl, $this->curl_options);

    $response = curl_exec($this->curl);
    if(!$response){
      //terminate the execution of the script if there's no response
        die('Error:  "'  .  curl_error($this->curl)  .  '"  -  Code:  '  .
curl_errno($this->curl));
```

```
            }
            curl_close($this->curl); //close the connection
            return $response;
      }
      public function get_all(){

            $this->set_options();
            $this->curl_options[CURLOPT_URL]                                    =
'https://api.del.icio.us/v1/posts/all';
            $response = $this->execute();
            $xml = simplexml_load_string($response); //converts the string response
into an xml object
            return $xml;
      }
      public function get_url($url) {
            $this->set_options();
            $this->curl_options[CURLOPT_URL]=
'http://api.del.icio.us/v1/posts/suggest?url=' . $url;
            $response = $this->execute();
            $xml = simplexml_load_string($response);
            return $xml;
      }
      }
```

4. Url.class.php

```
      <?php
      class Url {
            private $codUrl;
            private $linkUrl;

                  public function __construct($aux1, $aux2) {
                  $this->setCodUrl($aux1);
                  $this->setUrl($aux2);
            }
            public function setCodUrl($aux) {
                  $this->codUrl = $aux;
            }
                  public function getCodUrl() {
                  return $this->codUrl;
            }
                  public function setUrl($aux) {
                  $this->linkUrl = $aux;
            }
                  public function getUrl() {
                  return $this->linkUrl;
            }
```

```
}
?>
```

5. UrlDao.class.php

```
<?php
class UrlDao {

	static public function getUrls($conn) {
		$sql = "select cod, url from FolkUrl";
		$result = sqlsrv_query($conn, $sql);
		$listaUrl = array();
		while ($dados = sqlsrv_fetch_array($result))
			array_push($listaUrl,new
Url($dados["cod"],$dados["url"]));
		return $listaUrl;
	}

	static public function getXmlByUrlFromDelicious($auxUrl) {
		$delicious = new Delicious("juliano.varella","12345678");
		$retorno = $delicious->get_url($auxUrl);
		return $retorno;
	}

	static public function getTagsPopularesByXmlFromDelicious($auxXml) {
		$listaTagsPopulares = array();

		foreach ($auxXml->popular as $popular) {
			array_push($listaTagsPopulares, $popular["tag"]);
		}
		return $listaTagsPopulares;
	}

	static public function getTagsRecomendadasByXmlFromDelicious($auxXml)
{
		$listaTagsRecomendadas = array();

		foreach ($auxXml->recommended as $recomendada) {
			array_push($listaTagsRecomendadas, $recomendada["tag"]);
		}
		return $listaTagsRecomendadas;
	}

	static public function saveTags($conn, $auxCodUrl, $auxUrl,
$auxListaTags) {
		date_default_timezone_set("UTC");
```

```
                $sql = "update FolkUrl set ano = " . date("Y") . ", mes = " .
date("m");

                $sqlTags = "";
                $i = 1;
                foreach ($auxListaTags as $tag)
                        $sqlTags .= ", tag" . $i++ . "='" . $tag . "'"; //
collate SQL_Latin1_General_CP1_CI_AI

                $sql .= $sqlTags . " where cod = " . $auxCodUrl;

                //echo "SQL: $sql . <br />";
                //echo $sql; die;
                $result = sqlsrv_query($conn, $sql);
                //echo "Pronto! <br /><br />";
        }

        static public function getTags($conn) {
                date_default_timezone_set("UTC");
                $sql = "select tag2 from FolkUrl";
                //echo "SQL: $sql . <br />";
                //echo $sql; die;
                $result = sqlsrv_query($conn, $sql);
                while ($dados = sqlsrv_fetch_array($result))
                        echo $dados["tag2"] . "<br/>";
                //echo "Pronto! <br /><br />";
        }

}
?>
```