UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

VINÍCIUS FERNANDES SCHMITT

UMA ANÁLISE COMPARATIVA DE TÉCNICAS DE APRENDIZAGEM DE MÁQUINA PARA PREVER A POPULARIDADE DE POSTAGENS NO FACEBOOK

Prof. Dr. Dante Augusto Couto Barone
Orientador

Porto Alegre, Dezembro de 2013

AGRADECIMENTOS

Primeiramente agradeço aos meus pais, José e Rosane, que sempre tentaram me guiar e incentivar em todos os momentos difíceis dessa caminhada. Agradeço, à meu irmão, Guilherme, pelo apoio e companheirismo.

Agradeço ao professor Dr. Dante Augusto Couto Barone pela orientação, oportunidade e paciência.

Agradeço também aos colegas de curso que partilharam comigo alguma parcela deste tempo como aluno do curso de Bacharelado em Ciência da Computação.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

KDD	Knowledge Discovery in Databases
DCBD	Descoberta de Conhecimento em Base de Dados
AM	Aprendizado de Máquina
SVM	Supporting Vector Machine
SMO	Sequential minimal optimization
WEKA	Waikato Environment for Knowledge Analysis
TP	True Positives (Positivos Verdadeiros)
TN	True Negatives (Negativos Verdadeiros)
FP	False Positives (Falsos Positivos)
FN	False Negatives (Falsos Negativos)

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

As Redes sociais ganharam muita popularidade nos últimos anos. Isto deu origem à disciplina emergente de Social Media Analytics, a qual está diretamente ligada à diversas outras áreas como Análise de Redes Sociais, Aprendizado de Máquina, Mineração de Dados, Recuperação de Informação e Processamento de Língua Natural.

Este trabalho faz uso de uma página de fãs do *Facebook* como base para avaliar o desempenho de três classificadores de Aprendizado de Máquina supervisionados (o Naive Bayes, Sequential Minimal Optimization e C4.5) com a tarefa de classificar a popularidade de postagens.

Além de analisar o comportamento desses algoritmos, o trabalho tem como objetivo testar diferentes técnicas de pré-processamento de texto e verificar qual combinação produz melhores resultados no processo de gerar um modelo de aprendizado para prever a popularidade de postagens nessa rede social.

Palavras-Chave: Mineração de Dados, Classificadores, Aprendizado de Máquina, Redes Sociais.

A COMPARATIVE ANALYSIS OF MACHINE LEARNING TECHNIQUES TO PREDICT THE POPULARITY OF POSTS ON FACEBOOK

ABSTRACT

Online Social Networks have gained increased popularity in recent years. This has given rise to the emerging discipline of Social Media Analytics, which is directly connected to some other areas such as: Social Network Analysis, Machine Learning, Data Mining, Information Retrieval, and Natural Language Processing.

This work makes use of one fan page on Facebook as dataset to evaluate the performance of three supervised machine learning classifiers (Naive Bayes, Sequential Minimal Optimization and C4.5) when classifying the popularity of posts.

Besides analysing how these algorithms behave, this study has also as objective to test different types of text preprocessing techniques and to verify which combination can provide better results in the process of generating a learning model to predict the popularity of posts on this social network.

Keywords: Data Mining, Classifiers, Machine Learning, Social Networks

1 INTRODUÇÃO

As seções a seguir apresentam uma visão geral sobre os tópicos básicos necessários para a compreensão deste documento assim como os objetivos do trabalho.

1.1 Classificação de Dados

A tarefa de classificação é uma função de aprendizado que mapeia dados de entrada, ou conjuntos de dados de entrada, em um número finito de categorias. Nela, cada exemplo pertence a uma classe, entre um conjunto predefinido de classes (GOLDSCHMIDT; PASSOS, 2005).

Cada classe corresponde a um padrão único de valores dos atributos previsores (demais atributos que caracterizam o exemplo). Esse padrão único pode ser considerado a descrição da classe. O conjunto de todas as classes é definido como C e, a cada classe C_i, correspondem uma descrição D_i das propriedades selecionadas. Dessa forma, usando essas descrições é possível construir um classificador o qual descreve um exemplo e do conjunto de exemplos T como sendo um exemplo pertencendo à classe C_i, quando aquele exemplo satisfaz D_i (CARVALHO,2005).

O principal objetivo da construção de um classificador é descobrir algum tipo de relação entre os atributos previsores e as classes (FREITAS; LAVINGTON,1998). Por exemplo, na figura 1.1 o classificador em questão tem como objetivo a identificação da relação existente entre os atributos previsores A1 e A2 e os valores da classe (“+” e “-”). O procedimento de construção deste classificador é baseado em particionamentos recursivos do espaço de dados. O espaço é dividido em áreas e a cada estágio é avaliado se cada área deve ser dividida em subáreas, a fim de obter uma separação das classes.

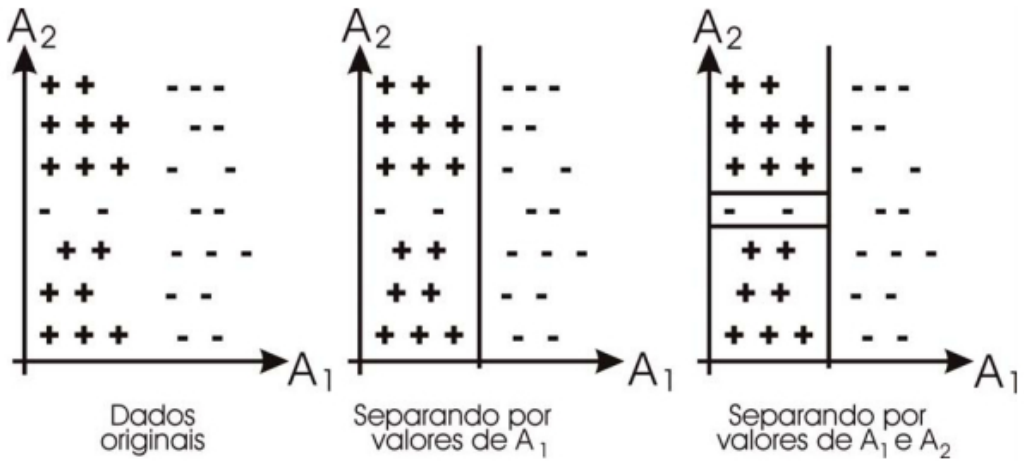

Figura 1.1 Exemplo de Classificação (FREITAS; LAVINGTON,1998)

Um importante conceito da tarefa de classificação é a divisão dos dados entre conjunto de treinamento e conjunto de teste. Inicialmente, um conjunto de treinamento é disponibilizado e analisado, e um modelo de classificação é construído baseado nesses dados. Então o modelo construído é utilizado para classificar outros dados, chamados dados de teste, os quais não foram contemplados pelo algoritmo durante a fase de treinamento. Cabe ressaltar que o modelo construído a partir dos dados de treinamento só será considerado um bom modelo, do ponto de

vista de precisão preditiva, se ele classificar corretamente uma alta porcentagem dos exemplos (registros) dos dados de teste. Em outras palavras, o modelo deve representar conhecimento que possa ser generalizado para dados de teste, não utilizados durante o treinamento (CARVALHO, 2005).

Existem diferentes técnicas para a realização dessa tarefa, como Redes Neurais, Algoritmos Genéticos, Métodos Bayesianos, Árvores de Decisão, Máquinas de Vetor de Suporte entre outras (CARVALHO; DIAS, 2008). Dentre estas, utiliza-se neste trabalho Árvores de Decisão (através do algoritmo C4.5), Naive Bayes e Máquinas de Vetor de Suporte (através do algoritmo SMO – Sequential Minimal Optimization).

1.2 Motivação e Objetivos

Este trabalho utiliza uma página de fans do *Facebook*[1] como base para avaliar o desempenho de três classificadores de aprendizado de máquina supervisionados na tarefa de classificar a popularidade de postagens (considerando o conteúdo textual das postagens).

Diferentemente de alguns dos trabalhos realizados na área de análise de sentimento (KAYA; FİDAN; TOROSLU, 2012) (PANG; LEE; VAITHYANATHAN, 2002) em que se realizou uma análise semântica para minerar opiniões, este trabalho levou em consideração o *feedback* (neste caso número de *likes*) provenientes dos usuários de uma rede social para determinar as classes no processo de classificação.

Os métodos de classificação escolhidos foram o Naive Bayes que modela cada distribuição condicional com uma simples função Gaussiana (JOHN & LANGLEY, 1995), Support Vector Machines (SVM, 2013), ou Máquina de Vetor Suporte, que mapeia no espaço n-dimensional os pontos mais próximos à superfície de separação dos conjuntos que representam cada classe (VAPNIK, 1995) e um classificador baseado em árvores de decisão, o C4.5 (CARVALHO, 2005).

Foram escolhidos especificamente esses algorítmos pelo fato deles estarem entre os algoritmos de classificação mais influentes utilizados pela comunidade de pesquisa na área de classificação (WU et al., 2007).

As coleções (denominadas *corpus*) utilizadas foram um conjunto de textos extraídos de postagens da uma *fan page*. São 4021 mensagens classificadas em dois conjuntos: com duas classes (popularidades “boa” e “ruim”) e três classes (popularidades “boa”, “média” e “ruim”) confome o número de *likes* que elas obtiveram.

O objetivo geral deste trabalho é avaliar como se comportam diferentes classificadores quando utilizadas técnicas de mineração de dados para predição da popularidade das postagens de uma página do Facebook.

Os objetivos específicos para a realização deste trabalho são:

- Realizar a coleta e o armazenamento de dados (texto das postagens e o seus respectivos números de *Likes*) para realização de testes e validações;

[1] Facebook. Disponível em: http://www.facebook .com

- Realizar o pré-processamento dos dados a fim de avaliar o desempenho do modelo para diferentes dados com diferentes características;
- Avaliar o uso dos classificadores nesse tipo de abordagem (usando o fator de número de Likes como delimitador das classes).
- Analisar os resultados a fim de comparar o desempenho dos algoritmos de classificação escolhidos.

1.3 Estrutura do Texto

A monografia está estruturada em seis capítulos, incluindo a introdução. Os outros cinco capítulos são descritos a seguir.

O capítulo dois apresenta alguns conceitos básicos sobre as redes sociais assim como algumas características da maior rede social da atualidade: o *Facebook*.

O terceiro capítulo explora alguns conceitos de mineração de dados e aprendizado de máquina. Também são apresentados os algoritmos de classificação escolhidos para realização dos testes.

O quarto capítulo apresenta toda metodologia do trabalho: desde a captura de dados, armazenamento, classificação dos dados, pré-processamento e por fim a geração e validação dos modelos preditivos.

No capítulo cinco são apresentados os resultados obtidos para ambos conjuntos de testes e também é feita uma análise sobre esses resultados.

Ao final do documento, no capítulo seis, são apresentadas as conclusões deste trabalho juntamente com sugestões para trabalhos futuros e as referências utilizadas na realização do mesmo.

2 REDES SOCIAIS

O objetivo deste capítulo é apresentar uma visão geral de redes sociais: suas origems, aplicações que hoje em dia fazem uso desse conceito e algumas definições do ponto de vista teórico sobre esse tipo de rede.

2.1 As Redes Sociais Digitais

Em se tratando da redes sócias digitais, o ClassMates.com, criado em 1995, é considerado a primeira rede social. Acessado nos Estados Unidos e no Canadá, o objetivo do site era possibilitar reencontros entre amigos que estudaram juntos, seja no colégio ou na faculdade. O serviço era pago, porém conseguiu fazer sucesso e está on-line até hoje (RBCCV, 2013).

Desde então, surgiram diversas redes sociais. As principais delas são: o *Facebook* (2004), o *QZone*[2] (2005) , *Twitter*[3] (2006), e o *Google+*[4] (2011). De uma maneira geral, o objetivo de todas essas redes sempre foi o mesmo: promover a interação entre aqueles que participam, e o compartilhamento de interesses através da criação de comunidades e grupos. Todos se comunicam com interatividade e troca de conteúdo. A figura 2.1 apresenta as maiores redes socias da atualidade (STATISTA,2013).

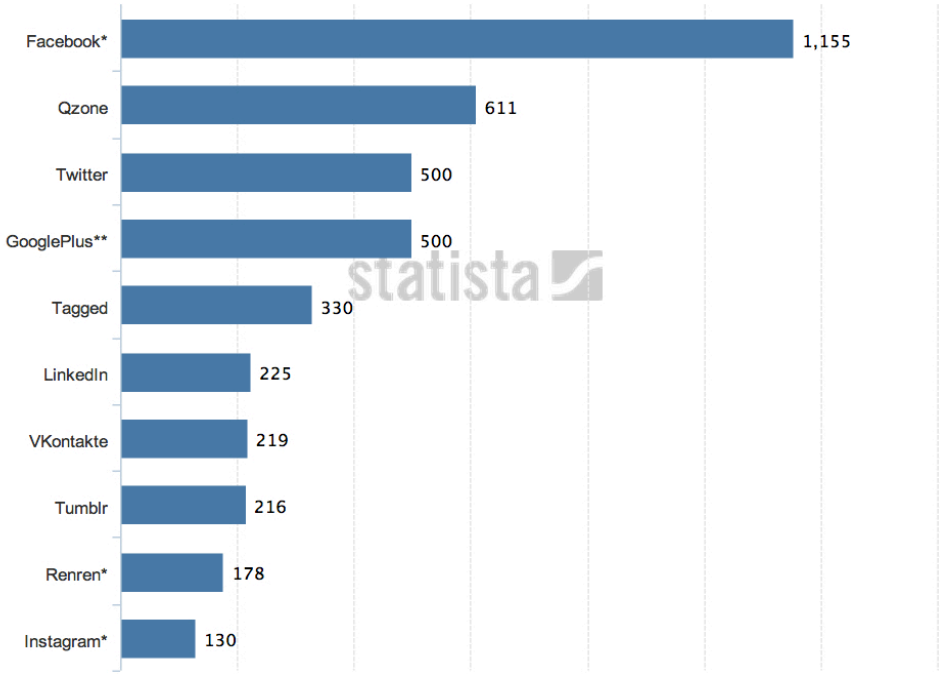

Figura 2.1 Gráfico do número de usuários registrados nas redes sociais (em milhões) (STATISTA, 2013)

[2] Disponível em: http://qzone.qq.com
[3] Disponível em: http://twitter.com
[4] Disponível em: https://plus.google.com

Durante os últimos anos, as redes sociais digitais ganharam um destaque considerável na comunidade científica. Esse tipo de estrutura passou a fazer parte das aplicações mais utilizadas desde então, as redes de relacionamento. Sites como Facebook, Google+ e Twitter passaram a fazer parte da vida das pessoas. Um exemplo prático é a rede de colaboração de pesquisadores. Nela têm-se pesquisadores colaborando entre si e formando uma espécie de rede de contatos, na qual os indivíduos da rede são os pesquisadores, e os relacionamentos são os trabalhos desenvolvidos em cooperação (CERVI, 2008).

Do ponto de vista computacional, uma rede social é um grafo, orientado ou não, que mapeia uma realidade ou um mundo restrito, no qual os nodos representam as entidades (indivíduos ou classes de indivíduos – também chamados atores) e as arestas representam os relacionamentos entre essas entidades (EMIRBAYER; GOODWIN, 1994). Os relacionamentos podem ser o compartilhamento de um ou mais atributos. A realidade representada pelas redes sociais são fontes de dados heterogêneos e multi-relacionais, cujos relacionamentos podem ser unidirecionais e não necessariamente precisam ser binários.

De acordo com (HAN; KAMBER, 2006), define-se que, do ponto de vista de implementação, uma rede social é um conjunto de dados heterogêneos e multirelacionais representados por um grafo. O grafo é tipicamente muito grande, com nós correspondendo os objetos e as arestas correspondendo as ligações que representam relacionamentos ou interações entre os objetos. Os objetos e as ligações possuem atributos, sendo que os objetos podem ter rótulos de classe e as ligações podem ser unidirecionais.

2.2 O Facebook

Atualmente, a maior das redes sociais existentes é o *Facebook*, fundado em 2004 pelos ex-estudantes da Universidade de Harvard Mark Zuckerber, Dustin Moskovitz, Eduardo Saverin e Cris Hughes.

No seu inicio funcionaria de forma restrita somente para os estudantes de Harvard. Com o passar do tempo foi expandindo para outros campos estudantis e somente em 2006, qualquer usuário com mais de 13 anos poderia criar o seu perfil no Facebook.

O Facebook alcançou a marca de mais de 1 bilhão de pessoas conectadas em 2013. Tornou-se a rede social mais frequentada do mundo, tendo o dobro das que estão como segundas colocadas, o Google+ e o Twitter, com 500 milhões cada.

Os usuários Facebook já postaram mais de 250 bilhões de fotos na rede social e continuam a publicar cerca de 350 milhões de novas imagens diariamente. O número representa uma média de 217 fotos de cada um dos 1,15 bilhão de usuários do site.

O número de publicações compartilhadas também é expressivo: 4,75 bilhões por dia – incluindo compartilhamento de fotos, status, vídeos e comentários. O botão "Curtir" é clicado 4,5 bilhões de vezes a cada 24 horas e mais de 10 bilhões de mensagens são enviadas em apenas um dia (TECH, 2013).

A figura 2.1 mostra o crescente número de usuários ativos por quadrimestre nessa rede social, desde março de 2009 até setembro de 2013.

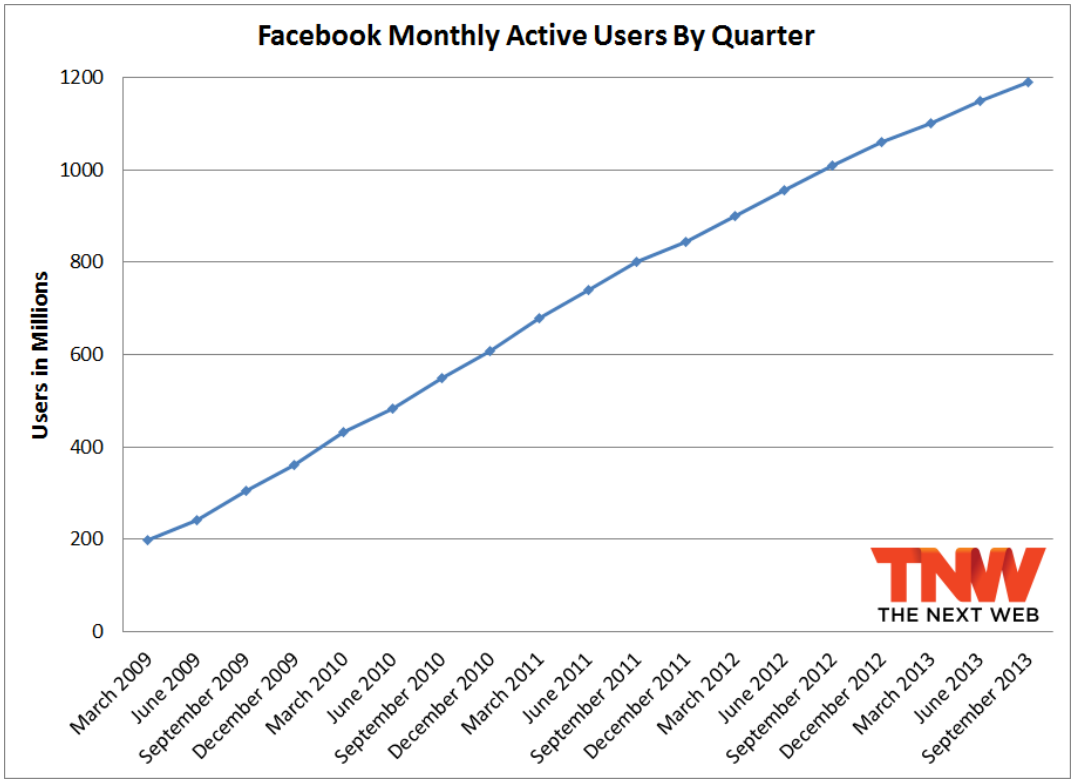

Figura 2.2 Gráfico do número de usuários ativos por quadrimestre no Facebook entre 03/2009 até 09/2013 (GIZ, 2013)

2.2.1 Página de Fãs

As páginas de fãs (*fan pages*) existem para que as organizações, empresas, celebridades e transmitam informações ao seus seguidores ou ao público que escolher se conectar a elas. Semelhante aos perfis, esse tipo de páginas podem ser aprimoradas com aplicativos que ajudem as entidades a se comunicarem e interagirem com o seu público e adquirirem novos usuários por recomendações de amigos, históricos dos Feeds de notícias e eventos.

Grande parte do crescimento da rede social Facebook acontece, sem dúvida, por conta do relacionamento entre os consumidores e as marcas, proporcionado através das *fan pages*.

Neste trabalho, a página de fãs da Azaléia[5] foi utilizada como objeto para o estudo de como diferentes técnicas de Mineração de dados afetam algoritmos de classificação de textos na predição da popularidade das postagens dessa página. A figura 2.2 ilustra a página utilizada nesse trabalho.

[5] Disponível em http://www.facebook.com/azaleiaoficial

Figura 2.3: Página de fãs da Azaléia

2.2.2 Postagens

Uma postagem no Facebook é uma mensagem que tem um limite de 63,206 caracteres, sendo que nela é possível inserir textos, vídeos, links e imagens. Os usuários da rede social podem realizar algumas ações sobre o conteúdo dessas postagens, como: curtir, comentar e compartilhar. A figura 2.3 apresenta uma postagem da página de fãs utilizada no trabalho.

Figura 2.4 Exemplo de uma postagem da página de fãs da Azaléia

Dentre as ações disponíveis aos usuários, a opção de “curtir” (“*like”*) conteúdo foi a escolhida nesse trabalho para classificar a popularidade das postagens. Essa escolha foi feita pelo fato de que o número de *likes* das postagens é o que, em geral, apresenta o maior valor dentre as ações possíveis. A ação realizada através do botão de “curtir” é descrita na seção seguinte.

2.2.3 Botão Curtir

Em relação a origem da idéia de curtir conteúdos o *StumbleUpon*, ferramenta de busca lançada em 2001, já tinha o termo “*I like it*” e um polegar para curtir links. Em 2007, outro site, o Friendfeed, criou um botão com o mesmo nome e funcionamento do “Curtir”. A empresa foi comprada por Mark Zuckerberg em 2009.

O botão curtir ou "like button" é um plug-in social que foi adaptado ao Facebook em abril de 2010 como uma forma de as pessoas compartilharem seus interesses em conteúdo fora do Facebook (artigos, vídeos, produtos etc.) e oferecer recomendações para seus amigos na própria rede social.

Um botão *like* é um recurso de softwares de comunicação, como os serviços de redes sociais, fóruns na Internet, sites de notícias e blogs onde o usuário pode expressar se ele gosta ou apoia um conteúdo. Serviços de Internet que apresentam botões de “curtir” geralmente indicam a quantidade de usuários que gostaram de cada conteúdo, e pode mostrar uma lista completa ou parcial deles.

Nesse trabalho, a informação do número de *likes* das postagens capturadas do Facebook será usada como fator determinante na etapa de classificação das postagens.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados diversos aspectos básicos sobre Descoberta de Conhecimento em Base de Dados (KDD) e a ferramenta de KDD utilizada no trabalho, Mineração de dados e Aprendizado de Máquina.

Em seguida, são apresentados alguns algoritmos de Aprendizado de Máquina usados comumente para classificação de dados e que foram utilizados nesse trabalho: o Naive Bayes, Máquinas de Vetor de Suporte ou SVM (e sua variação o *Sequential Minimal Optimization* - SMO) e as Árvores de Decisão (em especial, o algoritmo C4.5).

3.1 KDD e Mineração de Dados

Os termos Mineração de Dados (Data Mining) e Descoberta de Conhecimento em Base de Dados (Knowledge Discovery in Databases – KDD) ou DCBD muitas vezes são confundidos como sinônimos para identificar o processo de descoberta de conhecimento útil a partir de bancos de dados. O termo KDD foi estabelecido no primeiro workshop de KDD em 1989 para enfatizar que conhecimento é o produto final de uma descoberta baseada em dados (*data-driven*). Desta forma KDD se refere a todo o processo de descoberta de conhecimento enquanto Data Mining se refere a uma das etapas deste processo.

A DCBD apresenta caráter multidisciplinar, já que envolve a participação de outras áreas de pesquisa como Aprendizagem de Máquina, Estatística, Banco de Dados, Sistemas Inteligentes e Visualização de Dados, como visualizado na Figura 3.1.

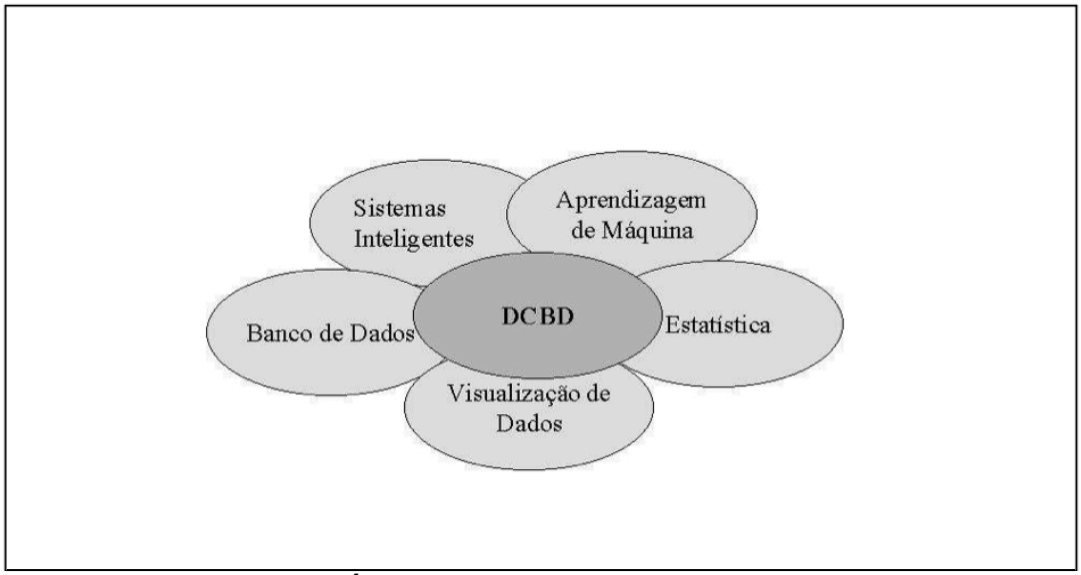

Figura 3.1 Áreas de Apoio a DCBD (NEVES, 2003)

O processo de descoberta de conhecimento em bases de dados pode ser descrito através de três etapas: o pré-processamento, a Mineração de Dados (ou extração de padrões) e o pós-processamento (GOLDSCHMIDT; PASSOS, 2005).

A etapa de pré-processamento compreende a escolha do conjunto de dados visando:

- realizar a extração de padrões;
- conhecer quais valores são válidos para os atributos, seus tipos e formatação;
- determinar quais são os atributos mais relevantes,
- determinar as relações entre os atributos quando estes estiverem contidos em mais de uma tabela;
- realizar a “limpeza” dos dados, removendo registros incompletos ou corrompidos;
- reduzir o número de dados, caso não haja espaço disponível suficiente ou o algoritmo de mineração utilizado apresente alguma limitação.Neste caso deve- se tomar o cuidado de não reduzi-los excessivamente, a ponto de comprometer a precisão e a qualidade do resultado.

Nesta fase, também pode ser feita a discretização - substituição de atributos contínuos ou com uma grande quantidade de valores distintos por outros - sem perda significativa da informação necessária para o processo de mineração de dados e interpretação posterior dos resultados.

O pré-processamento é fundamental, pois normalmente os dados disponíveis não se encontram na forma adequada para extração do conhecimento. Esta etapa envolve as tarefas de: extração e integração de dados em formatos heterogêneos; transformação para um formato legível ao algoritmo de extração de padrões; limpeza dos dados coletados, evitando informações mal formadas ou fora de intervalos aceitáveis e, finalmente, a redução dos dados, melhor aproveitando os limitados recursos computacionais (REZENDE 2003).

A próxima etapa, caracteriza-se pela extração do conhecimento da base de dados e pela escolha das atividades auxiliares para isto. Entre elas está a escolha da tarefa de Mineração de Dados e do algoritmo, ou dos algoritmos, a serem utilizados (REZENDE, 2003).

A etapa de *Data Mining* (Mineração de Dados) é o núcleo do processo, onde são aplicados os algoritmos para extrair padrões dos dados. (FAYYAD et al., 1996). Mineração de Dados é a extração de informação ou conhecimento útil nos dados. A Mineração de Dados tem como base a Estatística, a Inteligência Artificial, a Aprendizagem de Máquina, o Reconhecimento de Padrões e os Sistemas de Bases de Dados. É adequada à análise de grandes quantidades de dados, dados com alta dimensionalidade e com natureza distribuída e heterogênea (TAN et al., 2005).

Esta é uma etapa iterativa, direcionada ao cumprimento dos objetivos definidos na identificação do problema. Nesta etapa é realizada a escolha, a configuração e a execução de um ou mais algoritmos de extração do conhecimento. Uma tarefa é escolhida de acordo com os objetivos desejáveis para a solução a ser encontrada, podendo estar agrupada em atividades preditivas ou descritivas. A predição consiste na generalização de exemplos (classificação, regressão), enquanto a descrição consiste na identificação de comportamentos intrínsecos do conjunto de dados, sem possuírem classes específicas (clusterização, regras de associação, etc.) Por fim, um algoritmo é escolhido para realizar a extração de padrões (REZENDE 2003) (FAYYAD, 1998).

As principais tarefas de mineração de dados estão relacionadas à classificação, associação e agrupamento de padrões.

Na Classificação, principal foco desse trabalho, cada padrão contém um conjunto de atributos e um dos atributos é denominado classe. O objetivo da classificação é encontrar um modelo para predição da classe como função dos outros atributos (TAN et al., 2005). A regressão é um caso particular da classificação, já que seu objetivo é encontrar um modelo para predição de um atributo contínuo como função dos outros atributos.

Já na Associação, o objetivo é produzir regras de dependência que irão predizer a ocorrência de um atributo baseado na ocorrência de outros atributos (TAN et al., 2005).

O Agrupamento ou Segmentação (também conhecido *Clustering*) procura grupos de padrões tal que padrões em um grupo são mais similares uns aos outros e dissimilares a padrões em outros grupos (TAN et al., 2005).

Após o pré-processamento e a mineração de dados, a etapa de pós-processamento é de suma importância para que o resultado final seja obtido. Essa etapa visa propiciar que o conhecimento extraído possa ser utilizado na resolução de problemas reais (REZENDE 2003).

Esta é a fase que engloba a visualização, a análise e a interpretação das saídas geradas na etapa de Mineração de Dados. Com isso, é gerado um modelo de conhecimento, ou seja, qualquer abstração de conhecimento, expresso em alguma linguagem, que descreva algum conjunto de dados (GOLDSCHIMDT; PASSOS, 2005) (FAYYAD, 1998).

3.2 Aprendizado de Máquina

Aprendizado de Máquina (AM) é uma subárea da Inteligência Artificial concentrada em desenvolver modelos que possam ”aprender” através da experiência. O aprendizado se dá através de algoritmos dedutivos que baseados em estatística, extraem regras e padrões em grandes massas de dados. ML (do inglês, *Machine Learning*) tem sido bastante utilizado em tarefas de pré-processamento de textos como: etiquetagem morfossintática (HAYKIN,1999) e no processo de classificação automática de textos (CAMARGO,2007).

As técnicas de AM empregam um princípio de inferência denominado indução, no qual obtém-se conclusões genéricas a partir de um conjunto particular de exemplos. A indução é caracterizada pelo raciocínio originado em um conceito específico e generalizado, ou seja, da parte para o todo. Na indução, um conceito é aprendido efetuando-se inferência indutiva sobre os exemplos apresentados. Portanto, as hipóteses geradas através da inferência indutiva podem ou não preservar a verdade (REZENDE, 2003).

O aprendizado indutivo pode ser dividido em dois tipos principais: supervisionado e não-supervisionado. Na figura 3.2 é mostrada a hierarquia do aprendizado indutivo.

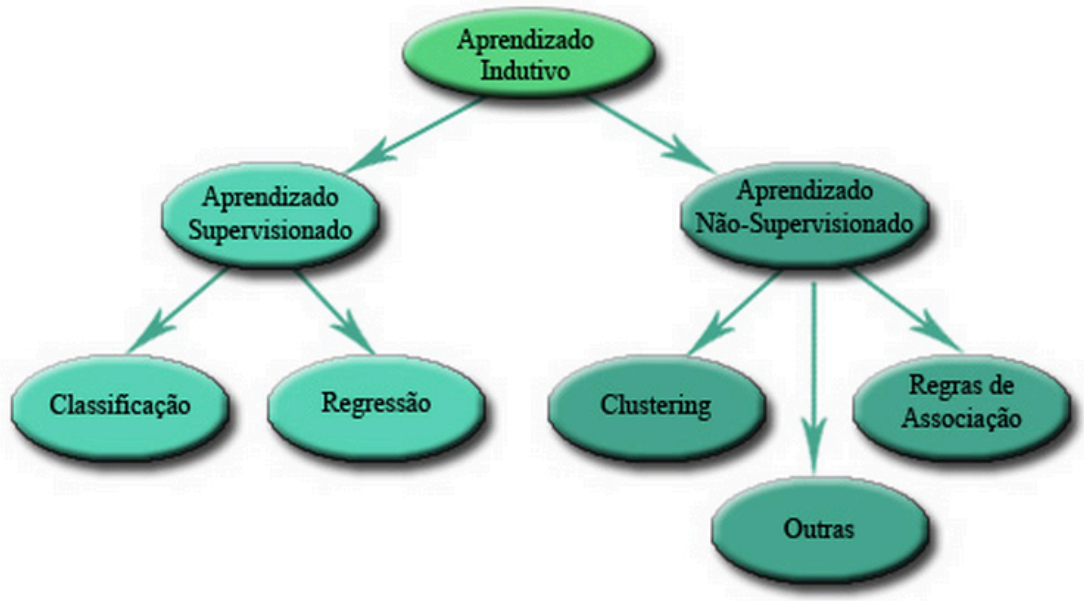

Figura 3.2 Hierarquia do aprendizado indutivo (REZENDE, 2003)

No aprendizado supervisionado tem-se a figura de um professor externo, o qual apresenta o conhecimento do ambiente por conjuntos de exemplos na forma: entrada, saída desejada (HAYKIN,1999). O algoritmo de AM extrai a representação do conhecimento a partir desses exemplos. O objetivo é que a representação gerada seja capaz de produzir saídas corretas para novas entradas não apresentadas previamente.

No aprendizado não-supervisionado não há a presença de um professor, ou seja, não existem exemplos rotulados. O algoritmo de AM aprende a representar (ou agrupar) as entradas submetidas segundo uma medida de qualidade. Essas técnicas são utilizadas principalmente quando o objetivo for encontrar padrões ou tendências que auxiliem no entendimento dos dados

O tipo de aprendizado abordado neste trabalho é o supervisionado. A figura 3.3 ilustra os conceitos referentes à geração de um classificador com esse tipo de aprendizado.

Neste caso, considerando-se um conjunto com *n* dados. Cada dado x_i possui *m* atributos, ou seja, $x_i = (x_{i1},\ldots,x_{im})$. As variáveis y_i representam as classes. A partir dos exemplos e as suas respectivas classes, o algoritmo de AM extrai um classificador. Pode-se considerar que o modelo gerado fornece uma descrição compacta dos dados fornecidos.

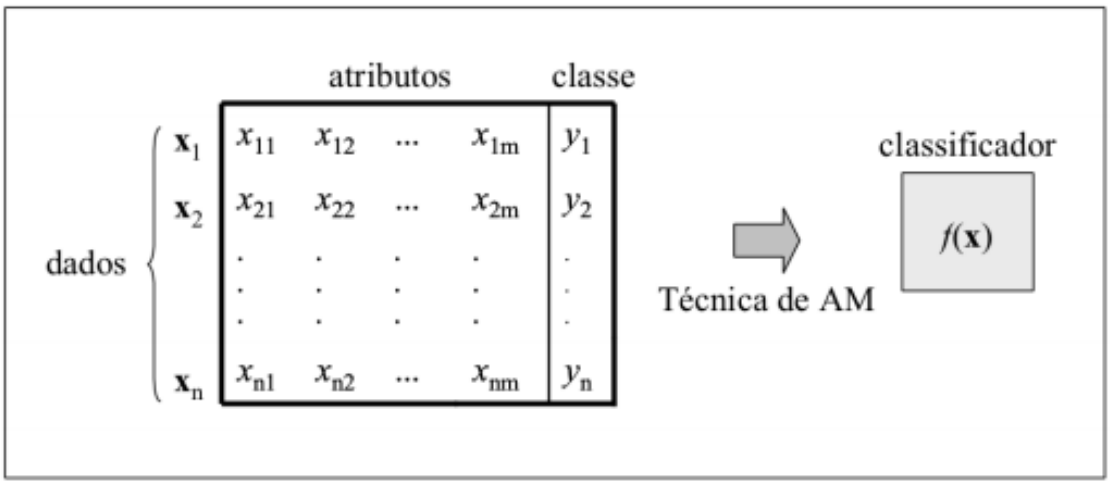

Figura 3.3 Indução de classificador em aprendizado supervisionado (LORENA; CARVALHO, 2007)

A obtenção de um classificador por um algoritmo de AM a partir de uma amostra de dados também pode ser considerada um processo de busca (MITCHELL, 1997). Procura-se, entre todas as hipóteses que o algoritmo é capaz de gerar a partir dos dados, aquela com melhor capacidadede descrever o domínio em que ocorre o aprendizado.

Abaixo seguem alguns conceitos e definições utilizados, tanto neste trabalho, e na literatura de Aprendizado de Máquina (BARANAUSKAS, MONARD, 2000):

• **Indutor**: conhecido como programa de aprendizado, ou algoritmo de indução. Tem como objetivo extrair um bom classificador a partir de um conjunto de exemplos rotulados. A saída do indutor, o classificador, pode ser usada para classificar exemplos novos (ainda não rotulados) com a meta de predizer corretamente o rótulo de cada um.

• **Exemplo**: também denominado caso, registro ou dado. É uma tupla de valores de atributos (ou um vetor de valores de atributos).

• **Atributo**: descreve uma característica ou um aspecto de um exemplo. Normalmente, há pelo menos dois tipos de atributos: nominal, quando não existe uma ordem entre os valores (por exemplo, cor: vermelho, verde, azul) e contínuo, quando existe uma ordem linear nos valores (por exemplo, peso: pertencente ao conjunto dos números reais).

• **Classe**: todo o exemplo possui um atributo especial, denominado rótulo ou classe, que descreve o fenômeno de interesse, isto é, o conceito-meta que se deseja aprender para fazer previsões a respeito.

• **Conjunto de exemplo**: um conjunto de exemplos é composto por exemplos contendo valores de atributos bem como a classe associada.

• **Classificador ou Hipótese**: dado um conjunto de exemplos de treinamento, um indutor gera como saída um classificador (também denominado hipótese ou descrição de conceito) de forma que, dado um novo exemplo, ele possa predizer com a maior precisão possível sua classe.

• **Ruído**: são dados imperfeitos, podem ser derivados do próprio processo que gerou os dados, do processo de aquisição dos dados, do processo de transformação dos dados ou mesmo devido a classes rotuladas incorretamente.

• **Underfitting e Overfitting**: Ao induzir, a partir dos exemplos dispononíveis, é possível que a hipótese seja muito específica para o conjunto de treinamento utilizado. Como o conjunto de treinamento é apenas uma amostra de todos os exemplos do domínio, é possível induzir hipóteses que melhorem seu desempenho no conjunto de treinamento, enquanto pioram o desempenho em exemplos diferentes daqueles pertencentes ao conjunto de treinamento. Nesta situação, o erro (ou outra medida) em um conjunto de teste independente evidenciaum desempenho ruim da hipótese. Neste caso, diz-se que a hipótese ajusta-se em excesso ao conjunto de treinamento ou que houve um *overfitting*. Por outro lado, também é possível que poucos exemplos representativos sejam dados aosistema de aprendizado ou o usuário pré-defina o tamanho do classificador como muito pequeno (por exemplo, um número insuficiente de neurônios e conexões são definidos em uma rede neural ou um alto fator de poda é definido para uma árvore de decisão) ou uma combinação de ambos. Portanto, é também possível induzir hipóteses que possuam uma melhora de desempenho muito pequena no conjunto de treinamento, assim como em um conjunto de teste. Neste caso, diz-se que a hipótese ajusta-se muito pouco ao conjunto de treinamento ou que houve um *underfitting*.

- **Poda**: é uma técnica de lidar com ruído e *overfitting*. A idéia geral consiste em lidar com o problema de *overfitting* através do aprendizado de uma hipótese bem genérica a partir do conjunto de treinamento de forma a melhorar o desempenho em exemplos não vistos. Há, basicamente, dois métodos de poda (1) pré-poda: durante a geração da hipótese, alguns exemplos de treinamento são deliberadamente ignorados, de forma que a hipóstese final não classifique todos os exemplos de treinamento corretamente; (2) pós-poda: inicialmente, uma hipótese que explica os exemplos de treinamento é gerada. Após isso, a hipótese é generalizada através da eliminaçãoo de algumas partes, tais como o corte de alguns ramos em uma árvore de decisão, algumas condições nas regras induzidas ou eliminando neurônios redundantes em uma rede neural.

3.3 WEKA

A ferramenta de KDD utilizada nesse trabalho foi o software WEKA[6]. O WEKA (*Waikato Environment for Knowledge Analysis*) é uma ferramenta, construída na Universidade de Waikato na Nova Zelândia, que compreende diversos algoritmos de preparação de dados, de aprendizagem de máquina (mineração) e de validação de resultados. Nela há também mecanismos para filtragem e normalização de dados, através dos chamados “filters”. A ferramenta Weka possui algoritmos que podem ser agrupados nas seguintes categorias: (EIBE et al., 2005),

- **Classificação**: nessa categoria estão implementados os mais usados e conhecidos algoritmos de classificação. Existem métodos bayesianos, árvores de decisão como ID3 (GOLDSCHMIDT; PASSOS, 2005), C4.5 (QUINLAN, 1993) e aprendizagem de regras como OneR;

[6] Disponível em http://www.cs.waikato.ac.nz/ml/weka/

- **Regressão**: estão implementados diversos algoritmos de regressão, tais como regressão linear simples e múltipla, pace regression, regressão vetorial, perceptron multicamadas, locally-weighted learning e árvores de decisão;
- **Clusterização**: apenas alguns algoritmos estão implementados, tais como, K- Means (HUANG, 1998), algoritmos EM (Estimation- Maximization), Cobweb e Farthest-First Clustering;
- **Regras de associação**: estão presentes os algoritmos Apriori, PredictiveApriori e Tertius;
- **Seleção de atributos**: são suportadas as abordagens filter e wrapper (GOLDSCHIMDT; PASSOS, 2005): Filter: Trata-se de uma abordagem onde a seleção de atributos independe do algoritmo de mineração que será aplicado aos atributos selecionados; Wrapper: Consiste em experimentar o algoritmo de mineração de dados para cada conjunto de atributos e posteriormente avaliar os resultados obtidos.

As implementações de seleção de atributos na Weka, incluem seleção baseada em correlação, estatística qui-quadrado, *gain ratio*, *information gain*, *symmetric uncertainty e vector machine-based criterion*. Existem também uma variedade de métodos de pesquisa, como seleção sequencial para trás (*backward selection*) e seleção seqüencial para frente (*forward selection*), algoritmo de busca *best-fit*, *genetic search algorithm* e pesquisa aleatória.

- **Filters**: processos que transformam instâncias e conjuntos de instâncias são chamados de "filters", e são classificados de acordo com o que faz mais sentido, apenas em um contexto de predição (chamado supervisionado) ou em qualquer contexto (chamado não-supervisionado). Além disso, são divididos em "*attribute filters*", que trabalham em um ou mais atributos de uma instância, e "*instance filters*", que trabalham em todas as instâncias.

A ferramenta Weka suporta a manipulação de diferentes formatos de arquivos como CSV, binário e C45, bem como, possibilita a busca direta de dados de servidores de banco de dados e também de servidores Web. O formato nativo da Weka é o ARFF (Attribute-Relation File Format), que é um arquivo ASCII usado para definir atributos e seus dados (WEKA, 2013). A apresentação dos resultados é disponibilizada em forma gráfica de histogramas, árvores de decisão, diagramas de dispersão, além de prover modelos gráficos para montagem de redes neurais (GOLDSCHIMDT; PASSOS, 2005). A figura 3.4 mostra a Interface *Explorer* do Software Weka.

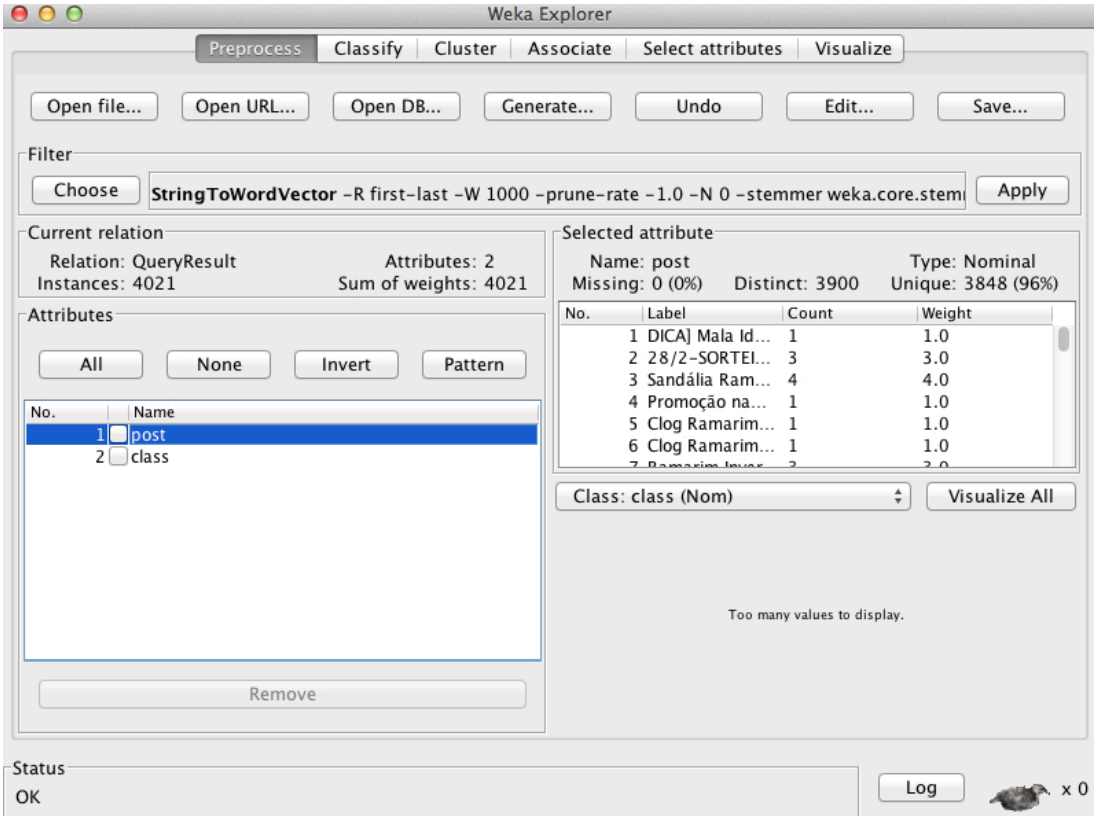

Figura 3.4 Interface *Explorer* do Software Weka

3.4 Classificadores

Nas seções seguintes serão apresentados os classificadores de AM que foram escolhidos para realização dos testes (todos classificadores usados nesse trabalho estão disponíveis no WEKA). Os três algoritmos aplicam o aprendizado supervisionado, ou seja, eles objetivam construir um modelo de distribuição de classes (categorias) em função das características dos dados em questão. Esse modelo é formado a partir de um conjunto de treinamento, onde as classes de suas instâncias são previamente conhecidas. A partir daí, o algoritmo é capaz de extrair características importantes para futuras classificações. São eles: *Naive Bayes*, o *Sequential Minimal Optimazation* (SMO) e o C4.5.

3.4.1 Naive Bayes

Os métodos de aprendizado Bayesiano utilizam um modelo probabilístico baseado no conhecimento prévio do problema, o qual é combinado com os exemplos de treinamento para determinar a probabilidade final (MITCHELL, 1997).

O *Naive Bayes* (JOHN; LANGLEY, 1995) é um dos algorimos mais utilizados em Aprendizado de Máquina apresentando excelentes resultados para a categorização de textos (JUNIOR; PASSOS, 2007). É baseado no Teorema de Bayes, formulado no século XVIII por Thomas Bayes (equação 1) e, como classificador, é considerado um dos mais eficientes em tempo de processamento e precisão quando da rotulação de novas amostras, características

justificadas pela abordagem simplória com que trata as características dos exemplos.

O Naive Bayes parte do principio que dado um conjunto contendo grupos divididos por valores e atributos é possível predizer em qual grupo uma nova instância pertence. A fórmula da equação 1 exemplifica esse conceito. Ela é definida como “a probabilidade de A dado B”, ou seja, dado um conjunto de evidências B, qual é a probabilidade da hipótese A estar em B (SEGARAM, 2007).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad (1)$$

O classificador é denominado ingênuo (*naive*) por assumir que os atributos são condicionalmente independentes, ou seja, a informação de um evento não é informativa sobre nenhum outro (o que não ocorre na maioria dos problemas práticos). Apesar desta premissa “ingênua” e simplista, o classificador reporta o melhor desempenho em várias tarefas de classificaçao (OGURI,2006). A Figura 3.5 ilustra seu modelo. Ela indica que cada atributo A_i influencia a classe C, mas nenhum exerce influência sobre o outro.

O aprendizado bayesiano é do tipo supervisionado, já que são fornecidas ao algoritmo de aprendizado de máquina as instâncias juntamente com seus rótulos (ou seja, as classes). Seguindo o paradigma estatístico, o algoritmo faz uso de fórmulas estatísticas e cálculo de probabilidades para realizar a classificação (MITCHELL, 1997). Considerado um algoritmo de aprendizado indutivo eficiente para a AM e mineração de dados. Em alguns domínios o desempenho tem mostrado compatível com aprendizado de redes neurais e árvore de decisão. (MITCHELL,1997).

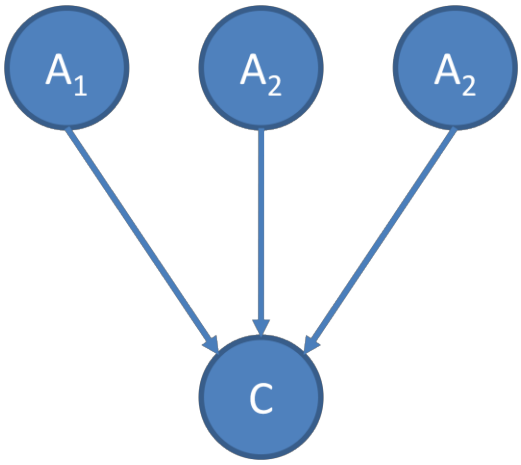

Figura 3.5 Modelo Naive Bayes

Outro fator que torna o Naive Bayes eficiente é que ele realiza a leitura dos dados do conjunto de treinamento apenas uma vez, para com isso estimar todas as probabilidades requeridas na classificação. O modelo pode ser usado de forma incremental, além do fato de que pode ser alterado facilmente com a inclusão de novos dados uma vez que probabilidades podem ser convenientemente revisadas.

Existem dois tipos de modelos estatísticos para os classificadores Naıve Bayes, o modelo binário e o modelo multinomial. O modelo binário representa um documento através de um vetor binaário, indicando a não-ocorrência de algum atributo com o valor 0 (zero), enquanto o valor 1 (um) representa no mínimo uma ocorrência. Já o modelo multinomial assume que o documento é representado por um vetor de valores inteiros, caracterizando o número de vezes que cada termo ocorre no documento (OGURI, 2007). Neste trabalho foi utilizado o modelo binário (disponível no WEKA).

3.4.2 Support Vector Machines (SVM)

Máquinas de vetor de suporte (Support Vector Machines) é uma técnica de classificação desenvolvida por Vapnik, baseada no princípio da minimização de risco estrutural (*Structural Risk Minimization*- SRM) (VAPNIK,1995). O SVM deriva da técnica de aprendizado por RNA, na sua forma computacional não algorítmica para obtenção do conhecimento. No entanto, a técnica SVM utiliza o espaço de hipóteses das funções lineares em um espaço de características de alta dimensão, treinada com um algoritmo de aprendizagem que deriva da teoria de aprendizado estatístico.

O SVM padrão toma como entrada um conjunto de dados e prediz, para cada entrada dada, qual de duas possíveis classes a entrada faz parte, o que faz do SVM um classificador linear binário não probabilístico. Dados um conjunto de exemplos de treinamento, cada um marcado como pertencente a uma de duas categorias, um algoritmo de treinamento do SVM constrói um modelo que atribui novos exemplos a uma categoria ou outra.

No método SVM, uma variável de predição é denominada atributo, e este atributo quando empregado na construção de hiperplanos é chamado de característica. O conjunto de características selecionadas para descrever um documento na classificação é chamado de vetor, e os vetores que se encontram próximos dos hiperplanos construídos para separar as categorias são chamados de vetores de suporte (SVM, 2013), como pode ser visto na Figura 3.6.

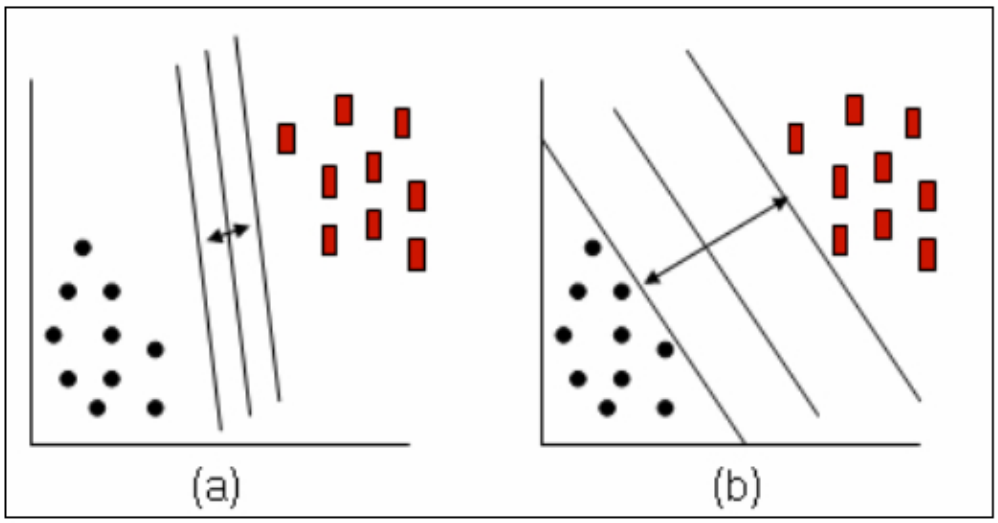

Figura 3.6 SVM - Hiperplano com margem de separação estreita (a) e margem de separação larga (b).

O método SVM é utilizado para reconhecimento de padrões sobre o espaço vetorial, a idéia deste método consiste em construir uma superfície de decisão (hiperplano) que melhor separe os padrões de treinamento das diferentes classes, ou seja, formar o plano que maximize a margem de separação das categorias para o processo de categorização de documentos. Na Figura 3.6 (a) pode ser visto um hiperplano com uma margem de separação estreita, e na Figura 3.6 (b) o hiperplano com uma margem de separação mais larga, que deverá ocasionar uma maior generalização para os padrões dados como entrada no processo de classificação.

O SVM conseguiu uma notável precisão para problemas importantes (CRISTIANINI; SHAWE-TAYLOR, 2003) (WEBB, 2002).

3.4.3 Sequential Minimal Optimization (SMO)

A técnica de reconhecimento SVM tem como objetivo separar os padrões em classes distintas através de uma superfície separadora, esta com a maior distância possível das margens, que definem uma segurança na classificação de novos dados. As margens são determinadas pelos vetores suportes, definidos pelo valor de seu respectivo Multiplicador de Lagrange, proveniente da modelagem dual. O produto interno que aparece no problema dual pode ser substituído por uma função kernel que possibilita encontrar uma superfície separadora não linear, abrangendo mais aplicações (ALES, 2008).

A utilização desta requer a escolha de uma função *kernel* adequada e a calibração dos parâmetros para que o processo apresente menos erros e generalize a situação. Para obtenção da solução do SVM utilizou-se o algoritmo SMO como kernel, desenvolvido por John Platt (1999), que escolhe dois pontos por iteração para fazer a otimização.

O algoritmo Sequential Minimal Optimization (SMO) (PLATT, 1999) é reconhecido como um dos mais rápidos e o mais fácil para implementar em software para o caso geral de SVM, o não linear e não separável. O algoritmo é iterativo e adota uma solução analítica para a otimização de um par de Multiplicadores de Lagrange em cada iteração, evitando o armazenamento de matrizes de grandes dimensões em memória (HERNANDEZ, 2009).

3.4.4. Árvores de Decisão

A Árvore de Decisão (AD) consiste de uma hierarquia de nós internos e externos que são conectados por ramos. Uma das principais características de uma Árvore de Decisão é o seu tipo de representação: uma estrutura hierárquica que traduz uma árvore invertida a qual se desenvolve da raiz para as folhas. A estrutura hierárquica traduz uma progressão da análise de dados no sentido de desempenhar uma tarefa de previsão/classificação. Em cada nível da árvore tomam-se decisões acerca da estrutura do nível seguinte até atingir os nós terminais (nós folha). (RODRIGUES, 2005)

A aprendizagem por árvore de decisão é um dos métodos mais usados e práticos para a inferência indutiva. A indução mediante árvores de decisão é uma das formas mais simples de algoritmos de aprendizagem e de maior sucesso. Recebe como entrada um objeto ou uma situação descrita por um conjunto de propriedades ou atributos, e retorna como saída uma decisão. Em termos de árvore de decisão, um exemplo é descrito pelos valores dos atributos e pelo predicado meta. O valor do predicado meta é chamado classificação do exemplo. Para cada

um dos possíveis valores de atributos, tem-se um ramo para outra árvore de decisão (sub-árvore). Cada sub-árvore contém a mesma estrutura de uma árvore.

Uma árvore de decisão é formada por um conjunto de regras de classificação. Cada caminho da raiz até uma folha representa uma destas regras. Cada percurso da árvore de decisão, desde um nó raiz até um nó folha, é convertido em uma regra, onde a classe do nó folha corresponde à classe prevista pelo conseqüente (parte "Então" da regra) e as condições ao longo do caminho correspondem às condições do antecedente (parte "Se" da regra). A figura 3.7 mostra um exemplo simplicado de uma AD para diagnóstico de pacientes (BARANAUSKAS; MONARD, 2000).

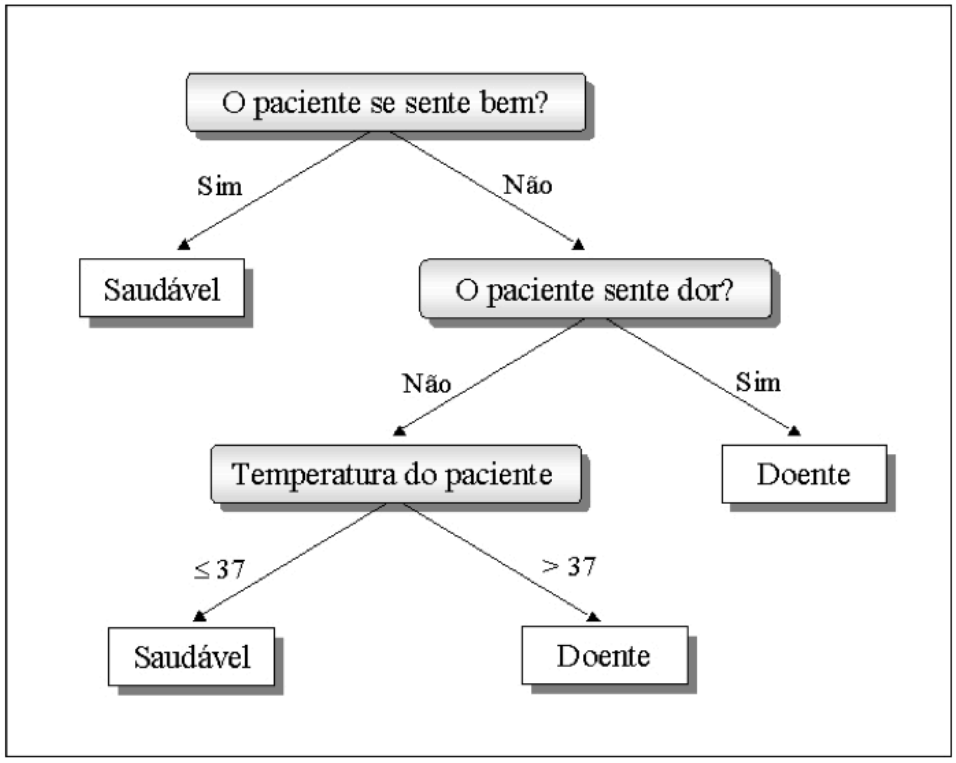

Figura 3.7 Exemplo de um AD simplificada (BARANAUSKAS; MONARD, 2000)

De acordo com. (FAYYAD, 1998) e (RODRIGUES, 2005), as regras de classificação que resultam da transformação de árvores de decisão podem ter as seguintes vantagens:

- São uma forma de representação do conhecimento amplamente utilizadas em sistemas especialistas;
- Em geral são de fácil interpretação pelo ser humano;
- Geralmente melhoram a precisão preditiva pela eliminação das ramificações que expressam peculiaridades do conjunto de treinamento que são pouco generalizáveis para os dados do teste.

Podem identificar-se algumas vantagens na utilização de Árvores de Decisão/Classificação,

das quais se destacam as seguintes:

- Ausência dos pressupostos típicos de modelos paramétricos de verificação difícil, especialmente se o número de variáveis explicativas é elevado;
- Possibilidade de utilização de variáveis explicativas em qualquer número e em várias escalas de medida e disponibilização de técnicas para lidar com valores omissos, permitindo evitar extensos e demorados tratamentos prévios aos dados;
- As variáveis podem ser utilizadas sem necessidade de transformação ou codificação como acontece com os atributos nominais em modelos de regressão ou discriminantes, onde a logaritmização por vezes é necessária para evitar problemas de heterocedasticidade;
- Possibilidade de integração de relações complexas entre as variáveis explicativas e a dependente e não apenas relações lineares, como acontece na maioria dos procedimentos estatísticos;
- Interpretabilidade dos resultados muito simples e clara, por simples observação da árvore.

As principais desvantagens centram-se essencialmente:

- Na instabilidade, pois pequenas perturbações do conjunto de treino podem provocar grandes alterações no modelo aprendido;
- Na fragmentação de conceito, ou seja, podem ocorrer replicações de sub-árvores.

3.4.5 C4.5

Este algoritmo segue um princípio orientador que é o princípio de *Occam* (*Occam´s Razor*), criado por William Occam, que dá preferência à escolha de hipóteses menos complexas, compatíveis com a realidade observada (QUINLAN, 1986).

O C4.5, assim como outros algoritmos de árvores de decisão, utiliza o método conhecido como “dividir para conquistar” que emprega uma pesquisa top-down para denotar os possíveis percursos na árvore em relação ao exemplo disponibilizado. A seleçao de quais atributos devem ser associados ao nó de decisão é feita através do uso do critério de ganho de informacão (*information gain*). O ganho mede o quanto um atributo´e capaz de separar um conjunto de exemplos em categorias. Aquele que possuir o maior ganho é selecionado para ser incluído na árvore.

Após a construçao da árvore de decisão, o C4.5 adota o estratégia pós-poda. Podar uma árvore neste contexto, significa reduzir algumas sub-árvores a folhas, ou de outra forma, um ramo de árvore, a partir de determinado nó é cortado (ou seja, transformado em folha). O corte de um ramo da árvore é guiado por um teste estatístico que leva em conta os erros em um nó e soma dos erros nos nós que descendem desse nó. Sendo assim, para cada nó da árvore, a poda só se concretiza se o desempenho da árvore não diminuir consideravelmente (QUINLAN, 1986).

Outra vantagem desse algoritmo é a capacidade que o mesmo possui de gerar regras de decisão a partir de árvores e de as compararem entre si independentemente das árvores construídas.

Um dos mecanismos de poda utilizados por este algoritmo é baseado na comparação das taxas de estimativas de erro de cada sub-árvore e do nó folha. São processados sucessivos testes a partir

do nó raiz da árvore, se a estimativa de erro indicar que a árvore será mais precisa se os nós descendentes (filhos) de um determinado nó forem eliminados, então estes nós descendentes serão eliminados e o nó *n* passará a ser o novo nó folha (CARVALHO, 1999).

Nesse trabalho foi utilizado o método J48 que é a implementação na linguagem de programação Java do algoritmo C4.5 (desenvolvida originalmete na linguagem C). Especificamente o método j48 procura gerar uma arvore de decisão a partir de uma abordagem recursiva de particionamento da base de dados (GOLDSCHIMDT; PASSOS, 2005). A árvore de decisão é um método simples de representar o conhecimento extraído a partir de uma base de dados. Sua função é sistematizar os dados facilitando a decisão a ser tomada (RUSSEL, 2002).

4 METODOLOGIA

O capitulo seguinte apresenta os passos realizados para o desenvolvimento do trabalho. Desde a obtenção do conteúdo das postagens, até a geração e validação do modelo preditivo.

4.1 Tecnologias Utilizadas

A fonte de dados utilizada neste trabalho está disponível em uma ambiente web, logo foi necessário primeiramente capturar os dados. Em seguida, armazenou-se esses dados em um base de dados para que fosse possível transformá-los com o objetivo de realizar uma classificação manual das postagens. Dessa forma utilizou-se um software de Mineração de dados para realizar o pré-processamento do conteúdo textual das postagens e posteriormente fazer uso de algoritmos de AM (descritos no capítulo 3).

As tecnologias que foram utilizadas no processo de desenvolvimento do trabalho foram:

- Facebook SDK for PHP;
- Linguagem de programação PHP (crawler);
- Linguagem de banco de dados SQL;
- MySQL Server;
- Biblioteca Apache;
- Weka;

4.2 Coleta e Armazenamento de Dados

Neste trabalho, o método de coleta foi feito através da utilização de um *crawler*, que pode ser definido como um robô que percorre a web de uma maneira específica, salvando as páginas visitadas de acordo com as necessidades do usuário. Os *crawlers* possuem vários nomes, sendo alguns destes *web spider* ou *web robot*.

Sua execução é bem simples e consiste no fornecimento de um site raiz (ou vários) para que ele inicie a navegação, sendo que este site recebe o nome de semente (ou *seed*) na terminologia de recuperação de informação. A partir desse site inicial dado, o crawler realiza uma "leitura" da página em busca dos links contidos na mesma, e à medida que vai encontrando essas novas urls, vai armazenando cada um em uma fila que é responsável por gerenciar os sites que o crawler deve visitar.

A coleta das postagens foi realizada a partir da implementação de um *crawler* que faz acesso a API do *Facebook* (nesse trabalho foi utilizado o *Facebook Software Development Kit*[7]). Através dela foi possível obter o conteúdo textual de cada postagem, assim como a informação do número de *likes* que cada postagem obteve.

[7] Dísponivel em: https://developers.facebook.com/docs/sdks/

No total foram capturadas 4871 postagens (escritas em português) contendo comentários sobre diversos produtos relacionados à moda como sapatos, bolsas, acessórios, sandálias, entre outros. As postagens capturadas datam desde o inicio 2010 (data de criação da página) até o início de 2013.

Cada postagem consistia originalmente de textos, na sua maioria curtos, acompanhados ou não de links, videos e/ou imagens de produtos.

Do número total de postagens capturadas, foram descartadas as postagens que apresentavam somente imagens (850 postagens), ou seja, foram armazenadas 4021 postagens que somaram um total de 1.257.164 *likes*. As postagens que apresentaram como conteúdo somente imagens foram excluídas pelo fato de que não seria possível aplicar nenhum tipo de pré-processamento nesses tipos de postagens.

4.4 Classificação das postagens

Após a coleta e armazenamento dos dados se realizou a classificação das postagens.

Para tal, levou-se em consideração o número de curtidas ou *likes* de cada postagem. Dessa forma, a popularidade de uma postagem foi definida conforme o número de vezes que os usuários do *Facebook* realizaram essa ação. A classificação foi feita considerando duas e três classes possíveis de popularidade.

Para realização dos testes considerando apenas duas classes possíveis (“boa” e “ruim”), as postagens foram ordenadas pelo seus respectivos número de *likes* e então dividas em dois grupos com aproximadamente 50% para cada classe, ou seja, cerca de 2011 postagens foram classificadas como “boas“ e 2010 mensagems foram consideradas “ruins”.

Para realização dos testes considerando três classes possíveis (“boa”, “média” e “ruim”), as postagens foram ordenadas pelo seus respectivos número de *likes* e então dividas em três grupos com aproximadamente 33% para cada classe, ou seja, cerca de 1341 postagens foram classificadas como “boas“, 1340 postagens foram classificadas como “médias“ e 1340 postagens foram classificadas como “ruins“,

4.5 Pré-processamento dos textos

Pré-processamento é a etapa realizada imediatamente após a Coleta, com o objetivo de se obter alguma estrutura para a massa textual. Pré-processar textos é, por muitas vezes, o processo mais oneroso da metodologia de mineração de textos, uma vez que não existe uma única técnica que possa ser aplicada para a obtenção de uma representação satisfatória em todos os domínios. Assim sendo, para se chegar à representação adequada, pode ser necessária a realização de muitos experimentos empíricos (JUNIOR; PASSOS, 2007).

O pré-processamento de textos consiste em um conjunto de transformações realizadas sobre alguma coleção de textos com o objetivo de fazer com que esses passem a ser estruturados em um representação atributo-valor. De modo geral, a etapa de pré-processamento tem por finalidade melhorar a qualidade dos dados já disponíveis e organizá-los. As ações realizadas na etapa de pré-processamento de dados visam prepará-los para serem submetidos a algum algoritmo de indexação ou mineração de dados (ARANHA, 2009).

Deve-se então realizar vários tipos de filtros para aumentar a qualidade inicial dos dados, aonde diversas técnicas podem ser aplicadas e até mesmo combinadas (JUNIOR; PASSOS, 2007). Essa fase é umas das principais etapas em todo o processo de Mineração de Textos, pois uma abstração ruim do documento pode impedir que este seja analisado corretamente.

Como resultado final do pré-processamento, em geral, tem-se uma representação de atributos e valores associados a eles, formando um par atributo/valor. Além disso, nessa etapa são decididas estratégias para manusear (tratar) campos que não são necessários no processo posterior de mineração de dados.

Antes que ocorra a classificação efetiva do documento ou texto é necessário convertê-lo em uma representação intermediária, um vetor de termos, a ser direcionada ao algoritmo de classificação. No Weka, a função *StringtoWordVector* foi utilizada para "quebrar" os textos em palavras antes de realizar as etapas de pré-processamento que são descritas a seguir:

4.5.1 Tokenização

O processo de tokenização possibilita a separação dos elementos constituintes do texto, identificando cada palavra que poderá compor a representação estatística do exemplo. O primeiro passo nesse processo de separação dos elementos do texto ocorre com a quebra do texto em palavras, mais precisamente *tokens*, através da identificação de *tokens* delimitadores como tabulação, espaço e nova linha. Para obter as melhores características presentes no texto, os processos de tokenização são sempre otimizados para avaliação dos textos utilizados, pois o processo de tokenização é dependente da linguagem.

Nessa etapa são descartados os caracteres especiais, que não contribuem para a extração do conhecimento. Por exemplo: `, ~, !, $, %, ^, &, *, (,), +, |, \, /, {, }, [,], :, ;, ?, ', =, ¨, -, §, °, ª, £, ¢, <, >, ¬, }, {, etc.

No caso do Facebook, os caracteres @ e # apresentam significados particulares. O @ (at) serve para referenciar usuários na rede social e o # (*hashtag*) serve para indexar conteúdo. Estes dois casos foram tratados em especial na etapa de pré-processamento.

4.5.2 N-grama

Em um processo de mineração de textos escolhe-se a forma de representação dos atributos, após selecionar a coleção de textos com a qual se vai trabalhar (MOURA et al., 2008b). Supondo que se vá trabalhar exclusivamente com a coleção de textos como uma "*bag of words*", desconsiderando o contexto semântico e/ou ordem de ocorrência, utilizam-se as palavras presentes nos textos ou suas combinações. Essas palavras ou suas combinações são chamadas de *n-gramas*.

A ocorrência de palavras em sequencia pode conter mais informação do que palavras isoladas. Desse modo, criando-se atributos pela união de duas ou mais palavras consecutivas, pode-se gerar atributos com um maior poder de predição. O n-grama é exatamente essa junção de palavras, onde N representa o número de palavras que foram unidas para a geração de um atributo. Nesse trabalho, foram realizados testes com unigramas (n=1), bigramas (n=2) e trigramas (n=3).

O pré-processamento unigrama considera o termo como sendo um único elemento de texto, ou seja, uma única palavra.

O pré-processamento de bigrama considera o termo como sendo dois elementos de texto, ou seja, duas palavras. Quando utiliza-se esse tipo de processamento o termo é gerado pela concatenação de uma palavra com a próxima palavra da mensagem separada por um espaço em branco,

O pré-processamento de trigrama considera o termo como sendo três elementos de texto, ou seja, três palavras. Quando utiliza-se esse tipo de processamento o termo é gerado pela concatenação de três palavras separadas por dois espaços em branco.

4.5.3 Remoção de Stopwords

Outra tarefa na preparação dos dados é a identificação das palavras que podem ser desconsideradas nos passos posteriores do processamento dos dados. Nesta fase tenta-se retirar tudo que não constitui conhecimento no texto. O resultado é uma lista com as palavras a serem descartadas ou uma *Stoplist*. Este conjunto de palavras é chamado de *Stopwords*.

Stopwords são palavras que não apresentam um conteúdo semântico de significância no contexto em que se encontram no texto, ou seja, são palavras consideradas irrelevantes na análise do texto. Geralmente trata-se de palavras auxiliares ou conectivas (por exemplo: *e, para, eles, elas*), que não fornecem nenhuma informação que venha a representar conteúdo dos textos. Na construção de uma lista de *stopwords* são acrescentadas palavras como preposições, pronomes, artigos e também podem ser adicionadas palavras que apresentam uma incidência muito alta em uma coleção de documentos e que, portanto não apresentam influência na categorização dos textos.

A análise dos dados textuais neste trabalho utiliza a ferramenta WEKA, que para executar a rotina *Stopwords*, responsável pelo processo de criação do dicionário de palavras de relevância nos documentos, utiliza uma lista *stoplist* passada como parâmetro no arquivo de propriedades da ferramenta, para assim desconsiderar as palavras que não apresentam significância no contexto dos documentos nas etapas posteriores.

4.6 Geração e Validação dos modelos preditivos

Após feitas as etapas de coleta, armazenamento, classificação e pré-processamento dos textos foram executados os algoritmos de classificação citados no capítulo 3 sobre o “bag of words“, e gerou-se modelos preditivos, os quais foram validados utilizando a técnica de estatística Validação Cruzada ou *Cross Validation*.

4.6.1 Cross Validation

O Cross Validation é uma técnica estatística de validação de algoritmos de aprendizagem utilizada para avaliar como os resultados de um algoritmo de aprendizagem se comporta ao receber conjuntos de dados independentes. A utilização mais comum dessa técnica se da quando se quer avaliar o quanto um modelo preditivo irá funcionar na prática (REFAEILZADEH et al., 2009).

Uma das maneiras de utilizar o Cross Validation é o *k-fold Cross Validation* onde k é o

número de subconjuntos de dados e número de iterações que será necessário para a execução da técnica. A figura 4.1 exemplifica um processo de validação cruzada (usando 3 subconjuntos de dados).

O processo consiste em dividir o conjunto de dados que se deseja utilizar de forma aleatória, para validação do algoritmo em N subconjuntos de dados igualmente distribuídos. Após a divisão iniciam-se os testes submetendo um dos subconjuntos para que o algoritmo classifique (conjunto de teste), e os demais como conjunto de treinamento. Repete-se o processo até que todos os subconjuntos tenham sido utilizados como conjunto de teste e como conjunto de treinamento. O resultado desse processo é uma matriz, onde é apontada a média de acertos e erros de cada classe (REFAEILZADEH et al., 2009). Nesse trabalho foi utilizado o 10-fold Cross Validation,

Para comparação de desempenho dos testes utilizou-se as medidas de acurácia, revocação, medida-f e precisão (todas descritas na seção 4.6.2) geradas a partir do resultado do *Cross Validation.*

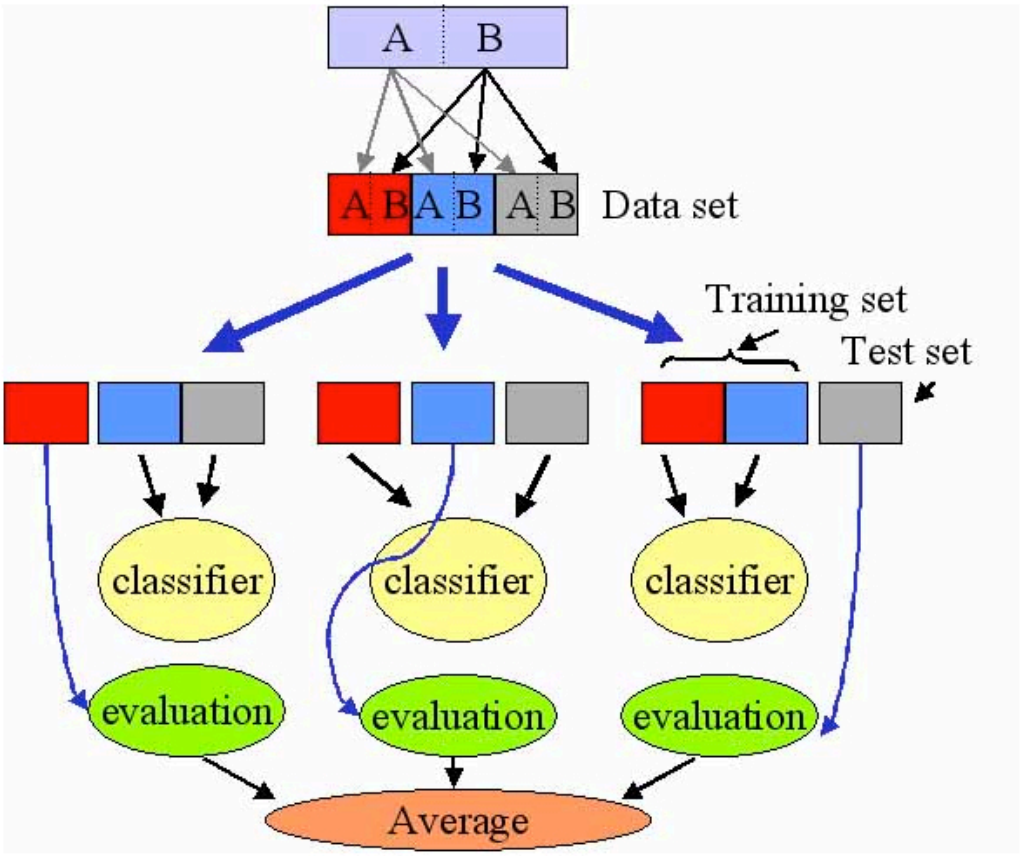

Figura 4.1: Exemplo do N-Fold Cross Validation (para N=3) (BABELOMICS, 2013).

4.6.2 Avaliação do conhecimento

Para tarefas de classificação que envolvem classes com valores discretos, algumas medidas de qualidade podem ser estimadas a partir dos seguintes resultados observados:

- **TP** (*true positive* ou positivos verdadeiros): número de exemplos positivos que foram corretamente classificados;
- **FP** (*false positive* ou falsos positivos): número de exemplos negativos classificados como positivos;
- **FN** (*false negative* ou falsos negativos): número de exemplos positivos classificados como negativos;
- **TN** (*true negative* ou negativos verdadeiros): número de exemplos negativos corretamente classificados;

A partir dessas medidas cria-se uma matriz de confusão, que é uma forma de visualização dos resultados utilizada para avaliar modelos de predição que utilizem algoritmos de classificação. Cada linha da matriz representa as instâncias preditas de uma classe, enquanto cada coluna da matriz representa as instâncias reais de uma classe, assim a diagonal principal representa os valores que foram preditos corretamente, e a diagonal secundária representa os valores que foram preditos incorretamente (FONTOURA, 2011).

A tabela 4.2 mostra a matriz de confusão para o caso de duas classes (P e N) onde constam os verdadeiros positivos e negativos (TP e TN), ou seja, a quantidade de mensagens que foram classificadas corretamente, e os falsos positivos e negativos (FP e FN), correspondentes a quantidade de mensagens classificadas de forma incorreta.

Classes	Classe verdadeira P	Classe verdadeira N
Classe prevista P	**TP**	**FP**
Classe prevista N	**FN**	**TN**

Tabela 4.2: Matriz de Confusão para duas classes

As medidas acima são utilizadas para estimar o desempenho das predições em geral, mas para a maioria das aplicações de texto, assim como classificação, é desejável uma análise mais detalhada dos erros (WEISS et al, 2005). As medidas de precisão (*Precision*), revocação (*Recall*), medida-F (*F-measure*) e acurácia (*Accuracy*) são interessantes medidas da qualidade de decisões binárias no problema de classificação de textos. Suas definições e fórmulas podem ser vistas nas equações abaixo utilizando as informações apresentadas na Tabela 5.4.

Precisão (*Precision***)**: proporção de exemplos positivos que foram classificados corretamente. Em problemas de classificação binária, a precisão pode ser obtida pela fórmula apresentada pela equação que segue:

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1)$$

Revocação (*Recall*): descreve a porção que foi classificada corretamente como exemplos positivos. Esta medida pode ser estimada pela fórmula da seguinte equação:

$$\text{Revocação} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

Medida-F *(F-Measure)*: As medidas precisão e revocação podem ser enganosas quando examinadas separadamente. Uma precisão elevada geralmente significa sacrificar a revocação e vice-versa. Quando as medidas precisão e revocação são sintonizadas para obter o mesmo valor, então este valor é chamado *break-even* (ponto de equilíbrio) do sistema e esse aspecto vem sendo muito utilizado em avaliações de sistemas de classificação de textos. (FRAGOS, et al., 2005)

A F-measure (ou medida F) é a média harmônica entre a precisão e a revocação e é definida pela equação:

$$\text{MedidaF} = \frac{2 \times (\text{Precisão} \times \text{Revocação})}{\text{Precisão} + \text{Revocação}} \quad (3)$$

Acurácia (*Accuracy)*: porcentagem de amostras positivas e negativas classificadas corretamente sobre a soma de amostras positivas e negativas, ou seja, trata-se da proporção de classificações corretas, e seu cálculo é descrito pela equação:

$$\text{Acurácia} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (4)$$

5 EXPERIMENTOS E RESULTADOS

Neste capítulo são apresentados os tipos de testes realizados para cada um dos três classificadores de AM escolhidos na realização desse estudo. Em seguida, são apresentados os resultados obtidos e são feitas comparações sobre o desempenho dos algoritmos para os diferentes tipos testes realizados.

5.1 Tipos de testes

Na realização dos testes, além de avaliar o comportamento dos algoritmos de classificacão, foi avaliado como diferentes tipos de pré-processamento influenciam no desempenho desses algoritmos na tarefa de gerar um modelo preditivo.

A tabela 5.1 apresenta os nove diferentes tipos de testes aplicados aos dois conjuntos propostos neste trabalho (com duas e três classes) para os algoritmos de classificação Naive Bayes, Sequential Minimal Optimization e C4.5. Dessa forma, são 18 testes para cada algoritmo, resultando num total de 54 testes.

Pré-Processamento	Teste ID
Unigrama	**1**
Unigrama + Remoção de Stopwords	**2**
Unigrama + Remoção de Stopwords + Remoção de Links, @, #	**3**
Bigrama	**4**
Bigrama + Remoção de Stopwords	**5**
Bigrama + Remoção de Stopwords + Remoção de Links, @, #	**6**
Trigrama	**7**
Trigrama + Remoção de Stopwords	**8**
Trigrama + Remoção de Stopwords + Remoção de Links, @, #	**9**

Tabela 5.1 Tipos de pré-processamento utilizados nos experimentos

Nas duas próximas seções, são apresentadas tabelas com as principais medidas para avaliação de desempenho citadas na seção 4.6.2. As informações de precisão, revocação e medida-f apresentadas nas seções 5.2 e 5.3 são representações da média dessas medidas entre as classes.

Também são apresentadas as matrizes de confusão para o melhor caso de cada conjunto de testes.

5.2 Testes com duas classes

Esta seção apresenta os resultados obtidos para os algoritmos Naive Bayes, SMO e C4.5 utilizando-se duas classes para classificação da popularidade das postagens (popularidades “boa” e “ruím”).

Resultados para Naive Bayes:

Teste ID	Precisão	Revocação	Medida-F	Acurácia
1	76,6%	75,9 %	75,7%	75.863%
2	**77,9%**	**76,8%**	**76,6%**	**76.829%**
3	70,5%	69,8%	69,6%	69.787%
4	75%	72%	71,2%	71.969%
5	72,7%	69%	67,8%	69.041%
6	71,1%	70,2%	69,9%	70.229%
7	68,2%	61,2%	57,3%	61.226%
8	64,1%	57,4%	52,0%	57.442%
9	70,2%	60,0%	54,5%	60.038%

Tabela 5.2 Resultados para duas classes usando Naive Bayes

O melhor resultado para duas classes usando Naive Bayes foi o com o pré-processamento número 2 (Unigrama com remoção de *stopwords*). A matriz de confusão gerada a partir desse teste foi a seguinte:

Classes	Classificada como “boa”	Classificada como “ruim”
“boa”	1724	660
“ruim”	272	1365

Tabela 5.3 Matriz de confusão: melhor resultado com Naive Bayes para duas classes

Resultados para o SMO:

Teste ID	Precisão	Revocação	Medida-F	Acurácia
1	80%	80%	80%	80.033%
2	**80,6%**	**80,6%**	**80,6%**	**80.557%**
3	75,9%	75,9%	75,9%	75.890%
4	79,4%	79,3%	79,3%	79.315%
5	77,8%	77,6%	77,6%	77.602%
6	74,4%	73,7%	73,5%	73.708%
7	75,8%	75,0%	74,7%	74.951%
8	74,5%	72,2%	71,5%	72.245%
9	74,5%	71,9%	71,1%	71.941%

Tabela 5.4 Resultados para duas classes usando SMO

O melhor resultado para duas classes usando SMO foi o com o pré-processamento número 2 (Unigrama com remoção de *stopwords*). A matriz de confusão gerada a partir desse teste foi a seguinte:

Classes	Classificada como "boa"	Classificada como "ruim"
"boa"	1592	378
"ruim"	404	1647

Tabela 5.5 Matriz de confusão: melhor resultado com SMO para duas classes

Resultados para o C4.5:

Teste ID	Precisão	Revocação	Medida-F	Acurácia
1	73,1%	73,1%	73,1%	73.073%
2	**74,4%**	**74,3%**	**74,3%**	**74.344%**
3	72,1%	72,0%	72,0%	72.024%
4	72,2%	72,2%	72,2%	72.217%
5	73,2%	70,8%	70,1%	70.809%
6	69,4%	64,0%	61,2%	64.015%
7	69,2%	59,8%	54,4%	59.790%
8	68,1%	57,9%	51,2%	57.884%
9	67,8%	56,5%	48,6%	56.503%

Tabela 5.6 Resultados para duas classes usando C4.5

O melhor resultado para duas classes usando C.45 foi o com o pré-processamento número 2 (Unigrama com remoção de *stopwords*). A matriz de confusão gerada a partir desse teste foi a seguinte:

Classes	Classificada como “boa”	Classificada como “ruim”
“boa”	1454	488
“ruim”	543	1536

Tabela 5.7 Matriz de confusão: melhor resultado com C4.5 para duas classes

5.3 Testes com três classes

A seguir são apresentados os resultados obtidos para os algoritmos de AM considerando-se três classes (“boa”, “média” e “ruim”).

Resultados para o Naive Bayes:

Teste ID	Precisão	Revocação	Medida-F	Acurácia
1	57,7%	56,3%	56,4%	56.255%
2	**58,6%**	**56,4%**	**56,6%**	**56.448%**
3	53,1%	51,6%	51,8%	51.588%
4	57,0 %	55,0%	54,6%	54.984%
5	56,2%	53,7%	53,8%	53.714%
6	56,5%	51,5%	51,9%	51.477%
7	55,3%	51,0%	48,9%	50.980%
8	52,8%	44,8%	40,1%	44.849%
9	56,9%	51,8%	49,7%	51.781 %

Tabela 5.8 Resultados para três classes usando Naive Bayes

O melhor resultado para três classes usando Naive Bayes foi o com o pré-processamento número 2 (Unigrama com remoção de *stopwords*). A matriz de confusão gerada a partir desse teste foi a seguinte:

Classes	Classificada como “boa”	Classificada como “média”	Classificada como “ruim”
“boa”	940	439	161
“média”	536	675	350
“ruim”	51	214	655

Tabela 5.9 Matriz de confusão: melhor resultado com Naive Bayes para três classes

Resultados para o SMO:

Teste ID	Precisão	Revocação	Medida-F	Acurácia
1	61,5%	61,9%	61,5%	61.861%
2	62,4%	63,0 %	62,5%	62.960%
3	60,9%	61,3%	60,8%	61.309%
4	60,6%	61,1%	60,7%	61.143%
5	60,0%	60,5%	59,8%	60.452%
6	58,8%	59,0%	58,3%	58.961%
7	**63,5%**	**63,0%**	**62,4%**	**62.966%**
8	62,9%	61,2%	60,4%	61.226%
9	63,1%	61,1%	60,1%	61.060%

Tabela 5.10 Resultados para três classes usando SMO

O melhor resultado para três classes usando SMO foi o com o pré-processamento número 7 (Trigrama). A matriz de confusão gerada a partir desse teste foi a seguinte:

Classes	Classificada como “boa”	Classificada como “média”	Classificada como “ruim”
“boa”	806	259	71
“média”	193	637	236
“ruim”	299	432	1088

Tabela 5.11 Matriz de confusão: melhor resultado com SMO para três classes

Resultados para o C4.5:

Teste ID	Precisão	Revocação	Medida-F	Acurácia
1	**62,1%**	**64,0%**	**62,8%**	**64.015%**
2	54,5%	55,5%	54,6%	55.481 %
3	52,9%	53,8%	53,0%	53.769 %
4	60,2%	63,9 %	60,6%	63.905 %
5	54,1%	52,4%	52,9%	52.416 %
6	49,7%	46,6%	46,5%	46.589 %
7	47,4%	41,8%	37,3%	41.811 %
8	47,6 %	40,9 %	35,8%	40.927 %
9	51,3 %	40,4%	36,5 %	40.430 %

Tabela 5.12 Resultados para duas classes usando C4.5

O melhor resultado para três classes usando C4.5 foi o com o pré-processamento número 1 (Unigrama). A matriz de confusão gerada a partir desse teste foi a seguinte:

Classes	Classificada como “boa”	Classificada como “média”	Classificada como “ruim”
“boa”	**465**	**242**	**199**
“média”	**201**	**227**	**185**
“ruim”	**301**	**319**	**1882**

Tabela 5.13 Matriz de confusão: melhor resultado com C4.5 para três classes

5.4 Comparação e Análise dos resultados

Pode-se perceber analisando as tabelas da seção 5.2 que se obtêm resultados melhores quando se considera apenas duas classes na análise do que quando se considera três classes. A maior diferença entre os testes equivalentes de duas e três classes chega a 40%.

É aceitável que os algoritmos lidem melhor com duas classes do que três, visto que se considerarmos um “chute” ao acaso em uma das três classes as chances de erro é de 66% enquanto que a chance de acerto é e apenas 33%. Ao fazer a mesma análise com duas classes percebe-se que as chances tanto de acerto quanto de erro são de 50%.

Outro fator que pode influenciar no desempenho é que as mensagens utilizadas nos teste com duas classes foram classificadas como “boas” e “ruins”. Os termos existentes nessas classes são muitas vezes opostos, ou seja, é mais improvável que um termo presente na classe “boa“ também esteja presente na classe “ruim“. Isso permite uma melhor separação dos dados e conseqüentemente melhor desempenho do modelo.

Diferentemente das características das classes “boa” e “ruim”, a classe “média” possui termos que podem se repetir com mais frequência nas demais classes. Uma hipótese mais provável da obtenção de piores resultados com três classes, é que classe “média” cause ruído no modelo, pois sua característica diminui a separação dos dados.

Em relação ao desempenho dos modelos gerados, considerando-se as suas respectivas acurácias, foi possível notar que:

O algoritmo SMO apresentou as melhores médias de resultados para duas e três classes (acurácias de 75,6% e 61,% respectivamente) assim como a melhor combinação (unigrama e remoção de stopwords) dentre todos os testes realizados, apresentando uma acurácia de 80.5%.

O algoritmo Naive Bayes teve uma boa média de resultados para duas classes (acurácia de 67,5%) com seu melhor desempenho sendo 76,8% (unigrama e remoção de stopwords) . Todavia, os resultados obtidos utilizando três classes mostraram uma desempenho bem abaixo dos resultados do SMO (52% e 61% respectivamente). Esse mesmo comportamento foi notado na utilização do C4.5. O algoritmo que utiliza arvores de decisão apresentou um rendimento razoável quando formam realizados os testes com duas classes (média de acurácia igual a 67,5%) Porém, os resultados com três classes obteve uma média de acurácia de apenas 51%. Apesar disso, utilizando a tokenização Unigrama o algoritmo conseguiu a melhor acurácia do conjunto de três classes (64%).

Em relação às outras medidas de desempenho foi possível perceber que:

- As medidas-f têm um rendimento muito parecido com os valores de acurácia e revocação quando se considera o classificador SMO, ou seja, ele é o algoritmo que, em média, mais consegue se aproximar de um ponto de equilibrio entre os três algoritmos analisados.
- Conforme se aumenta o valor N do N-grama, as medidas de precisão e revocação apresentam resultados mais dispersos, ou seja, menor é a medida-f.

Com base nos resultados apresentados, para os dois conjuntos de testes realizados, é possível notar algumas tendências em relação ao desempenho dos algoritmos de aprendizagem conforme modificam-se o tipo de pré-processamento dos textos:

- De maneira geral, o uso de bigramas e trigramas não melhorou o desempenho dos modelos quando comparados com unigrama. O fato das mensagems serem relativamente curtas nesse tipo de página pode ter contribuído para tal resultado, visto que os contextos são pequenos.
- O uso de stopwords é interessante somente quando se trabalha com unigramas, visto que no momento em que se analisa o contexto de uma forma mais ampla, a retirarada artigos, preposições e conjunções acabam afetando de forma acentuada a remoção dos atributos.
- O filtro de remoção de links e os símbolos reservados (@ e #) diminui a acurácia dos modelos. Isso pode ser explicado pelo fato de que os símbolos reservados são seguidos geralmente de nome de usuários no caso do @ e tópicos relacionados às postagens no caso do #. Logo, ambos podem ter influência no contexto das mensagems.

As figuras 5.1 e 5.2 apresentam os melhores resultados que cada algoritmo obteve nos conjuntos de duas e três classes respectivamente. As figuras 5.3 e 5.4 mostram as médias de acurácia dos três algoritmos utilizados nesse trabalho para ambos conjuntos de testes.

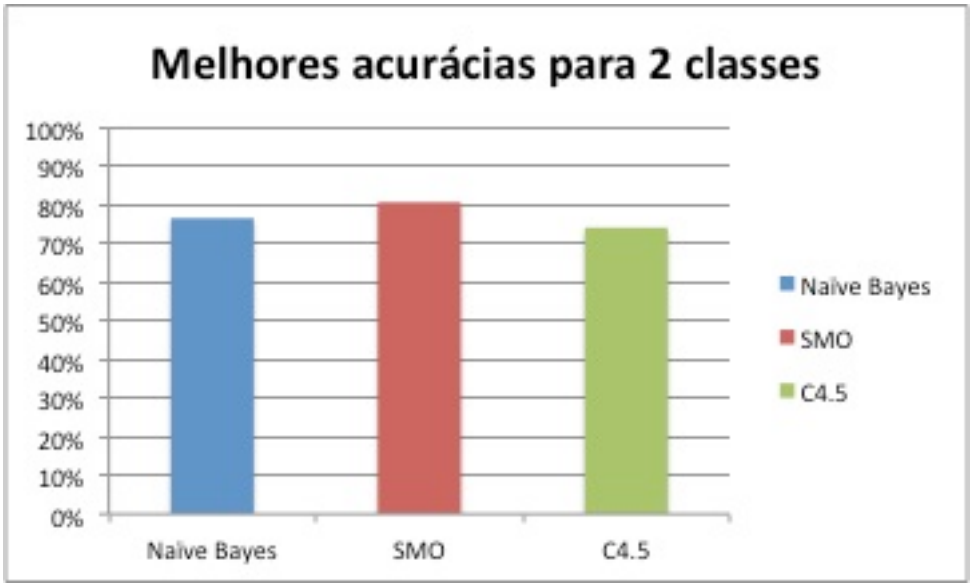

Figura 5.1 Gráfico dos melhores resultados para duas classes

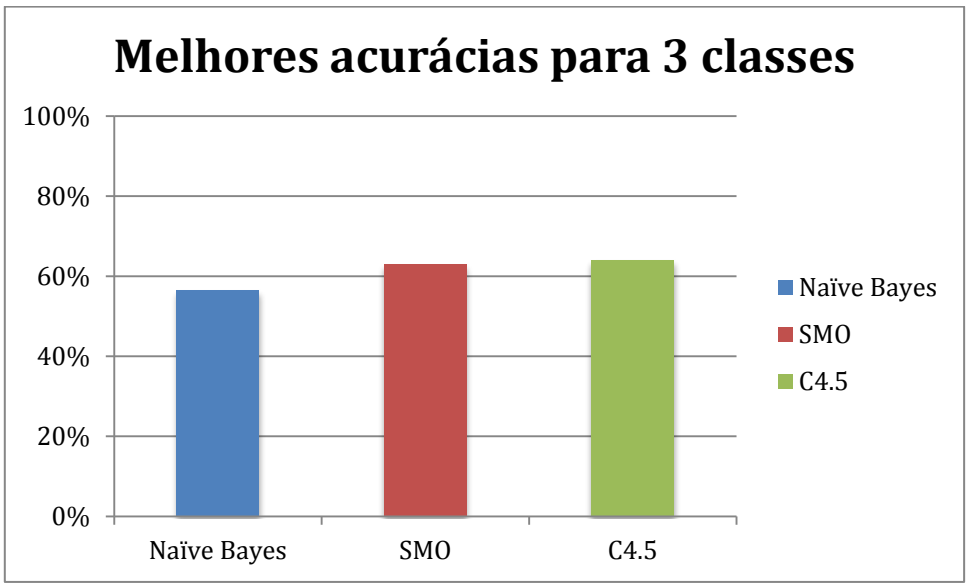

Figura 5.2 Gráfico dos melhores resultados para três classes

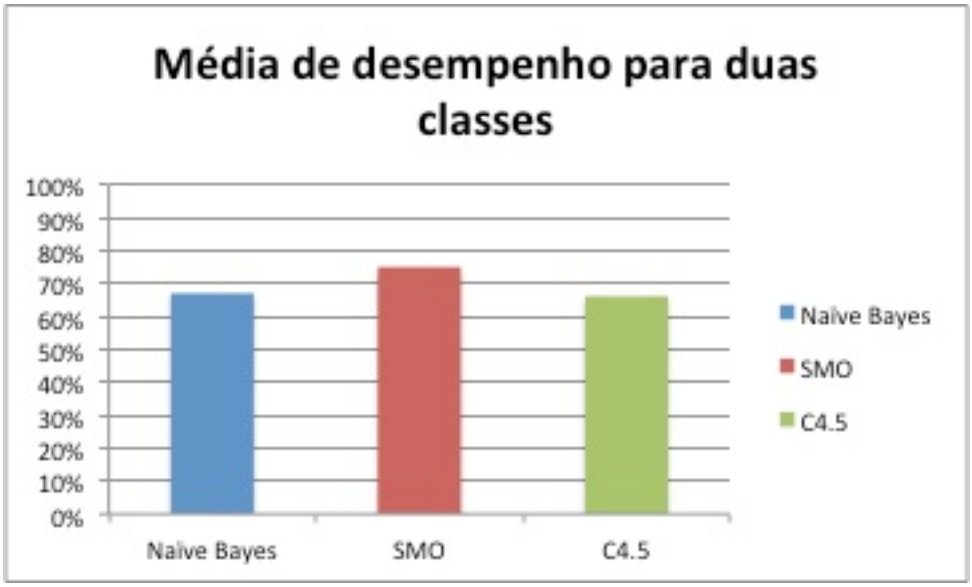

Figura 5.3 Gráfico da média de resultados para duas classes

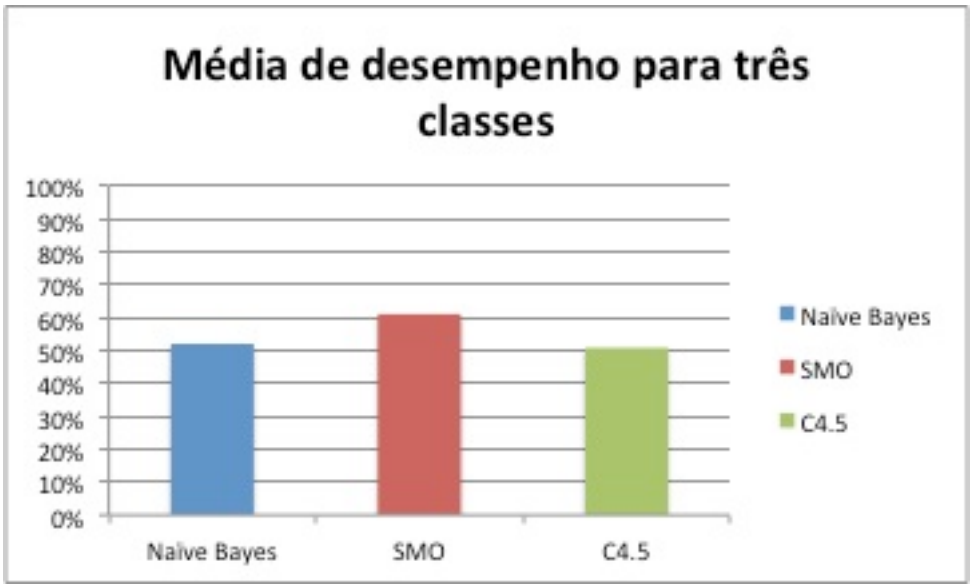

Figura 5.4 Gráfico da média de resultados para três classes

6 CONCLUSÕES

O trabalho teve como proposta avaliar a utilização de técnicas de pré-processamento de texto com três classificadores diferentes (Naive Bayes, SMO e C4.5) para predição da popularidade de postagens de uma fan page da rede social *Facebook*. Para isso, foram realizadas as seguintes etapas: coleta e armazenamento do conteúdo textual das postagens assim como a informação do número de *likes* de cada postagem; classificação das mensagems; pré-processamento de texto; geração e validação dos algoritmos de classificação.

Foram executados dois conjuntos de testes para cada classificador variando-se o número de classes. As postagens foram divididas em duas classes (“boa” e “ruim”) e três classes (“boa”, “médial” e “ruim”) conforme a popularidade das mesmas tendo em vista o número de *likes* que cada postagem obteve.

O algoritmo SMO apresentou os melhores resultados tanto na média dos resultados (a qual não variou da mesma forma como as médias dos algoritmos Naive Bayes e C4.5) quanto individualmente (acurácia de 80,5% para os testes com duas classes).

Com base nos testes realizados utilizando os classificadores já citados foi possível concluir, em relação ao pré-processamento realizados, que:

A remoção de stopwords é interessante apenas quando se trabalha com unigrama, visto que o contexto não é afetado quando se retiram palavras de parada (*stopwords*). Dessa forma, a remoção de stopwords com bigrama ou trigrama acabam prejudicando a acurácia dos modelos.

É possível que a perda de qualidade no desempenho dos classificadores quando aplicado o filtro de remoção de links e os símbolos reservados (@ e #) esteja relacionada com o número de características extraídas, o qual possa ter sido muito elevado e com isso fez com que a precisao do modelo diminuísse.

Outra conclusão em relação aos resultados apresentados é que o fator N-Grama piora os resultados do modelo conforme aumentasse o valor N, ou seja, quando contexto das postagens foi analisado de uma forma mais abrangente a acurácia do modelo caiu significativamente (de 75% para 61% no caso do Naive Bayes).

Isso pode ter acontecido devido ao fato de que as postagens presentes nesse tipo de página contêm textos não muito longos (muitas vezes acompanhados de vídeos, links ou imagens). Dessa forma, a análise dessas mensagems acaba não sendo interessante num processo de análise contextual.

6.1 Sugestões para Trabalhos Futuros

Esse trabalho mostrou os resultados alcançados na aplicação de três algoritmos de aprendizagem supervisionados para classificação de postagens encontradas no *Facebook.* Utilizou-se o aspecto de número de *likes* das postagens para determinar a popularidade das mesmas. Existem algumas alternativas que podem ser aplicadas em trabalhos futuros:

- Os conjuntos de treinamento podem ser classificados de outras formas. No caso mais específico do *Facebook,* poderiam ser analisados outros *feautures* como por exemplo o compartilhamento de postagens (botão *share*) ou o número de comentários das postagens;
- Utilização de diferentes *datasets* com um maior volume de dados para que a análise seja mais ampla;
- Tratamento de imagens para análise de sentimento, considerando que o conteúdo de postagens em rede sociais contêm um grande número de imagems associadas;
- Sabendo-se que uma quantidade considerável das mensagens postadas no *Facebook* possuem imagems e/ou links (para um vídeo ou texto de outro site), pode-se projetar *crawlers* para navegar até o site do link e obter os comentários registrados para o vídeo ou texto referido pelo link;
- Fazer uso de diferentes técnicas de pré-processamento de textos, como por exemplo o *stemming;*
- Por fim, poderia ser usado outros tipos de classificadores além dos que foram utilizados nesse trabalho, comparando-se com outros tipos de abordagens como por exemplo o aprendizado não-supervisionado.

REFERÊNCIAS

ALES Vanessa, T. **O algoritmo sequential minimal optimisation para resolução do problema de support vector machine: uma técnica para reconhecimento de padrões.** Tese de Mestrado em Matemática - Programas de Pós-graduação da CAPES. Curitiba, 2008. Dísponível em: < http://www.ppgmne.ufpr.br/arquivos/diss/200.pdf >. Acesso em: Novembro/2013

ARANHA C, Nunes **Uma Abordagem de Pré-Processamento Automático para Mineração de Textos em Português: Sob o Enfoque da Inteligência Computacional** Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 2007

BABELOMICS. **Predictors Tutorial** Disponível em: <http://docs.bioinfo.cipf.es/projects/1/wiki/Predictors_methods> Acesso em: Novembro/2013

BARANAUSKA, J. A., MONARD, M. C.: **Reviewing some Machine Learning Concepts and Methods**. 2000, Technical Report 102, Instituto de Ciencias Matematicas e de Computaçao, Universidade de São Paulo, São Carlos, Brazil,

CAMARGO Y. B. L. de, **Abordagem Linguística na Classificação Automática de Textos em Português Tese de Mestrado** Programa de Pós-graduação em Engenharia, UFRJ, Mestrado em Ciências em Engenharia Elétrica, Rio de Janeiro, 2007.

CARVALHO D. Ribeiro. **Árvore de Decisão/ Algoritmo Genético para tratar o problema de pequenos disjuntos em classificação de dados.** Programas de Pós-Gaduação em Computação de Alto Desempenho/Sistemas Computacionais. Universidade Federal do Rio de Janeiro, 2005.

CARVALHO Daniel; DIAS Maxwell, **Descoberta de conhecimento em ambientes virtuais de apredizagem: Um estudo de caso no Labsql** Trabalho de Graduação (Bacharelado em Ciência da Computação) – Faculdade de Computação, UFPR, Paraná, 2008.

CERVI, C. R. **Um Estudo sobre Mineração de Dados em Redes Sociais**, Programa de Pós-Graduação em Computação.Trabalho Individual II, UFRGS, Porto Alegre, 2008.

CRISTIANINI, Nello; SHAWE-TAYLOR, John. **An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods**. Cambridge University Press, 2003.

EIBE, Frank, HALL Mark, HOLMES Geoffrey, KIRKBY Richard, PFAHRINGER Bernhard, WITTEN Ian H. Witten Weka: A Machine Learning Workbench for Data Mining. In: T**he Data Mining and Knowledge Discovery Handbook**. New York: Springer, 2005. P. 1305-1314.

EMIRBAYER, M.; GOODWIN, J. **Networ analysis, culture and the problem of agency**. American Journal of Sociology, 1994.

ERICKSON, Thomas (2013): **Social Computing**. In: Soegaard, Mads and Dam, Rikke Friis (eds.). "**The Encyclopedia of Human-Computer Interaction, 2nd Ed.**". Aarhus, Denmark: The Interaction Design Foundation. Disponível em: <http://www.interaction-design.org/encyclopedia/social_computing.html>Acesso em: agosto de 2013.

FAYYAD, U. M, PIATETSKY-SHAPIRO, G., SMYTH, P., UTHRUSAMY, R. **Advances in knowledge Discovery & Data Mining**. AAAI/MIT, 1996.

FAYYAD, U. M, **Mining Databases: Towards Algorithms for Knowledge Discovery.** Data Engineering IEEE Computer Society. Washington. 1998.

FERAUCHE, T. **Aplicação de técnicas de mineração de textos para classificação de ementas da jurisprudência da justiça do trabalho de São Paulo** CEETEPS – Programa de pós-graduação;Mestrado em Tecnologia: Gestão, Desenvolvimento e Formação, 2011.

FONTOURA V. D. da. **Predição de falhas em projetos de software livre baseada em métricas de redes sociais** Trabalho de Conclusão de Curso - Tecnologia em Sistemas para Internet da Coordenação de Informática da Universidade Tecnológica Federal do Paraná – UTFPR, Campo Mourão, 2011.

FRAGOS, K., MAISTROS, Y., SKOURLAS, C.,"A weighted Maximum Entropy Language Model for Text Classification". In: **Proceedings of the 2nd International Workshop on Natural Language Understanding and Cognitive Science (NLUCS)**, pp. 55-67, Miami FL, 2005.

FREITAS, A.A. LAVINGTON, S.H. **Mining Very Large Databases with Parallel Processing,** MA: Kluwer Academic Publishers, 1998.

GIZ. **The Net Web.** Disponível em: <http://gizmodo.uol.com.br/facebook-3q2013/>. Acesso em Novembro de 2013.

GOLDSCHMIDT, R. R; PASSOS, E. P. L. **Data Mining: Um Guia Prático - Conceitos, Técnicas, Ferramentas, Orientações e Aplicações**. Rio de Janeiro: Editora Campus, 2005. v. 1. 250 p.

HAN, J.; KAMBER, M. **Data Mining - Concepts and Techniques**. 2a edição. Nova York: Morgan Kaufmann, 2006.

HAYKIN S. **Neural Networks - A Compreensive Foundation**. Prentice-Hall, Nova Jersey, segunda edição, 1999.

HERNANDEZ R, A. **MP-SMO: Um algoritmo para implementação VLSI do treinamento de máquinas de vetores de suporte,** Dissertação de Mestrado – Departamento de Engenharia, Universidade de São Paulo, 2009.

HUANG, Zhexue**.** Extensions to the k-means algorithm for clustering large data sets with categorical values. In**: Data Mining and Knowledge Discovery**, Kluwer Academic Publishers, Hingham, MA, USA, v. 2, p. 283-304. 1998.

JOHNN, G. H; LANGLEY, P. **Estimating continuous distributions in Bayesian classifiers.** Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (pp. 338-345). Montreal, Quebec: Morgan Kaufmann, 1995.

JUNIOR, J, R. Carrilho; PASSOS, E. P. Lopes (Orientador). **Desenvolvimento de uma Metodologia para Mineração de Textos.** Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 2007, 98p.

KAYA M; FİDAN G; TOROSLU I. H.; **Sentiment Analysis of Turkish Political News** partment of Computer Engineering Middle East Technical University, Ankara, Turkey, 2012.

LORENA A, C; CARVALHO A, C. P. L. F. de **Uma Introdução às *Support Vector Machines*** Departamento de Ciências de Computação, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2007.

MOURA, M. F.; MARCACINI, R. M.; NOGUEIRA, B. M.; CONRADO, M. S.; REZENDE, S. O. **A proposal for building domain topic taxonomies**. In: WORKSHOP ON WEB AND TEXT INTELLIGENCE, 1.; SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 19., 2008, Salvador. Proceedings... São Carlos, SP : USP, ICMC, 2008b. v. 1, p. 83-84.

MITCHELL T. M. **Machine Learning**. McGraw-Hill Education (ISE Editions), 1997, Pag. 5 – 9.

NEVES, R. D. C. D. D. **Pré-Processamento no Processo de Descoberta de Conhecimento em Banco de Dados** 2003, UFRGS Programa de Pós-Graduação em Computação.

OGURI, Pedro. **Aprendizado de Máquina para o problema de Sentiment Classification**. 2006. Tese. Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

PANG, Bo; LEE, Lilian; VAITHYANATHAN, Shivakumar. **Thumbs up? Sentiment Classification using Machine Learning Techniques**. Proceedings of EMNLP 2002, pp. 79–86.

PLATT, J. C. Fast training of support vector machines using sequential minimal optimization. In: SCHÖLKOPF, B.; BURGES, C. J. C.; SMOLA, A. J. **Advances in kernel methods: support vector learning**. 1st. ed. Cambridge, MA, USA: MIT Press, 1999. Chap. 2, pp. 185-208.

QUINLAN, J. R. **C4.5: programs for machine learning**. San Mateo, CA: Morgan Kaufmann, 1993.

QUINLAN, J. R. **Induction of decision trees**. Machine Learning. Pág. 81-106, 1986.

RBCCV. **A RBCCV nas redes sociais.** Disponível em: <www.scielo.br/pdf/rbccv/v27n3/v27n3a01.pdf>**.** Acesso em: Outubro de 2013.

REFAEILZADEH P; TANG L; LIU H. **"Cross Validation", in Encyclopedia of Database Systems (EDBS),** Editors: Ling Liu and M. Tamer Özsu. Springer, pp6. 2009.

REZENDE, S. O.; PUGLIESI, J. B.; MELANDA, E. A.; PAULA, M. F., **Mineração de Dados, in REZENDE, S. O. (Eds.), Sistemas Inteligentes**, Editora Manole Ltda., p.307-335. 2003.

ROCHA C. Lucena., **Análise de Fronteiras de Reservatório de Petróleo através de Geoquímica de superficie e Mineração de Dados Programas de pós-graduação de engenharia;** Rio de Janeiro, 2005.

RODRIGUES M. A. dos Santos. **Árvores de Classificação** Universidade dos Açores-Departamento de Matemática-Monografia, 2005.

STATISTA**. Leading social networks worldwide.** Disponível em: <http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of registered-users/>. Acesso em: Novembro de 2013.

SVM. **Support Vector Machines**. Disponível em: <http://www.dtreg.com/svm.htm>, Acesso em: Setembro de 2013.

TECH. **Números do Facebook.** Disponivel em: <http://canaltech.com.br/noticia/facebook/Facebook-divulga-numeros-impressionantes-relacionados-a-sua-plataforma/>. Acesso em: Outubro de 2013.

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

TONG S.: KOLLER D. ;:**Support Vector Machine Active Learning with Applications to Text Classification** Computer Science Department, Stanford University, Stanford CA, 2001.

VAPNIK, V. **The Nature of Statistical Learning Theory**. Springer, Nova York, 1995.

WEBB, A. **Statistical Pattern Recognition**. John Wiley & Sons, second edition, 2002.

WEISS, S.M.; ZHANG, T.; DAMERAU, F. **Text Mining: Predictive Methods for Analyzing Unstructured Information**. Springer Editora, Edição 1, 2005.

WEKA, **Weka 3 - Data Mining with Open Source Machine Learning Software in JAVA**, <http://www.cs.waikato.ac.nz/ml/weka/>, Acesso em: Novembro de 2013.

WITTEN, Ian H.; FRANK, Eibe; HALL, Mark A. **Data Mining - Practical Machine Learning Tools and Techniques.** 3ª ed Burlington**,** 2011.

WEBB, A. **Statistical Pattern Recognition**. John Wiley & Sons, Segunda Edição, 2002.

WU, Xindong; KUMARD Vipin; QUINLAN J. Ross; GHOSH Joydeep; YANG Qiang; MOTODA Hiroshi; MCLACHLAN G. J; NG Angus; LIU Bing; YU Philip S; ZHOU Zhi-Hua. STEINBACH Michae; HAND David J; STRINBERG Dan; **Top 10 algorithms in data mining.** Survey Paper, Department of Computer Science, University of Vermont, Burlington, VT, USA, 2007.

ZAGHLOUL W, LEE S, M; TRIMI S. **Text classification: neural networks vs support vector machines.** Industrial Management and Data Systems 109(5): 708-717 (2009).