Moderação de Sítios Utilizando Aprendizado Semissupervisionado Ativo

Felipe Vargas Rigo

SERVIÇO DE PÓS-GRADUAÇÃO DA UFMS

Data de Depósito:

Assinatura:_____

Felipe Vargas Rigo

Orientador: *Prof Dr Edson Takashi Matsubara*

Dissertação apresentada à Faculdade de Computação - FACOM-UFMS como parte dos requisitos necessários para obtenção do título de Mestre em Ciência da Computação, na Área de Inteligência Artificial.

UFMS - Campo Grande
Junho/2013

Dedicatória

Dedico este trabalho a minha esposa que sempre me apoiou e teve muita paciência comigo.

"Há uma força motriz mais poderosa que o vapor, a eletricidade
e a energia atômica: a vontade" - Albert Einstein

Agradecimentos

Gostaria de agradecer primeiramente a minha esposa, que me acompanha desde o começo desta batalha, sempre esteve ao meu lado e me ajudou a superar minhas dificuldades.

Agradeço também aos meus pais, que me deram toda a base para que eu pudesse chegar até aqui. Meu pai, Odilon, que é e sempre foi meu grande companheiro e contador de histórias. Minha mãe, Rosane, que sempre me deu muito amor. Meus irmãos, Alexandre e Guilherme, que são meus espelhos, meus ídolos, meus admiradores e minha admiração. A minha sobrinha querida e amada, Isadora.

Também quero expressar meus agradecimentos a todos amigos que de alguma forma me ajudaram a chegar até aqui, como Brivaldo Junior, Leonardo Torres, Cauan, aos meus colegas do Laboratório de Inteligência Artificial como Anderson, Hudson, Daiane e todos outros, mas especialmente ao Glauder Guimarães que sempre me ajudou e me aconselhou sempre que precisei, além de me ajudar fornecendo materiais e também ao Murillo Nicácio que me ajudou a desenvolver algumas ferramentas e com os testes.

Sou muito grato também à Universidade Federal de Mato Grosso do Sul, da qual aproveitei-me de sua infraestrutura e dos valiosos ensinamentos de seus distintos docentes, assim como a prazerosa convivência com meus colegas discentes. Também agradeço muito a meus colegas do NTI pelo suporte, apoio e compreensão, sem o qual não conseguiria completar esta jornada.

Por último, mas não menos importante, quero agradecer imensamente ao meu orientador, Professor Doutor Edson Takashi Matsubara, este mestre que logo será pai. Uma pessoa obstinada pelos seus estudos, pesquisas e extremamente dedicado ao seu trabalho. Um professor esforçado e que deixa muitos alunos surpresos ao compartilhar conosco o estudo, durante sábados, domingo e feriados, aprendendo e escrevendo durante o período em que muitos descansam e se divertem.

Sinceramente, meu muito obrigado a todos.

Resumo

A internet não é um lugar seguro para crianças. Sítios com conteúdo adulto e violento podem ser facilmente acessados por meio de qualquer computador, celular ou tablet. Uma possível solução é o uso de sistemas de controle de acesso a internet. Porém, os sistemas disponíveis atualmente necessitam de muita supervisão, nos quais deve-se bloquear sítio por sítio. Como a internet está em constante mudança, todos os dias sítios novos são criados, há muita dificuldade em fazer o controle do conteúdo que se deve bloquear. Com isso em vista, este trabalho propõe um novo *sistema moderador de conteúdo*. Esse sistema foi desenvolvido usando técnicas de Aprendizado de Máquina para facilitar a moderação de sítios da internet, tornando a moderação menos custosa e mais eficiente. Para atingir esse objetivo, faz-se uso do Aprendizado Semissupervisionado Ativo combinado com técnicas de seleção de atributos. Neste trabalho, averiguou-se o desempenho de algoritmos supervisionados para serem usados como algoritmos-base em uma técnica semelhante ao CoTesting, porém, ao invés de usar duas visões, usou-se dois algoritmos supervisionados juntamente com técnicas de seleção de atributos. Os resultados experimentais foram obtidos utilizando nove conjuntos de dados, todos relacionados a categorias relevantes para moderação de conteúdo de internet. Foram explorados diferentes técnicas de escolha de exemplos para o aprendizado ativo com melhorias significativas em duas estratégias. Além disso, verificou-se o uso de seleção de atributos para o problema e foi detectada diferença significativa nos resultados, indicando que o uso de seleção de atributos pode melhorar a qualidade da classificação. As contribuições deste trabalho não ficam apenas na resolução do problema de moderação de sítios de internet, mas também nas recomendações do uso de aprendizado semissupervisionado ativo e seleção de atributos para o problema.

Palavras-chave: aprendizado semissupervisionado, aprendizado ativo, moderação de conteúdo, seleção de atributos

Sumário

Lista de Figuras

Lista de Tabelas

Lista de abreviaturas

IA Inteligência Artificial

AM Aprendizado de Máquina

ASS Aprendizado Semissupervisionado

SVM *Suport Vector Machine* - Máquina de Vetores de Suporte

RI Recuperação de Informações

PLN Processamento de Linguagem Natural

MT Mineração de Textos

Capítulo 1

Introdução

A cada dia 500 mil novas pessoas entram pela primeira vez na internet. Em Novembro de 2012, o NIC.br (2012) registrou 44 milhões de usuários de internet domiciliares ativos somente no Brasil, com uma média de uso mensal superior a 33 horas. Stats (2013) indica que 88 milhões de brasileiros tiveram acesso à internet em 2012, com um aumento de mais de 17 vezes em relação ao ano 2000. De acordo com ITU (2013), mais de 2,7 bilhões de pessoas no mundo estavam usando a internet no primeiro trimestre de 2013, o que representa 39% da população mundial. E não somente o número de usuários tem crescido, mas também o conteúdo na internet. Em 1982 haviam 315 sítios na internet (Globo.com, 2010), atualmente são 670 milhões (Netcraft, 2013). Para se ter uma ideia desse crescimento, a cada minuto são disponibilizadas 20 horas de vídeo no YouTube e a cada segundo um novo blog é criado (NIC.br, 2012). Somente na leitura deste parágrafo, que deve demorar em torno de 30 segundos, foram criados aproximadamente 30 blogs e 10 horas de vídeo foram publicadas no Youtube.

Com o acesso facilitado, a internet permite que qualquer pessoa seja não somente consumidor de conteúdos web, mas também produtor. Nessa vantagem reside um perigo para crianças e adolescentes. Eles são expostos precocemente a informações indevidas para sua faixa etária. Algumas famílias simplesmente proíbem o acesso e em muitas outras o acesso é totalmente liberado. Neste trabalho acredita-se que exista um meio termo, diferente e individualizado para cada situação, e que pode ser alcançado com um *sistema moderador de conteúdo*.

Em um *sistema moderador de conteúdo*, o usuário tem acesso restrito à internet. Os sítios bloqueados compõem uma lista de sítios proibidos pelo sistema, como uma lista negra (*black list*) que é gerenciada periodicamente por um moderador. Abordagens assim são facilmente implementáveis em ambientes onde existe a figura de um administrador de rede, como em instituições, empresas e universidades.

Mesmo que a moderação de um sítio por um administrador de rede seja simples, sabe-se que a facilidade de criação de novas páginas de internet dificulta que seja feita a moderação de toda a internet, visto que praticamente todo usuário com acesso à internet pode criar novas páginas. A quantidade de sítios novos criados a cada dia torna quase

proibitivo mapear todos aqueles que se deseja bloquear na internet. Em ambientes domésticos a dificuldade é ainda maior, pois nem mesmo uma abordagem simples como a implementação de uma *blacklist* é praticável, devido a falta de conhecimento técnico para o bloqueio dos sítios.

Objetivo

Este trabalho visa o desenvolvimento de um sistema, usando técnicas de aprendizado de máquina, para facilitar a moderação de sítios da internet, tornando-a menos custosa e mais eficiente. Para atingir o objetivo deste trabalho, fez-se uso de Aprendizado Semissupervisionado (Chapelle et al., 2006) juntamente com *Rankings* (Burges et al., 2005) e Aprendizado Ativo (Settles, 2009). Com a união dessas técnicas é possível desenvolver um sistema customizável (*Rankings*) com pouca intervenção humana (Aprendizado Semissupervisionado Ativo).

Organização

Este trabalho está dividido da seguinte maneira: no Capítulo 2 é apresentada uma breve descrição sobre Aprendizado de Máquina e uma breve introdução dos conceitos. No Capítulo 3 é realizada uma revisão bibliográfica sobre os conceitos chaves utilizados no desenvolvimento deste trabalho e suas técnicas. No Capítulo 4 é descrito o projeto do sistema moderador, abrangendo seu funcionamento, implementação algorítmica, documentação e protótipo das interfaces e, posteriormente, no Capítulo 5, são apresentados os experimentos realizados com o sistema e seus resultados. Por último, o Capítulo 6 apresenta as considerações finais sobre o projeto desenvolvido neste trabalho.

Capítulo 2

Conceitos Introdutórios

O desenvolvimento de um *sistema moderador de conteúdo* se torna possível com os recentes avanços em Aprendizado de Máquina (AM), uma área de pesquisa associada à Inteligência Artificial (IA). O Aprendizado de Máquina pode ser definido como uma área que pesquisa métodos computacionais relacionados à aquisição automática de novos conhecimentos, novas habilidades e novas formas de organizar o conhecimento já existente (Mitchell, 1997). Este Capítulo visa introduzir o leitor nessa área de pesquisa, para se ter uma ideia geral de seu funcionamento e suas ambições.

Organização

Este capítulo está organizado da seguinte forma: na Seção 2.1 são introduzidos os conceitos gerais sobre Aprendizado de Máquina e, logo em seguida, na Seção 2.2 é apresentada a notação que será usada nas seções seguintes. Nas Seções 2.3 e 2.4 são apresentados os conceitos de Aprendizado Supervisionado e Não Supervisionado, respectivamente. Logo em seguida, na Seção 2.5 é apresentado brevemente o Aprendizado Semissupervisionado, que é o foco deste trabalho, o qual é tratado em mais detalhes no Capítulo 3.

2.1 Aprendizado de Máquina

Antes de definir o que é Aprendizado de Máquina (AM), é necessário primeiro definir o que é aprendizado. Ao buscar a definição de "aprender" em alguns dicionários (Michaelis, 2012; Oxford, 2008; Tolman e Honzik, 1930), são encontradas definições como:

- obter conhecimento sobre algo através do estudo, experiência ou sendo ensinado;
- se tornar ciente através de informações ou observações;
- guardar na memória;
- receber instruções;

- o modo como os seres adquirem novos conhecimentos, desenvolvem competências e mudam o comportamento.

Esse conceito mostra-se deficiente quando visto sob a perspectiva da computação. Para as duas primeiras definições é difícil verificar quando o aprendizado foi atingido ou não. Então, como saber quando uma máquina obtém conhecimento sobre algo? Mesmo que fossem feitas perguntas à máquina, não se estaria testando se ela realmente aprendeu, mas se ela está apta a responder perguntas.

Entretanto, para os outros significados, apesar de serem denotados em termos humanos, são próximos do desejado para o Aprendizado de Máquina. Para um computador é trivial armazenar informações na memória e receber instruções para uma tarefa. Porém, o objetivo está em melhorar o desempenho (através do aperfeiçoamento). Pode-se dizer que as máquinas aprendem quando elas alteram seu comportamento de modo a ter uma melhor performance no futuro. Apresentado dessa maneira o aprendizado está mais relacionado a *performance* do que ao *conhecimento* (Witten e Frank, 2011).

Pode-se dizer que o Aprendizado de Máquina, uma área da Inteligência Artificial, trata de questões relacionadas a como programas de computador podem se aperfeiçoar com a experiência. Os fundamentos do AM utilizam conceitos de outras áreas incluindo estatística, filosofia, matemática, teoria da informação, biologia, ciência cognitiva, complexidade computacional, teoria de controle, entre outros (Mitchell, 1997).

O Aprendizado de Máquina tem sido muito usado na mineração de dados (*data mining*) para descoberta de conhecimento em grandes volumes de dados, como encontrado em sistemas comerciais contendo registros de manutenção de equipamentos, pedidos de empréstimos, transações financeiras, registros médicos e outros do gênero. Conforme a tecnologia dos computadores amadurece, parece inevitável que o Aprendizado de Máquina ocupe um papel central cada vez maior na ciência da computação e na tecnologia de computadores (Mitchell, 1997).

Considere projetar um jogo de damas que utiliza um sistema de aprendizado. Nesse sistema de aprendizado deve-se escolher como será adquirida a experiência durante o treinamento. Existem dois meios de adquirir esta experiência: através do aprendizado direto ou do aprendizado indireto. No aprendizado direto o sistema executa a mesma tarefa várias vezes, como jogar damas contra si ou contra outros jogadores. Enquanto no aprendizado indireto o sistema já tem os exemplos prontos da tarefa executada. O aprendizado indireto é mais difícil pois no jogo de damas um bom começo, por exemplo, não garantirá a vitória e poderá resultar em uma derrota, o que não desmerece a jogada inicial. Então, é preciso determinar com que grau cada jogada contribui para o resultado final do jogo. Essa determinação é conhecida como **atribuição de crédito** (Mitchell, 1997).

Um segundo ponto importante na experiência com o treinamento é o grau com que o aprendiz controla a sequência dos exemplos de treinamento. Alguns exemplos de controle seriam:

- se o aprendiz apenas observa o jogo, ou seja, apenas recebe a informação;
- se ele joga do início ao fim, ou seja, se ele controla todas as decisões do jogo;
- se ele questiona o treinador nas jogadas que tiver dúvida, ou seja, há alguém supervisionando o jogo.

Outro ponto importante na experiência com o treinamento é o quão bem distribuído é o conjunto de exemplos e como é mensurado a performance final do sistema. Por exemplo, se o sistema treinar apenas contra si, ao encarar alguém com outra técnica, ele pode perder o jogo.

Há 3 elementos básicos necessários ao Aprendizado de Máquina:

- Uma tarefa T;
- Uma medida de performance P;
- Uma medida de experiência E.

Diz-se que um programa de computador **aprende** melhorando a experiência E em relação a uma classe de tarefas T mensurado pela performance P. Em outras palavras o programa aperfeiçoou a execução de sua tarefa.

No exemplo do jogo de damas tem-se esses dados como:

- Tarefa T: jogar damas;
- Performance P: porcentagem de jogos ganhos;
- Experiência E: a prática dos jogos de dama contra si ou contra outros programas ou pessoas, ou ainda, a gravação de jogos anteriores de outrem.

Para determinar que tipo de conhecimento adquirir e como ele deve ser usado, deve-se escolher uma **função alvo** f. No jogo de damas, essa função f deve devolver a melhor jogada em cada momento ou deve indicar pesos aos estados do tabuleiro. Caso não se consiga determinar uma função alvo diretamente, deve-se definir uma descrição operacional da função alvo, ou seja, aquela que guiará o movimento, como a que escolhe a melhor jogada a seguir. Caso essa função retorne apenas uma aproximação da melhor jogada, então diz-se que essa é uma **função de aproximação** (h).

Com uma função de aproximação, o problema de incrementar o desempenho P na tarefa T é reduzido ao problema de aprender alguma função-alvo em particular. Tendo uma função alvo ideal f, é preciso escolher uma representação que o programa irá usar para descrever a função h. Essa representação pode ser um vetor com o estado atual do tabuleiro, um grafo, um conjunto de preposições ou até uma rede neural.

Existem algoritmos de aprendizado que se adequam melhor a cada modelo, entre esses algoritmos há uma característica marcante que os difere pelo modo como os parâmetros são modificados e pela maneira com a qual se relacionam com o ambiente. Nesse contexto

pode-se dividir os algoritmos de Aprendizado de Máquina em aprendizado Supervisionado, Não Supervisionado e Semissupervisionado. Na próxima seção serão definidas algumas notações usadas pelos algoritmos de Aprendizado de Máquina.

2.2 Notação

Para manter uma maior clareza nos tipos de Aprendizado de Máquina deve-se definir alguns conceitos.

Zhu e Goldberg (2009) definem que no Aprendizado de Máquina tem-se o domínio dos exemplos (ou atributos) $X = (x_1, x_2, \dots, x_{n_{\text{ex}}})$ e, no aprendizado supervisionado, o domínio dos rótulos (ou classes) $Y = (y_1, y_2, \dots, y_{n_{\text{rot}}})$, onde n_{ex} é o número de exemplos e n_{rot} é o número de rótulos. Tendo $P(x, y)$ como uma junção de distribuições desconhecidas dos exemplos e rótulos $X \times Y$. Dada uma amostra de treinamento, o aprendizado supervisionado induz uma função $h : X \longrightarrow Y$ que se aproxima da verdadeira função desconhecida $f : X \longrightarrow Y$, com o objetivo de que $h(x)$ faça a predição do verdadeiro rótulo y_i de um novo exemplo x_i. Ou seja, o algoritmo de aprendizado constrói um modelo que mapeia o exemplo x_i a um rótulo de classe y_i.

Um exemplo é frequentemente representado por um vetor de atributos de n_{atr} dimensões $x_i = (x_{i1}, ..., x_{in_{\text{atr}}})$ com $i = 1, ..., n_{\text{ex}}$. Um exemplo x_i também pode ser visto como um ponto num espaço n_{atr} dimensional. Cada atributo do exemplo representa uma dimensão e o comprimento n_{atr} do vetor de atributos é denominado dimensionalidade do vetor de atributos (Zhu e Goldberg, 2009).

Na Tabela 2.1 é apresentado o formato geral de uma tabela no formato atributo-valor com n_{ex} exemplos x_i. Assim, x_{ij} refere-se ao valor do atributo j do exemplo x_i, podendo ser representado por valores contínuos, discretos ou booleanos. Os valores y_i, os quais podem ou não existir (se o aprendizado for supervisionado ou não), referem-se ao valor do atributo *classe* pertencente ao conjunto Y.

Tabela 2.1: Exemplos no formato atributo valor

	x_{i1}	x_{i2}	$\dots$	$x_{in_{\text{atr}}}$	Y
x_1	x_{11}	x_{12}	$\vdots$	$x_{1n_{\text{atr}}}$	y_1
x_2	x_{21}	x_{22}	$\vdots$	$x_{2n_{\text{atr}}}$	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$x_{n_{\text{ex}}}$	$x_{n_{\text{ex}}1}$	$x_{n_{\text{ex}}2}$	$\vdots$	$x_{n_{\text{ex}}n_{\text{atr}}}$	$y_{n_{\text{ex}}}$

2.3 Aprendizado Supervisionado

O Aprendizado é dito supervisionado quando um agente externo indica ao algoritmo a resposta desejada para o padrão de entrada. No aprendizado supervisionado o rótulo

$y_i \in Y$ é a predição desejada para um exemplo $x_i \in X$. Os rótulos distintos entre si foram o conjunto de valores do atributo classe.

Em um jogo de damas, por exemplo, o valor do atributo classe poderia ser a vitória ou a derrota, ou seja, $Y = \{Vitoria, Derrota\}$. Uma outra maneira de representar a classe é defini-la como valores inteiros, sendo a vitória = 1 e a derrota = -1, ou seja, $Y = \{1, -1\}$. Essa codificação em particular é usada para rótulos binários e os valores da classe são denominados genericamente de positivo e negativo, respectivamente. Para codificações multiclasse, comumente se utiliza a codificação $y_i = 1, 2, ..., n_{\text{rot}} \in \mathbb{N}$, onde n_{rot} é o número de valores (rótulos) que a classe pode assumir. Por exemplo, pode-se ter os seguintes rótulos que representam o resultado do jogo: vitória = 1, empate = 2 e derrota = 3. A classe é um vetor nominal e portanto não possui informações de ordem e escala, ou seja, o rótulo 1 não está necessariamente mais próximo do valor de classe com rótulo 2 ou mais distante do valor de classe 3, os rótulos são atribuídos normalmente de maneira arbitrária.

Dependendo do domínio do rótulo y_i, os problemas de aprendizado supervisionado podem ser divididos em problemas de classificação e problemas de regressão, descritos a seguir.

Classificação: Constituem os problemas de aprendizado supervisionado onde os valores do atributo classe Y são discretos. Exemplo $Y = \{-1, 1\}$. A Tabela 2.2 exemplifica o conjunto X dos dados sobre as condições do tempo atuais e o conjunto Y com o resultado se deve ou não praticar esporte. Cada um desses resultados geralmente é codificado como um número inteiro arbitrário, pois muitos algoritmos de aprendizado utilizam este formato: Sim (positivo) = 1, Não (negativo) = -1. Após codificar os rótulos como inteiros, o algoritmo irá induzir uma função $h : X \longrightarrow Y$ com o objetivo de predizer o rótulo dos novos exemplos.

Regressão: Constituem os problemas de aprendizado supervisionado nos quais os valores do atributo classe Y são contínuos ($Y \subseteq \mathbb{R}$). Por exemplo, a Tabela 2.3 tem os dados coletados sobre os incêndios nas florestas de uma região de Portugal, na qual o objetivo é prever a área queimada dos incêndios florestais com o uso de dados meteorológicos. O algoritmo de aprendizado induz um modelo $h(x)$ que é uma aproximação da função desconhecida $f(x)$. Na Tabela 2.3, vários índices (atributos que formam o conjunto X) ajudam a determinar a área queimada (Y) em km^2, tais como o FFMC (*Fine Fuel Moisture Code*), DMC (*Duff Moisture Code*), DC (*Drought Code*), ISI (*Initial Spread Index*) do sistema canadense para determinação do perigo de incêndios, além da temperatura, umidade relativa (UR), vento, chuva, latitude e longitude.

As tarefas de regressão e classificação predizem o valor da classe correspondente ao exemplo, para exemplos não vistos previamente. Assim, as tarefas de classificação e regressão consistem basicamente na generalização dos exemplos que tenham uma classe

Tabela 2.2: Exemplos de treinamento positivos e negativos para o conceito alvo de Praticar Esporte (Mitchell, 1997)

Ex.	Céu	Temp. Ar	Umidade	Vento	Prev. Tempo	Praticar (Y)
1	Ensolarado	Quente	Normal	Forte	Mesma	Sim
2	Ensolarado	Quente	Alta	Forte	Mesma	Sim
3	Chuvoso	Frio	Alta	Forte	Mudança	Não
4	Ensolarado	Quente	Alta	Forte	Mudança	Sim

Tabela 2.3: Dados sobre Incêndios Florestais em Portugal (Adaptado de Cortez e Morais (2007))

#	long.	lat.	FFMC	DMC	ISI	temp	UR	vento	chuva	area (Y)
1	8	6	90,1	108,0	12,5	21,2	51	8,9	0	0,61
2	3	5	88,1	25,7	3,8	14,9	38	2,7	0	0,0
3	3	5	93,5	139,4	20,3	17,6	52	5,8	0	0,0
4	3	6	92,4	124,1	8,5	17,2	58	1,3	0	0,0
5	3	6	90,9	126,5	7,0	15,6	66	3,1	0	0,0
6	9	9	85,8	48,3	3,9	18,0	42	2,7	0	0,36
7	1	4	91,0	129,5	7,0	21,7	38	2,2	0	0,43
8	2	5	90,9	126,5	7,0	21,9	39	1,8	0	0,47
9	1	2	95,5	99,9	13,2	23,3	31	4,5	0	0,55
10	7	5	96,1	181,1	14,3	27,3	63	4,9	6,4	10,82

conhecida em uma linguagem capaz de reconhecer a classe de um exemplo com rótulo desconhecido (Zhu e Goldberg, 2009). Comumente, o conjunto de dados ($X \times Y$) é dividido em dois conjuntos disjuntos: o conjunto de treino (usado para induzir a hipótese) e um conjunto de teste (usado para validar a hipótese, verificando se ela é suficientemente genérica, mostrando que o algoritmo *aprendeu* com os exemplos). Entretanto deve-se tomar o cuidado ao se induzir uma hipótese, sob o risco cair em um dos seguintes problemas:

Sobreajuste (*overfit*): É o caso no qual a hipótese induzida é altamente especializada no conjunto de treino, atingindo uma alta taxa de acertos no conjunto de treino, mas uma baixa taxa de acertos no conjunto de teste, demonstrando que na verdade o algoritmo *decorou* os exemplos e não conseguiu generalizar o conceito contido nos exemplos. Em uma rede neural, por exemplo, se o número de neurônios for muito grande, a função induzida pode incorporar o ruído dos dados, perdendo sua capacidade de generalização.

Subajuste (*underfit*): É a situação oposta do sobreajuste. Ocorre quando o algoritmo de aprendizado não consegue abstrair o conhecimento do conjunto de treino. Isto pode ocorrer devido a um subdimensionamento de parâmetros do algoritmo ou o conjunto de exemplos não é representativo. Em uma rede neural, por exemplo, se o número de neurônios não for suficientemente grande, a função induzida pode não ser genérica o suficiente para contemplar todos os exemplos de treino, gerando uma baixa taxa de acertos tanto no conjunto de teste quanto no conjunto de treino.

Tanto os problemas de classificação quanto os problemas de regressão assumem que há dados rotulados suficientes e que há informação de todos os rótulos. Para casos nos

quais não há informação, *a priori*, sobre os rótulos dos exemplos é usado o aprendizado não supervisionado, conforme visto a seguir.

2.4 Aprendizado Não Supervisionado

Em Aprendizado de Máquina, o aprendizado não supervisionado é uma classe de problemas na qual procura-se determinar como os dados estão organizados. A principal característica do aprendizado não supervisionado é que ele recebe apenas exemplos não rotulados. Esse é um problema muito comum na mineração de dados, onde são usados muitos métodos de aprendizado não supervisionado para processá-los (Zhu e Goldberg, 2009).

No aprendizado não supervisionado não existe a supervisão de um moderador (oráculo) na manipulação dos exemplos. A entrada do algoritmo de aprendizado não supervisionado é um conjunto de dados X no qual cada exemplo consiste somente do vetor x, sem o atributo classe Y. O aprendizado não supervisionado visa encontrar alguma regularidade no conjunto de dados, formando agrupamentos de exemplos que tem alguma semelhança, ou seja, os exemplos são descritos em agrupamentos (*clusters*) de acordo com alguma medida de similaridade. Diferentemente de tarefas preditivas como a classificação, as tarefas descritivas consistem na identificação de propriedades intrínsecas dos conjuntos de dados que não possuem uma classe especificada. As tarefas mais comuns do aprendizado não supervisionado (mas não exclusivas deste) incluem:

Agrupamento: separar os exemplos em grupos usando alguma medida de similaridade;

Detecção de novidade: identificar os poucos exemplos que se diferem da maioria;

Redução da dimensionalidade: reduzir o número de atributos mantendo características chaves da amostra de treino.

A tarefa de agrupamento é muito importante, de maneira que a seleção do algoritmo de AM adequado a um determinado problema deve ser criteriosa. Pode-se utilizar um conjunto de critérios de admissibilidade para se comparar os diferentes algoritmos de agrupamento. Esses critérios são baseados em como os agrupamentos são formados, na estrutura dos dados e na sensibilidade a mudanças da técnica de agrupamento, de maneira que não afetem a estrutura dos dados (Fisher e Ness, 1971).

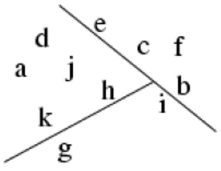
Não existe uma técnica de agrupamento universal. Devido a isso, deve-se avaliar entre as técnicas existentes qual se adapta melhor ao problema. As diferentes técnicas de agrupamento são classificadas de acordo com o tipo de resultado obtido pelo algoritmo, conforme ilustrado na Figura 2.1 e descrito a seguir (Martins, 2003):

Otimização: Visa encontrar uma k-partição ótima. Dado um valor k, o conjunto de dados é dividido em k agrupamentos, mutuamente exclusivos. As técnicas de otimização costumam ser caras, pois normalmente fazem buscas exaustivas pela partição k ótima. Devido a isso, seu uso é restrito a pequenos valores de k.

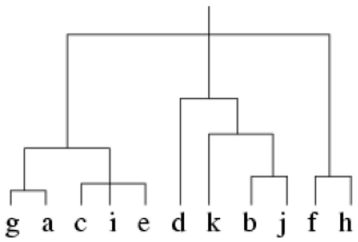
Hierárquica: Criam árvores, onde as folhas representam os exemplos e os nós internos representam os grupos de exemplos. Essa técnica pode ser dividida pela forma como o algoritmo constrói a árvore de maneira aglomerativa ou divisiva. Os algoritmos aglomerativos constroem a árvore de baixo para cima, ou seja, da folha para a raiz. Os algoritmos divisivos a constroem de forma inversa, de cima para baixo. Esses algoritmos tem geralmente menos complexidade computacional que os métodos de otimização.

Probabilística: Essa técnica associa a cada exemplo a probabilidade dele pertencer ou não a um agrupamento. Devido ao fato de não se ter evidências suficientes para saber se um exemplo pertence ou não a um determinado agrupamento, não se tem uma forma determinística de se tomar a decisão de a qual agrupamento o exemplo pertence. Assim, se faz necessário o cálculo das probabilidades.

Aglutinação: Essa técnica retorna grupos onde os exemplos podem se sobrepor, ou seja, pertencer a um ou mais grupos. As técnicas probabilísticas são muito similares e retornam a probabilidade de um exemplo pertencer a cada grupo. Desse modo técnicas probabilísticas podem ser vistas como uma técnica de aglutinação, se a probabilidade for usada para que um mesmo exemplo pertença a vários grupos.



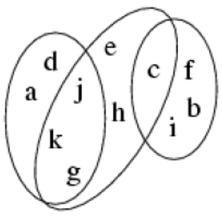
(a)



(b)

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1

(c)



(d)

Figura 2.1: Técnicas de agrupamento: (a) Otimização (b)Hierárquica (c) Probabilística (d) Aglutinação. Martins (2003) com adaptações.

Em suma, o Aprendizado Não Supervisionado é uma forma de aprendizagem na qual não existe um oráculo. O algoritmo procura descobrir sozinho as relações, os padrões, as regularidades ou as categorias nos dados que lhe vão sendo apresentados. Além do Aprendizado Supervisionado e do Aprendizado Não Supervisionado, há uma terceira abordagem

utilizada quando existem alguns dados rotulados e outros dados não rotulados. Nestes casos a alternativa é a utilização do Aprendizado Semissupervisionado, descrito na próxima seção.

2.5 Aprendizado Semissupervisionado

Para lidar com situações na qual estão disponíveis poucos dados rotulados e um grande número de dados não-rotulados, pode-se fazer uso de aprendizado supervisionado e não supervisionado. Entretanto, a quantidade de dados rotulados pode não ser suficiente para a indução de um bom classificador e, em diversos problemas práticos, torna-se caro, difícil ou demorado obter dados rotulados. O aprendizado semissupervisionado possui várias técnicas e ilustra um maquinário sofisticado desenvolvido em várias ramificações do Aprendizado de Máquina, como por exemplo métodos *kernel* ou técnicas Bayesianas, que ajudam a resolver esses problemas de rotulamento.

Por muito tempo o Aprendizado Não Supervisionado e o Aprendizado Supervisionado foram os tipos fundamentais de tarefas que dominaram a área de Aprendizado de Máquina. Contudo, nos últimos anos pesquisadores de Aprendizado de Máquina propuseram diversos tipos de aprendizado e entre eles surgiu o Aprendizado Semissupervisionado (ASS) que veio como uma alternativa para problemas nos quais estavam disponíveis uma grande quantidade de dados não rotulados, porém poucos dados rotulados (Chapelle et al., 2006). Em situações práticas isso ocorre frequentemente devido ao custo ou a dificuldade de rotular uma quantidade expressiva de dados.

Relembrando, o domínio de exemplos é representado por um conjunto X de n_{ex} exemplos (ou pontos). No Aprendizado Não Supervisionado, o objetivo é encontrar uma estrutura relevante em X. Enquanto que no Aprendizado Supervisionado o objetivo é encontrar um mapeamento de um dado x_i a um y_i, dado um conjunto de treinamento com pares $P(x, y)$.

O Aprendizado Semissupervisionado é o meio termo entre os Aprendizados Supervisionado e Não Supervisionado. Além dos dados não rotulados, o algoritmo necessita de alguns dados rotulados previamente. Deste modo, o conjunto X pode ser dividido em duas partes: os exemplos $X_l := (x_1, ..., x_l)$, para os quais os rótulos $Y_l := (y_1, ..., y_l)$ são conhecidos, e os exemplos $X_u := (x_{l+1}, ..., x_{l+u})$, para os quais os rótulos não são conhecidos.

O ASS pode ser visto como um Aprendizado Não Supervisionado guiado por restrições, ou como um Aprendizado Supervisionado com informações adicionais na distribuição dos exemplos X. O ASS pode ser visto ainda como um Aprendizado Supervisionado na qual o objetivo é prever um valor alvo para um x_i dado. Entretanto, esse tipo de abordagem não se aplica quando o número e/ou a natureza das classes não é conhecida previamente e tem que ser deduzida a partir dos dados (Chapelle et al., 2006).

Um outro conceito importante relacionado ao ASS é conhecido como *transductive le-*

arning, ou aprendizado transdutivo (Vapnik, 1998). De acordo com Vapnik (1998), a indução requer a solução de um problema mais geral (inferindo uma função) antes de resolver um problema mais específico (classificando novos exemplos). A ideia da transdução é realizar previsões somente para os casos de testes, ou seja, somente para os exemplos já conhecidos (um conjunto finito). Isso contrasta com o *inductive learning*, ou aprendizado indutivo, onde o objetivo é retornar uma função de indução que é definida para todo o espaço X (um conjunto virtualmente infinito de exemplos).

Há ainda alguns conceitos que devem ser levados em consideração quando se fala de ASS. Embora muitos métodos não derivam explicitamente dos conceitos a seguir, a maioria dos algoritmos podem ser vistos como correspondentes ou implementações de um ou mais deles (Chapelle et al., 2006):

Hipótese da Continuidade do Aprendizado Semissupervisionado

A hipótese da continuidade do ASS define que *se 2 exemplos (pontos)* x_1 *e* x_2 *estão próximos em uma região de alta densidade, então as saídas (rótulos)* y_1 *e* y_2 *correspondentes também devem estar próximas.* Sem essa hipótese não seria possível generalizar de um conjunto de treinamento finito para um conjunto possivelmente infinito de casos de testes não vistos. Uma consequência imediata disso é que se dois exemplos estão ligados por um caminho de alta densidade, ou seja, pertencem a um mesmo agrupamento, então o resultado deve estar próximo. Se, por outro lado, eles estão separados por uma região pouco densa, então seus resultados não precisam estar próximos.

Hipótese do Agrupamento

Suponha que se sabe que os exemplos de rótulos distintos tendem a formar um agrupamento. Então os dados não rotulados podem ajudar a encontrar um limite para cada agrupamento com maior precisão. Por exemplo, pode-se executar um algoritmo de agrupamento e usar os exemplos rotulados para definir o rótulo de cada agrupamento. Isso é de fato uma das primeiras formas de ASS. Disso tem-se que: *Se os exemplos estão no mesmo agrupamento, então é provável que eles tenham o mesmo rótulo.* Note também que o limite que divide os rótulos deve encontrar-se em uma região de baixa densidade. Esse conceito pode ser visto como um caso especial do conceito anterior, considerando que agrupamentos são frequentemente definidos como conjunto de exemplos que podem ser conectados por pequenas curvas que cruzam apenas regiões de alta densidade.

Hipótese do Desdobramento

A hipótese do desdobramento supõe que apesar dos dados terem centenas de atributos, alguns conjuntos de dados podem ser representados em um espaço de menor dimensão, pois sua dimensionalidade é artificialmente alta. Um exemplo é ilustrado na Figura 2.2, onde os dados estão em um espaço tridimensional formando um rolo

suíço. Nesta forma, dependendo de como os dados são observados, os rótulos parecerão misturadas, mas se estes mesmos dados forem mapeados para duas dimensões, a separação entre as rótulos é linear.

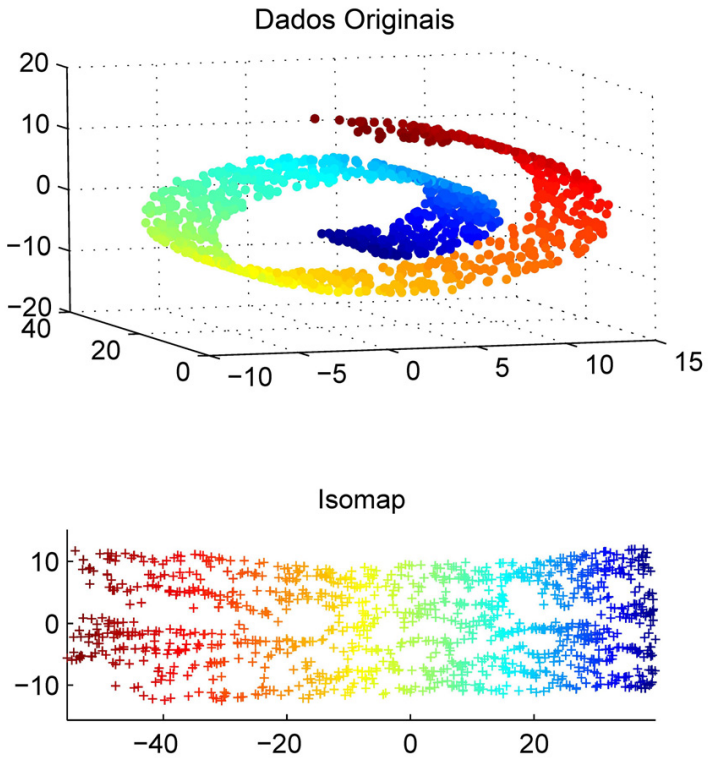


Figura 2.2: Rolo suíço (acima) e sua representação bidimensional (abaixo). Cayton (2005) com adaptações.

Transdução

No aprendizado transdutivo o objetivo é propagar os rótulos das instâncias rotuladas para as demais usando inferência. Neste tipo de aprendizado a preocupação é classificar os dados já existentes, sem supor a apresentação de novos dados, fazendo assim um mapeamento em um domínio menor de dados (e não em um domínio tido como infinito, que é o padrão). Note que a transdução não é o mesmo que ASS, alguns algoritmos semissupervisionados são transdutivos, porém outros são indutivos.

Com base nessas hipóteses foram criadas várias classes de algoritmos do ASS, como modelos gerativos, separação de baixa densidade e métodos baseados em grafos. Algumas das áreas de aplicação do ASS incluem: reconhecimento da fala, processamento automático de páginas da internet, solução de sequência de proteínas, entre outras.

2.6 Considerações Finais

A maioria dos métodos de aprendizado semissupervisionadas consistem em uma variação dos métodos de Aprendizado de Máquina tradicionais. Muitos desses métodos tentam

obter, de alguma maneira, um ganho sobre os poucos exemplos rotulados. Normalmente, esse ganho é obtido com algum método de escolha dos exemplos a serem rotulados. Esse método é de vital importância para o aprendizado semissupervisionado, pois, uma vez que um "erro" for introduzido ao rotular um novo exemplo, esse erro pode se propagar nas próximas iterações. Quando o erro acumula-se, tem como consequência um aumento no erro do classificador induzido por um algoritmo de aprendizado supervisionado que utiliza os exemplos rotulados pelo algoritmo semissupervisionado.

O próximo capítulo faz uma revisão bibliográfica sobre Aprendizado Semissupervisionado, Aprendizado Ativo e Filtros de Rede, assim como uma explicação de métodos utilizados no trabalho.

Capítulo 3

Revisão Bibliográfica e Métodos utilizados

Métodos diferentes são usados em conjunto de maneira a atingir o objetivo deste trabalho. Neste capítulo é feita uma revisão bibliográfica dos trabalhos relevantes de cada área, bem como o atual estado da arte. Ainda, neste capítulo são descritos alguns métodos usados neste trabalho e na avaliação experimental.

Organização

Na Seção 3.1 é feita a revisão sobre o Aprendizado Semissupervisionado e na Seção 3.2 sobre o Aprendizado Ativo. A seguir na Seção 3.3 são explicados os conceitos de Análise ROC usados na avaliação dos resultados dos experimentos. Na Seção 3.4 é descrito a Seleção de Atributos, método que busca otimizar os dados e acelerar a classificação. Na Seção 3.5 é apresentado o estado da arte sobre Filtros de Rede e na Seção 3.6 são apresentados os conceitos sobre *Ranking*. Por último na Seção 3.7 é explicado o funcionamento das *bag of words*, utilizadas neste trabalho.

3.1 Aprendizado Semissupervisionado

O auto-aprendizado (*self-learning*), também conhecido como auto-treinamento, auto-rotulamento ou aprendizado dirigido por decisão, surgiu entre os anos de 60 e 70. O auto-aprendizado foi provavelmente o primeiro algoritmo a utilizar dados não rotulados na classificação (Weston, 2008).

O auto-aprendizado consiste em um envoltório algorítmico que usa um método de aprendizado supervisionado repetidamente. O algoritmo de aprendizado inicia o treinamento utilizando os dados rotulados disponíveis e a cada iteração uma parte dos dados não rotulados é rotulada e acrescentada ao conjunto de dados rotulados. Então o método supervisionado é re-treinado usando esses novos dados rotulados adicionais e o processo se repete até que todos os dados sejam rotulados. Note que o algoritmo usa suas próprias

predições para o treinamento, o que pode fazer com que um erro de classificação seja propagado nas outras iterações. Alguns algoritmos tentam anular este efeito "desrotulando" os pontos erroneamente rotulados (Lallich et al., 2002).

O auto-aprendizado tem sido utilizado em problemas de mineração de texto. Por exemplo, Yarowsky (1995) utilizou o auto-aprendizado para desambiguar várias palavras da língua inglesa como *"plant"*, que dependendo do contexto pode ter o sentido de vegetal ou uma instalação industrial. Maeireizo et al. (2004) utilizou o algoritmo Co-Training para classificar diálogos como "emotivos" ou "não-emotivos". Rosenberg et al. (2005) demonstrou como o aprendizado semissupervisionado pôde obter resultados comparáveis ao estado da arte na detecção de objetos em imagens. Li et al. (2008) utilizaram um SVM auto-treinado para executar correções em uma interface de computador baseada em eletroencefalograma (*EEG - Electroencephalography*). Um problema no auto-aprendizado é a convergência, que depende do algoritmo utilizado, mas para alguns algoritmos básicos existem estudos da convergência, como em Haffari e Sarkar (2007).

Vapnik (1998) introduziu o conceito de inferência transdutiva, ou simplesmente transdução, ao criar uma implementação semissupervisionada do SVM. O SVM por si só é um conceito muito próximo ao aprendizado semissupervisionado, porém Vapnik (1998) foi além e criou o algoritmo semissupervionado chamado TSVM (*Transductive support vector machines*), que é a versão transdutiva do SVM (*Support Vector Machines*).

O interesse em Aprendizado Semissupervisionado aumentou na década de 1990, principalmente devido às aplicações em problemas de linguagem natural e classificação de texto (Yarowsky (1995); Kamal Nigam (1998); Blum e Mitchell (1998); Collins e Singer (1999); Joachims (1999)). Pela revisão bibliográfica realizada, e também apontado por outros pesquisadores, Merz et al. (1992) foram os primeiros a usar o termo "semissupervisionado" para a classificação com dados rotulados e não rotulados.

O Aprendizado Semissupervisionado é particularmente útil para tarefas de classificação de textos que envolvem fontes de dados on-line tais como páginas de internet, e-mails, blogs e notícias. A classificação de conteúdo tem grande importância dado o grande volume de materiais disponíveis on-line. No trabalho desenvolvido por Matsubara (2004), é descrito o uso do algoritmo de Aprendizado Semissupervisionado Co-Training e são apresentados várias extensões do algoritmo, as quais implementou. Os resultados mostraram que o Co-Training é afetado de maneira complexa pelos diferentes aspectos na indução dos modelos.

Braga (2010) criou um algoritmo chamado CoAL que aperfeiçoa o algoritmo Co-Training. A implementação de Braga (2010) incorpora aprendizado online como uma maneira de tratar pontos de contenção e usa unigramas e bigramas como duas descrições distintas de bases de textos. Dessa forma obtém-se uma melhora significativa no desempenho do Aprendizado Semissupervisionado.

Shi et al. (2011) propôs uma nova técnica de classificação de textos de classe binária com aprendizado semissupervisionado, no qual só há exemplos rotulados como positivo e

exemplos não rotulados. Wajeed e Adilakshmi (2011) desenvolveram um algoritmo KNN melhorado que mostrou ter uma melhor performance no aprendizado semissupervisionado. Sun et al. (2012) apresentaram uma árvore de agrupamento semissupervisionada usando aprendizado ativo aplicado ao problema de classificação de textos da internet, no qual usa-se a estratégia de agrupar textos similares para questionar a classe dos mesmos em lote.

Na próxima seção é abordado o Aprendizado Ativo que trata sobre como se pode melhorar a classificação através do rotulamento de alguns exemplos pelo moderador.

3.2 Aprendizado Ativo

O *Active Learning*, ou Aprendizado Ativo, é uma subárea de Aprendizado de Máquina e, de forma mais geral, da Inteligência Artificial. A hipótese chave dessa técnica é que o algoritmo de aprendizado pode escolher alguns dados, os quais o algoritmo aprende a questionar e, consequentemente, terá uma performance melhor com um menor treinamento. Para verificar o quão desejado é essa propriedade do algoritmo de aprendizado, deve-se considerar que, para qualquer sistema de aprendizado supervisionado funcionar bem, ele deve ter centenas (ou milhares) de exemplos rotulados. Às vezes, esses rótulos tem baixo ou nenhum custo, como o rótulo de "spam" que se marca em mensagens de e-mail indesejadas, ou as 5 estrelas que pode-se atribuir a filmes em uma rede social. Sistemas de aprendizado usam esses rótulos e estrelas para filtrar melhor o lixo de e-mail ou sugerir filmes preferidos. Nesses casos, recebe-se os rótulos sem custos, mas para muitas outras tarefas de aprendizado supervisionado, exemplos rotulados são muito difíceis, consomem muito tempo ou são muito custosos de se obter. Veja alguns exemplos (Settles, 2009):

- *Reconhecimento de Fala*: Rotulamento preciso de declarações de fala consome muito tempo e requer linguistas treinados.
- *Extração de Informações*: Sistemas de extração de informações ideais devem ser treinados usando documentos rotulados com anotações detalhadas.
- *Classificação e filtragem*: Para aprender a classificar documentos (ex.: artigos ou páginas web) ou qualquer outro tipo de mídia (ex.: arquivos de imagem, áudio e vídeo) é necessário que os usuários rotulem cada documento ou arquivo de mídia com um rótulo específico. Esse processo de rotulação torna-se tedioso e cansativo quando se tem milhares de exemplos.

Os Sistemas de Aprendizado Ativo tentam driblar este obstáculo através de consultas ao usuário na forma de exemplos não rotulados para que sejam rotulados. Dessa forma, o **Aprendiz Ativo** visa atingir altas taxas de acerto usando o mínimo de exemplos rotulados possível, assim minimizando o custo da obtenção de dados rotulados. O Aprendizado Ativo possui motivação em vários problemas de aprendizado de máquina modernos, nos

quais o conjunto de dados é abundante, porém o número de exemplos rotulados é escasso ou caro para obter.

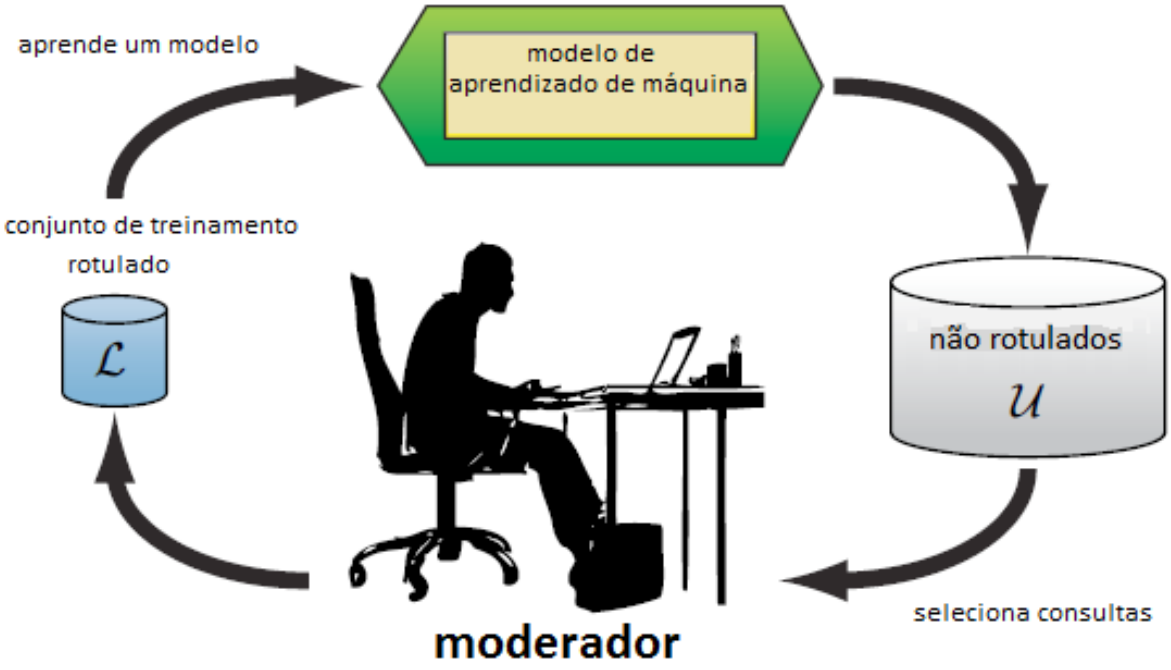


Figura 3.1: Ciclo baseado em consultas do Aprendizado Ativo. Adaptado de Settles (2009).

A Figura 3.1 ilustra o ciclo de Aprendizado Ativo baseado em consultas. O aprendiz pode começar com um número pequeno de exemplos no conjunto de treinamento rotulado $\mathcal{L}$. A partir do conjunto de treinamento $\mathcal{L}$, o aprendiz induz um modelo para classificação que vai recebendo e classificando exemplos não rotulados. A partir dos novos exemplos não rotulados ele cria um conjunto u. Dentre os elementos do conjunto u o aprendiz selecionará cuidadosamente alguns exemplos. Destes exemplos selecionados o aprendiz solicitará o rótulo ao moderador. Então o novo exemplo rotulado pelo moderador é simplesmente adicionado ao conjunto $\mathcal{L}$ reiniciando o ciclo.

Há diversos casos em que o **Aprendiz Ativo** pode consultar o moderador. Desses casos há diversas estratégias diferentes usadas para decidir quais exemplos são mais relevantes, ou seja, quais exemplos ajudarão melhor o classificador manter sua corretude e eficiência. Dentre as possibilidades de estratégias, vale ressaltar algumas seleções (Lewis e Catlett, 1994):

- *Por Limiar*: Esse tipo de amostragem leva em consideração quando um exemplo foi classificado muito próximo do limiar que separa a classificação dos rótulos.
- *Por incerteza*: Esse tipo de amostragem é usada quando o modelo acredita que irá rotular incorretamente um exemplo.
- *Por entropia*: A entropia é uma medida de informação-teórica que representa a quantidade de informação necessária para "codificar" uma distribuição. Desse modo,

é frequentemente usada como medida de impureza no aprendizado de máquina.

A seguir é abordado um algoritmo multidescrição de aprendizado ativo.

Co-Testing

O Co-Testing é um algoritmo multidescrição de aprendizado ativo, no qual a ideia é selecionar pontos de contenção e fornecê-los a um oráculo. O oráculo então rotula os exemplos selecionados e eles são adicionados ao conjunto de exemplos rotulados. Esse processo é repetido várias vezes, assim o aprendizado torna-se ativo.

Os pontos de contenção ocorrem quando o mesmo exemplo é rotulado diferente com alta confiança por diferentes classificadores. O Co-Testing usa dois classificadores h_1 e h_2, gerados a partir de visões diferentes do mesmo conjunto de dados, conforme descrito no Algoritmo 1. O algoritmo recebe como entrada os conjuntos das visões de exemplos rotulados L_v e de exemplos não rotulados U_v, onde $v = 1, 2$ que representa o índice da visão, e ainda um inteiro k que define o número máximo de iterações do algoritmo.

Algoritmo 1 Co-Testing

Entrada: L_1, L_2, U_1, U_2, k
  Para $it = 1$ **até** k **faça**
    obter o classificador h_1 a partir de L_1
    obter o classificador h_2 a partir de L_2
    $R_1 =$ todos os exemplos de U_1 rotulados por h_1
    $R_2 =$ todos os exemplos de U_2 rotulados por h_2
    $PC = \{(x_{1i}, x_{2i}) | h_1(x_{1i}) \neq h_2(x_{2i}) \wedge (x_{1i}, h_1(x_{1i})) \in R_1 \wedge (x_{2i}, h_2(x_{2i})) \in R_2\}$
    Se $PC = \emptyset$ **então**
      Retorna h_1, h_2, L_1, L_2
    Fim Se
    $(x_{1i}, x_{2i}) =$ **selecionar**(PC)
    $y_i =$ rótulo, após interrogar o oráculo, atribuído ao par (x_{1i}, x_{2i})
    $L_1 = L_1 \cup \{(x_{1i}, y_i)\}$
    $L_2 = L_2 \cup \{(x_{2i}, y_i)\}$
    $U_1 = U_1 - \{(x_{1i}, ?)\}$
    $U_2 = U_2 - \{(x_{2i}, ?)\}$
  Fim Para
  Retorna h_1, h_2, L_1, L_2

Primeiramente os classificadores h_v são induzidos utilizando, respectivamente, os conjuntos L_v de exemplos rotulados. Em seguida é construído o conjunto PC que contém os exemplos que foram rotulados com classes diferentes em cada descrição. Se $PC = \emptyset$ o processo termina, caso contrário, no próximo passo, a função **seleciona** é responsável pela escolha do exemplo a ser submetido ao oráculo para que ele forneça o rótulo correspondente. Os exemplos x_{vi}, onde i é o índice dos exemplos com $i = 1, .., n_{\text{ex}}$, rotulados pelo oráculo são retirados de U_v e inseridos, juntamente com o rótulo dado, nos respectivos conjuntos L_v de exemplos rotulados. O processo é repetido até k vezes, caso não se verifique $PC = \emptyset$.

O Co-Testing foi proposto para ser usado com qualquer algoritmo-base de aprendizado, inclusive por algoritmos que não dão um valor de confiança para classificação. Nesse caso, a função **selecionar** escolhe um exemplo aleatoriamente do conjunto PC. Caso seja utilizado um algoritmo-base que dá um valor de confiança (*score*) para cada possível valor da classe, pode-se então escolher o exemplo que foi classificado com *score* máximo em classes distintas. De qualquer maneira, o objetivo em cada iteração é selecionar o exemplo mais informativo para o oráculo rotular.

Observe que os exemplos rotulados com a mesma classe pelos classificadores não são adicionados em L_v durante as iterações do algoritmo. Somente os exemplos que foram rotulados pelo oráculo são adicionados a L_v.

Na próxima seção é explicado o que é a Análise ROC e como essa Análise pode ajudar a medir a qualidade dos resultados.

3.3 Análise ROC

A geração de gráficos *Receiver Operating Characteristics* (**ROC**), ou Características de Funcionamento do Receptor, é uma técnica para visualização, organização e seleção de classificadores baseado em sua performance. Os gráficos ROC eram usados inicialmente em processos de decisão médica, e posteriormente passou-se a utilizar também em aprendizado de máquina e em pesquisas de mineração de dados (Fawcett, 2006). Na Figura 3.2 é ilustrado um gráfico ROC.

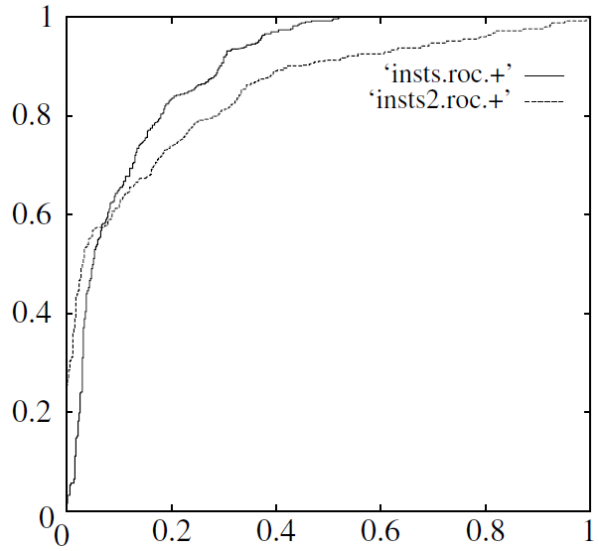


Figura 3.2: Curvas ROC. Fawcett (2006) com adaptações.

Um dos primeiros pesquisadores a usar gráficos ROC no aprendizado de máquina foi Spackman (1989), que demonstrou o valor das curvas ROC na validação e comparação de algoritmos. Nos anos mais recentes houve um aumento significativo no uso de gráficos ROC na comunidade de aprendizado de máquina, devido, em parte, ao fato de que a exatidão da classificação simples é frequentemente uma métrica pobre para analisar per-

formance (Fawcett e Provost, 1997; Provost e Fawcett, 1998). Além de ser um método útil de plotagem de performance, a curva ROC possui propriedades que a torna especialmente útil para domínios onde há uma distribuição de classes assimétricas e custos de erros de classificação diferentes.

A Análise ROC, na sua forma mais utilizada, trata apenas problemas de duas classes. Assim, dado um classificador binário que classifica exemplos em positivos e negativos, e um conjunto de exemplos rotulados, ao predizer a classe dos exemplos desse conjunto e compará-los com a classe verdadeira, pode-se distinguir quatro diferentes situações:

- **Verdadeiro Positivo** (VP): exemplo positivo predito como positivo
- **Falso Positivo** (FP): exemplo negativo predito como positivo
- **Verdadeiro Negativo** (VN): exemplo negativo predito como negativo
- **Falso Negativo** (FN): exemplo positivo predito como negativo

Quando um conjunto de exemplos é classificado, pode-se contar quantos exemplos se enquadram em cada uma dessas categorias. Com essa contagem pode-se compor uma matriz de contingência, ilustrada na Tabela 3.1, onde *Pos* é o total de exemplos positivos e *Neg* é o total de exemplos negativos (Matsubara, 2008).

Tabela 3.1: Matriz de contingência

	preditos positivos	preditos negativos	Total
exemplos positivos	*VP*	*FN*	*Pos*
exemplos negativos	*FP*	*VN*	*Neg*

Com esses valores é possível definir as seguintes medidas:

$$\text{taxa de verdadeiros positivos } (TVP) = \frac{VP}{Pos} \tag{3.1}$$

$$\text{taxa de falsos positivos } (TFP) = \frac{FP}{Neg} \tag{3.2}$$

Com as medidas *TVP* e *TFP* é possível desenhar o gráfico ROC. O gráfico ROC é um gráfico bidimensional, também conhecido como *espaço ROC*, no qual os eixos Y e X do gráfico sempre representam as medidas *TVP* e *TFP*, respectivamente. O gráfico serve para comparar diferentes classificadores de acordo com suas medidas *TVP* e *TFP*.

Quando a curva ROC passa pelo ponto (0,1) diz-se que há uma classificação perfeita. Essa condição ocorre quando todos os exemplos positivos e negativos são rotulados corretamente. Essa condição também é chamada de **Céu ROC**. O inverso desta condição

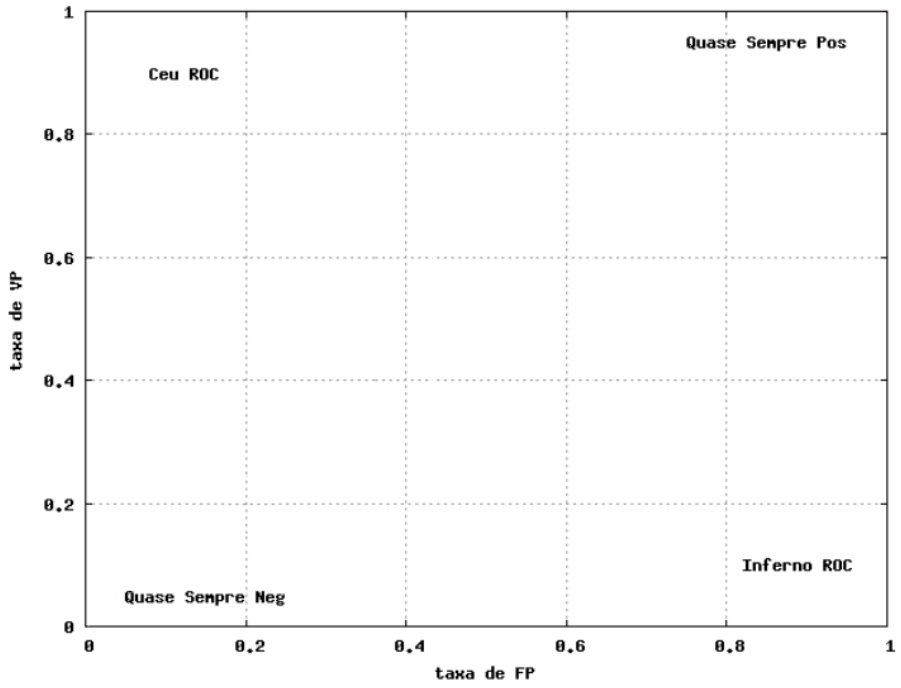


Figura 3.3: Extremos da curva ROC. Adaptado de Matsubara (2008).

seria o **Inferno ROC**, no qual a curva passaria pelo ponto (1,0) e todos os exemplos foram classificados incorretamente. No entanto, classificadores representados nessa região possuem informações com capacidade de distinguir as classes, mas não as utilizam corretamente (Flach e Wu, 2005). Essas condições são ilustradas pela Figura 3.3.

A curva ROC é concebida começando pelo ponto (0,0) do gráfico e a cada exemplo, deve-se desenhar de acordo com a classe do exemplo:

- Exemplo Positivo: reta de tamanho $1/Pos$ para cima;
- Exemplo Negativo: reta de tamanho $1/Neg$ para direita;

Quando um exemplo negativo aparece antes de um exemplo positivo, a curva do gráfico ROC, ou simplesmente curva ROC, vai em direção à direita prematuramente. Quando isso ocorre, o exemplo negativo causa um aumento na área sobre a curva, afastando a curva do ponto ideal (0,1) (Matsubara, 2008).

Outro conceito chave da análise ROC é conhecido por AUC (*Area under a ROC curve*, ou área abaixo da curva ROC). Como seu nome já diz, o valor da AUC é o cálculo da área abaixo da curva ROC. Como a AUC é uma porção de área de unidade quadrada, seu valor estará sempre entre 0 e 1,0. Entretanto, como valores randômicos produzem uma linha diagonal entre (0,0) e (1,1), o que resulta em uma área de 0,5, classificadores com este valor aproximado de AUC são equivalentes a classificadores aleatórios. A Figura 3.4 contém um curva tracejada representando valores randômicos. O valor da AUC tem uma propriedade estatística importante: o valor AUC de um classificador é equivalente a probabilidade que esse classificador irá classificar uma instância positiva escolhida aleatoriamente maior que uma instância negativa escolhida aleatoriamente (Fawcett, 2006).

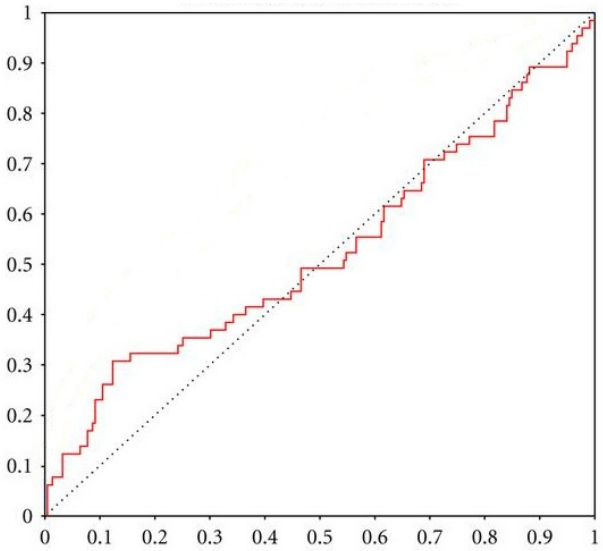


Figura 3.4: Exemplos de curvas ROC. A curva tracejada representa um classificador randômico e a outra representa um classificador qualquer.

Na próxima seção é abordada a Seleção de Atributos, que visa melhorar a classificação escolhendo os atributos mais relevantes.

3.4 Seleção de Atributos

A Seleção de Atributos pode ser dividida em três classes: *wrappers*, métodos embarcados e filtros (Guyon e Elisseeff, 2003). Os *wrappers* usam um método de aprendizado como uma caixa preta, apenas avaliam cada atributo de acordo com o seu poder de predição. O método define um espaço de busca de maneira a selecionar um subconjunto com os atributos com maior predição. Os Filtros selecionam atributos aplicando alguma métrica relacionada a classe e seleciona os atributos como um passo de pré-processamento. Os Métodos Embarcados fazem uma pontuação de atributos definida durante a fase de aprendizado de alguns métodos de aprendizado, como em árvores de decisão.

Guyon e Elisseeff (2003) dizem que "métodos sofisticados embarcados ou de *wrapper* aumentam a performance do preditor em comparação com métodos simples de comparação de *ranking* de variáveis como métodos de correção, mas as melhoras nem sempre são significativas". A razão para isso está relacionada com a maldição da dimensionalidade (*curse of dimensionality*), que leva ao *overfitting* dos dados nos métodos multivariados. Isso sugere que métodos simples podem evitar a maldição da dimensionalidade pelo seus bias forte.

Na Figura 3.5 pode-se ver um exemplo simples que ilustra a motivação do uso de seleção de atributos. Nesse exemplo, inicialmente há duas dimensões com exemplos de duas classes distintas (estrelas e triângulos), porém, fica difícil de perceber o que define cada classe. Entretanto, após análise um dos atributos é descartado, restando apenas uma dimensão e ficando claro o agrupamento de cada classe. Nesse exemplo simples, o

número de dimensões foi reduzido a metade, porém, em muitos casos reais de classificação de textos, onde há dezenas de milhares de atributos, se deseja uma redução ainda maior no número de atributos.

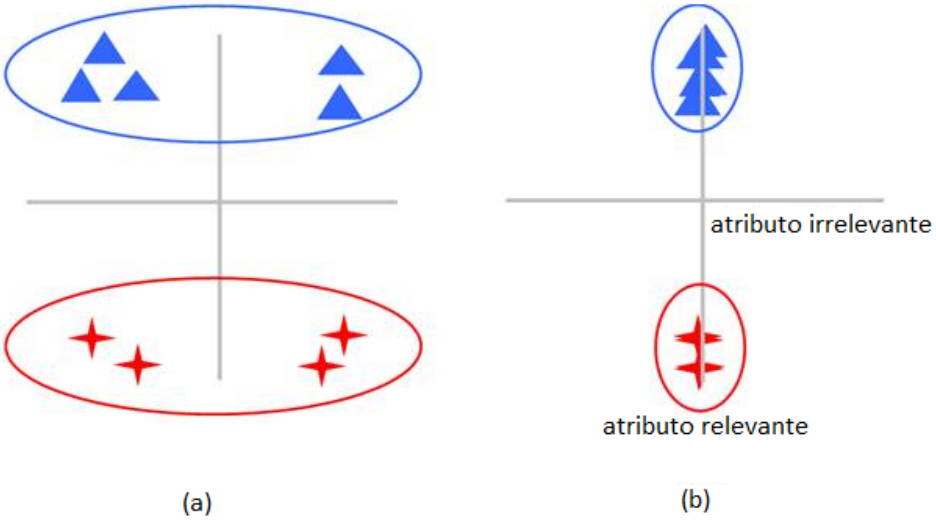


Figura 3.5: Ilustração da Motivação de Seleção de Atributos. Os triângulos e as estrelas pertencem a dois grupos diferentes em um espaço de duas dimensões. Em (a), os dois grupos não podem ser distinguidos. Além disso, os pontos do mesmo grupo parecem um pouco longes. Em contraste, em (b), após descartar o atributo horizontal, a estrutura dos agrupamentos se torna muito mais óbvia, assim o agrupamento será muito mais fácil e mais preciso. (Adaptado de (Zeng e Cheung, 2009))

Dada a natureza dinâmica da aplicação, a tarefa de classificação deve ser o mais eficiente possível. Entretanto, o domínio de classificação de textos é conhecido como um domínio de alta dimensão. Não é incomum que problemas de classificação de texto atinjam dezenas de milhares de atributos. Uma alta contagem de atributos pode deixar consideravelmente lento tanto o treinamento quanto a classificação. A seleção de atributos se torna então uma estratégia não apenas requerida como necessária no domínio de classificação de textos (Guyon e Elisseeff, 2003).

Há um método de seleção de atributos muito difundido que usa o coeficiente χ^2 chamado chi-quadrado (Chi2, em inglês *chi-square*). O chi-quadrado é um valor da dispersão para duas variáveis de escala nominal, usado em alguns testes estatísticos. Ele diz em que medida os valores observados se desviam do valor esperado (caso as duas variáveis não estivessem correlacionadas). Quanto maior o chi-quadrado, mais significativa é a relação entre a variável dependente e a variável independente.

Porém, de acordo com Forman (2003), uma métrica de seleção de atributos chamada *Bi-Normal Separation* (BNS) pode dar bons resultados no domínio de classificação de textos e também em conjunto de dados oblíquos (*skewed datasets*). Essa métrica é definida pela Equação 3.3:

$$F^{-1}(TVP) - F^{-1}(TFP) \tag{3.3}$$

onde F^{-1} é a função de probabilidade cumulativa inversa da distribuição Normal padrão (z-score).

Para uma melhor compreensão, observe a Figura 3.6. Agora suponha a existência de um dado atributo em cada exemplo modelado pelo ocorrência de uma variável Normal aleatória excedendo um limiar hipotético. A taxa de predomínio do atributo corresponde a área abaixo da curva após o limiar. Se o atributo é mais predominante na classe positiva, então seu limiar está mais longe da cauda da distribuição do que a classe negativa. A métrica BNS mede a separação entre esses dois limiares.

Figura 3.6: Duas visões da Separação Bi-Normal usando a distribuição de probabilidade Normal: (esquerda) Separação de limiares. (direita) Separação de curvas (análise ROC). (Forman, 2003)

Uma visão alternativa é motivada pela análise do limiar da curva ROC: A medida mede a separação horizontal entre duas curvas Normais padrão, onde a posição relativa delas é unicamente prescrita pela TVP e TFP, a área abaixo da cauda de cada curva (comparado com um teste de hipótese tradicional onde TVP e TFP estimam o centro de cada curva). A distância métrica do BNS é assim proporcional a área abaixo da curva ROC gerada por duas curvas Normais sobrepostas. Esse é um método robusto que tem sido usado no campo de testes médicos para ajustar curvas ROC aos dados de maneira a determinar a eficácia de um tratamento. As justificativas na literatura médica são muitas e diversas, como ajustar bem ambos os dados reais e artificiais que violam a suposição Normal (Forman, 2003).

Nas Tabelas 3.2 e 3.3 podem ser vistos dois exemplos de uso de seleção de atributos. Cada uma das tabelas mostra a análise de um atributo dos exemplos (no caso, gato e cachorro). Na primeira coluna são mostrados os valores do atributo analisado. Na segunda coluna tem-se o valor predito, pelo seletor de atributo, para a classe do exemplo mapeado diretamente pelo valor do atributo. Na última coluna é exibido o valor da classe real do exemplo.

A partir da Tabela 3.2, tem-se os valores de $TVP = 4/5$ e $TFP = 1/5$. Com isso aplicando a Equação 3.3, tem-se o seguinte:

$$\begin{aligned} BNS(\text{atributo gato}) &= |F^{-1}(4/5) - F^{-1}(1/5)| \\ &= |0,84 - -0,84| \\ &= 1,68 \end{aligned}$$

Tabela 3.2: Exemplo de classificação utilizando atributo gato

atributo gato	predito	classe
1	pos	pos
1	pos	pos
1	pos	pos
1	pos	pos
1	pos	neg
0	neg	pos
0	neg	neg
0	neg	neg
0	neg	neg
0	neg	neg

Tabela 3.3: Exemplo de classificação utilizando atributo cachorro

atributo cachorro	predito	classe
1	pos	pos
1	pos	pos
1	pos	pos
1	pos	neg
1	pos	neg
0	neg	pos
0	neg	pos
0	neg	neg
0	neg	neg
0	neg	neg

Na Tabela 3.3, é exibido outro exemplo e a partir dela se obtêm os valores de $TVP = 3/5$ e $TFP = 2/5$. Com isso, aplicando novamente a Equação 3.3, é obtido o seguinte:

$$\begin{aligned} BNS(\text{atributo cachorro}) &= |F^{-1}(3/5) - F^{-1}(2/5)| \\ &= |0,25 - -0,25| \\ &= 0,5 \end{aligned}$$

Assim, pode-se concluir que o atributo gato mapeia melhor a classe do exemplo, pois possui um maior valor BNS. Sendo assim, é melhor manter o atributo gato do que o atributo cachorro nos exemplos.

A próxima seção trata sobre os Filtros de Rede e como essa tecnologia é utilizada para filtrar e bloquear conteúdo indesejado.

3.5 Filtros de Rede

Antes de existirem filtros de rede, no final dos anos 80 com a formação da ARPANET e, posteriormente, da Internet, foram criados os *firewalls* que separavam a rede interna

corporativa da rede externa, visando dar segurança. O objetivo do *firewall* era permitir que os usuários internos pudessem acessar a rede de fora, mas não permitisse que usuários externos acessassem a rede interna. Com a evolução dos *firewalls* surgiram os *proxy*s.

O *proxy* é um servidor que atende a requisições, repassando os dados dos usuários à frente. Um usuário conecta-se a um servidor *proxy* requisitando algum serviço, como um arquivo, conexão, sítio, ou outro recurso disponível em outro servidor. Um servidor *proxy* pode, opcionalmente, alterar a requisição do cliente ou a resposta do servidor e disponibilizar o recurso solicitado sem precisar se conectar ao servidor especificado, por armazenar o recurso em forma de *cache*. Porém, o recurso do *proxy* que interessa a este trabalho é o da filtragem. Um *proxy* pode fazer filtragem de todos os pacotes ou apenas de determinadas aplicações.

A filtragem por pacotes analisa individualmente os pacotes à medida que são transmitidos, verificando as informações da camada de enlace e da camada de rede (camadas 2 e 3 do modelo ISO/OSI [1], respectivamente). Esse tipo de filtro somente se preocupa com o endereço de origem e destino, qual protocolo está usando e qual a porta. Ele não pode distinguir qual conteúdo está passando. Um administrador desejando proibir o uso de uma aplicação, poderia bloquear a porta padrão usada por essa aplicação, porém muitas aplicações hoje em dia permitem que a porta a ser usada seja modificada, burlando as proibições do *firewall* (Alecrim, 2004).

A filtragem por aplicação possui a vantagem de entender certas aplicações e protocolos como FTP, DNS e HTTP (navegação web). Ela pode detectar se um protocolo não desejado está espreitando em uma porta não-padrão, ou se algum protocolo está abusando de alguma forma prejudicial, ou seja, está sobrecarregando a rede além do normal prejudicando outras conexões (Alecrim, 2004).

Além desses dois modos de filtragem oferecidos pelo *proxy*, hoje em dia ainda existem outros modos de moderação de acesso, como listas negras de URL ou DNS, filtragem por expressões regulares em URL's, filtragem de conteúdo por palavra chave, entre outros.

Atualmente *proxys* usam bancos de dados padrões (expressões regulares) de URL associados com os atributos do conteúdo. Esses bancos de dados são usados pelo *proxy* para filtragem de conteúdo. Empresas de filtragem web fornecem esses bancos de dados com atualizações semanais através de uma assinatura online. O administrador instrui o *proxy* a banir classes de conteúdos (como esportes, pornografia, compras online, jogos e redes sociais) baseado nesses bancos de dados. As requisições que combinam com um padrão de URL contida em uma classe banida desses bancos de dados são rejeitadas imediatamente (Cohen et al., 1998).

Caso o conteúdo acessado pertença a uma classe de dados aceita, então o conteúdo é carregado pelo *proxy* e fornecido ao usuário. Até aqui a filtragem só foi aplicada a URL. Após esse passo poderia ser feita uma análise do conteúdo do sítio, para verificar se

[1] ISO foi uma das primeiras organizações a definir formalmente uma forma comum de conectar computadores. Sua arquitetura é chamada OSI (*Open Systems Interconnection*) ou Interconexão de Sistemas Abertos.

parte do conteúdo deveria ser banido. Nos aplicativos disponíveis comercialmente hoje, se uma organização quiser, por exemplo, bloquear todas as informações relacionadas a "futebol", então o filtro de conteúdo pode ser ativado para bloquear apenas essa palavra em particular, sem levar em consideração conteúdos relacionados.

A próxima seção trata sobre como classificar os dados ou sítios com o uso de algoritmos de *Ranking*.

3.6 *Ranking*

Durante anos, o problema de classificação tem sido foco de pesquisas em aprendizado de máquina. Nesse tipo de aprendizado, muitas vezes, ignora-se a informação do valor de confiança da classificação proposta pelo algoritmo, importando-se apenas com o valor da classe. Porém, o valor de confiança é um dado importante na geração de *rankings* e, dependendo do domínio da aplicação, o *ranking* torna-se mais atrativo que a própria classificação. Um exemplo disso são as páginas de internet retornadas por um buscador. Nesse caso, mais interessante que retornar os endereços de internet relevantes, é retornar esses endereços por ordem de relevância (Matsubara, 2008).

Considere um conjunto com 10 exemplos positivos e 10 exemplos negativos e com 2 atributos A_1 e A_2 ilustrados na Figura 3.7. Ao induzir uma árvore de decisão utilizando esse conjunto de treinamento obtém-se a árvore ilustrada na Figura 3.8.

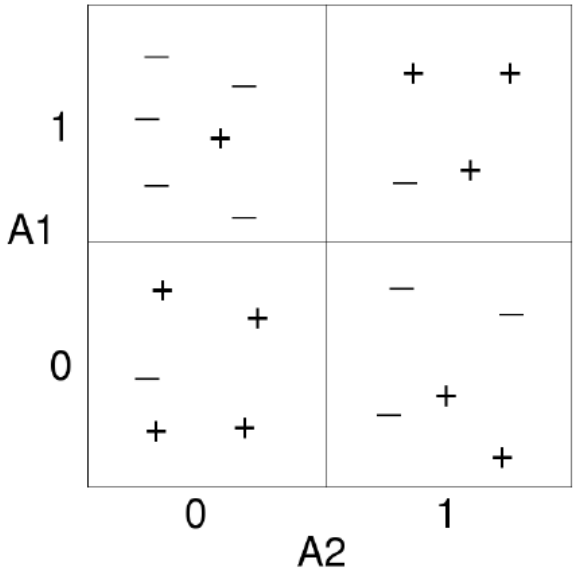


Figura 3.7: Conjunto de treinamento (Matsubara, 2008)

Ferri et al. (2002) e Provost e Domingos (2003) mostram que árvores de decisão podem ser utilizadas como estimadores de probabilidades, calculados conforme a equação 3.4, onde n_i^+ é o número de exemplos positivos e n_i^- é o número de exemplos negativos pertencentes a i-ésima folha. Desse modo, as probabilidades das folhas serem positivas na árvore ilustrada na Figure 3.8 são 0.80, 0.40, 0.16, 0.75 para as folhas 1, 2, 3 e 4,

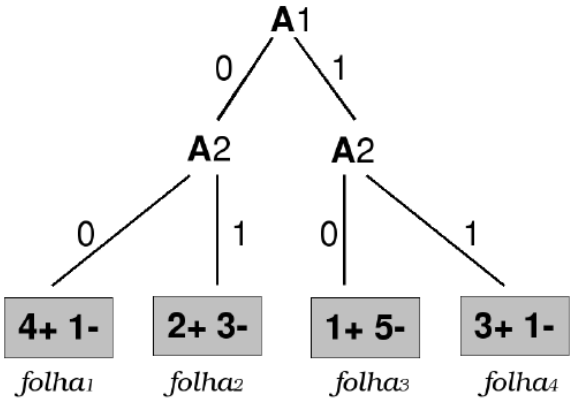


Figura 3.8: Árvore de decisão (Matsubara, 2008)

respectivamente.

$$P(+|folha_i) = \frac{n_i^+}{n_i^+ + n_i^-} \tag{3.4}$$

A maioria dos algoritmos de aprendizado supervisionado podem ser convertidos em ranqueadores ao utilizar a probabilidade do exemplo pertencer a uma classe. Classificadores frequentemente podem fornecer um valor numérico que representa o valor de confiança da classificação, chamado de *score*. A conversão de classificadores em *rankings* só é possível devido aos *scores* calculados pelos algoritmos. Para alguns algoritmos, a obtenção desse *score* é trivial. (Russell e Norvig, 2002)

Na próxima seção são abordados os conceitos de *bag of words* utilizados pelo SMART-SAINT.

3.7 *Bag Of Words*

Devido a natureza textual não estruturada, os documentos de texto necessitam de um pré-processamento para serem submetidos a algoritmos de aprendizado. A transformação dos documentos em uma representação mais adequada, como em uma tabela atributo-valor, é uma etapa de suma importância, visto que a representação desses documentos tem uma influência fundamental em quão bem um algoritmo de aprendizado poderá generalizar a partir dos exemplos (Sebastiani e Ricerche, 2002).

Na abordagem *bag-of-words* (saco de palavras), cada documento é representado como um vetor das palavras que ocorrem no documento ou, em representações mais sofisticadas, como um vetor de frases ou sentenças, sem levar em consideração a disposição das palavras dentro do documento.

Dada uma coleção de n_{ex} documentos $D = d_1, d_2, ..., d_{n_{\text{ex}}}$. A representação de documentos usando a abordagem *bag-of-words* pode utilizar o formato de uma tabela atributo-valor, como representada na Tabela 3.4. Cada documento d_i é um exemplo da tabela e cada termo (palavra) t_j é um elemento do conjunto de atributos.

Tabela 3.4: Exemplos no formato atributo valor

	t_1	t_2	$\dots$	$t_{n_{\text{atr}}}$
d_1	a_{11}	a_{12}	$\vdots$	$a_{1n_{\text{atr}}}$
d_2	a_{21}	a_{22}	$\vdots$	$a_{2n_{\text{atr}}}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$d_{n_{\text{ex}}}$	$a_{n_{\text{ex}}1}$	$a_{n_{\text{ex}}2}$	$\vdots$	$a_{n_{\text{ex}}n_{\text{atr}}}$

Mais especificamente, na Tabela 3.4 estão representados n_{ex} documentos (exemplos) compostos por n_{atr} termos (atributos). Cada documento d_i é um vetor $d_i = (a_{i1}, a_{i2}, ..., a_{in_{\text{atr}}})$, no qual o valor a_{ij} refere-se ao valor associado ao j-ésimo termo do documento i. O valor a_{ij}, do termo t_j no documento d_i, pode ser calculado utilizando diferentes medidas, tais como:

- *boolean*;
- *tf*;
- *tfidf*.

A medida *boolean* usa a representação binária para os termos. Quando o termo está presente no documento, o valor de a_{ij} é 1, caso contrário 0. Essa medida é muito simples e, geralmente, medidas estatísticas são empregadas levando em consideração a frequência que os termos são encontrados nos documentos (Salton e Buckley, 1988). A medida *tf (term frequency)* considera o valor de a_{ij} como a frequência que o termo aparece no documento. Neste trabalho optou-se pela medida *tf*. Essa medida é definida pela Equação 3.5, na qual $freq(t_j, d_i)$ é a frequência do termo t_j no documento d_i.

$$tf(t_j, d_i) = freq(t_j, d_i) \tag{3.5}$$

A medida *tfidf* também utiliza informações que indicam a frequência que o termo aparece na coleção de documentos. Porém, a medida *tfidf* usa um fator de ponderação para que os termos que aparecem na maioria dos documentos tenham uma "força" de representação menor.

Representar uma coleção de documentos utilizando a abordagem *bag-of-words* pode ser muito custoso, pois o vetor que representa cada documento deve conter todos os termos de toda a coleção de documentos. Facilmente um conjunto de somente 10 documentos de tamanho médio pode necessitar de mais de 2000 atributos (termos) para ser representado, ou seja, cada documento é representado por um vetor de alta dimensão. Mais ainda, considere uma coleção de 100 documentos, na qual em muitos desses documentos somente 3 termos aparecem. No entanto, cada documento deve ser representado por um vetor no qual encontram-se representados todos os termos que aparecem na coleção de documentos. Assim, quanto maior o número de documentos na coleção, o tamanho desses documentos bem como os termos tratados, provavelmente, menor será a porcentagem de valores não

nulos a_{ij} dos termos utilizados na representação da tabela atributo-valor (Tabela 3.4). Em outras palavras, essa tabela é uma tabela esparsa.

Vários métodos podem ser utilizados a fim de reduzir a quantidade de termos (atributos) necessários para representar uma coleção de documentos, visando uma melhor representação e melhor desempenho do processo de Mineração de Textos (MT). Um método para esse fim é a transformação de cada termo para o radical que o originou. Os algoritmos de *stemming* são um método amplamente utilizado e difundido para essa transformação. Uma outra forma para reduzir a dimensionalidade dos atributos é buscando os termos que são mais representativos na discriminação dos documentos usando os cortes de Luhn. Ainda é possível reconhecer e ignorar algumas palavras consideradas irrelevantes, conhecidas como *stopword*s. A seguir são descritas em detalhes cada uma dessas técnicas.

Stopwords

Várias palavras, em qualquer língua, são muito comuns e não são significativas para o aprendizado quando consideradas isoladamente. Entre essas palavras, denominadas *stopword*s, geralmente estão os pronomes, os artigos, as preposições, os advérbios, as conjunções e os verbos de ligação. Tais palavras formam o que se denomina uma *stoplist*, a qual contém as *stopwords* que devem ser desconsideradas ao processar o texto. Dessa forma, a remoção de *stopwords* minimiza consideravelmente a quantidade total de palavras usadas como atributos para descrever os documentos, mantendo apenas palavras consideradas mais relevantes para o aprendizado.

Stemming

Basicamente, o *stemming* consiste em uma normalização linguística, na qual as formas variantes de um termo são reduzidas a uma forma comum denominada *stem*. A consequência da aplicação do *stemming* consiste na remoção de prefixos ou sufixos de um termo, ou mesmo na transformação de um verbo para sua forma no infinitivo. Por exemplo, as palavras **trabalham, trabalhando, trabalhar, trabalho** e **trabalhos** podem ser transformados para um mesmo *stem* **trabalh**. Desse modo, o *stemming* podem ser utilizados para reduzir a dimensão da tabela atributo-valor.

Percebe-se que algoritmos de *stemming* são fortemente dependentes do idioma no qual os documentos estão escritos. Um dos algoritmos de *stemming* mais conhecidos é o algoritmo do Porter, que remove sufixos de termos em inglês (Porter, 1980). O algoritmo tem sido amplamente usado, referenciado e adaptado nos últimos 20 anos. Diversas implementações do algoritmo estão disponibilizadas na internet, entre elas a página oficial escrita e mantida pelo autor para distribuição do seu algoritmo (`http://www.tartarus.org/~martin/PorterStemmer`).

É pouco provável que o *stemming* retorne o mesmo *stem* para todas as palavras que tenham a mesma origem ou radical morfológico, devido a própria natureza intrínseca dos

sistemas relacionados à Recuperação de Informações (RI) e Processamento de Linguagem Natural (PLN). Já foi observado que a medida que o algoritmo se torna mais específico na tentativa de minimizar a quantidade de *stems* diferentes para palavras com um mesmo radical, a eficiência do algoritmo degrada (Porter, 1980).

Cortes por Frequência

Um outro modo para reduzir a dimensionalidade dos atributos é buscar os termos que são mais representativos na discriminação dos documentos. A Lei de Zipf descreve uma maneira de encontrar termos considerados pouco representativos em uma determinada coleção de documentos. A lei, formulada por George Kingsley Zipf, professor de linguística de Harvard (1902-1950), declara que a frequência de ocorrência de algum evento está relacionada a uma função de ordenação (Zipf, 1949).

Existem diversas maneiras de enunciar a Lei de Zipf para uma coleção de documentos. A mais simples é procedimental: pegar todos os termos na coleção e contar o número de vezes que cada termo aparece. Se o histograma resultante for ordenado de forma decrescente, ou seja, o termo que ocorre mais frequentemente aparece primeiro, então, a forma da curva é a "curva de Zipf" para aquela coleção de documentos. Se a curva de Zipf for plotada em uma escala logarítmica, ela aparece como uma reta com inclinação -1. A Lei de Zipf em documentos de linguagem natural pode ser aplicada não apenas aos termos mas, também, a frases e sentenças da linguagem.

Enquanto Zipf verificou sua lei utilizando jornais escritos em inglês, Luhn usou a lei como uma hipótese nula para especificar dois pontos de corte, os quais denominou de superior e inferior, para excluir termos não relevantes (Luhn, 1958). Os termos que excedem o corte superior são os mais frequentes e são considerados comuns por aparecer em qualquer tipo de documento, como as preposições, conjunções e artigos. Já os termos abaixo do corte inferior são considerados raros e, portanto, não contribuem significativamente na discriminação dos documentos. A Figura 3.9 mostra a curva da Lei de Zipf (I) e os cortes de Luhn aplicados na Lei de Zipf (II), no qual o eixo cartesiano f representa a frequência das palavras e o eixo cartesiano r as palavras correspondentes ordenadas segundo essa frequência.

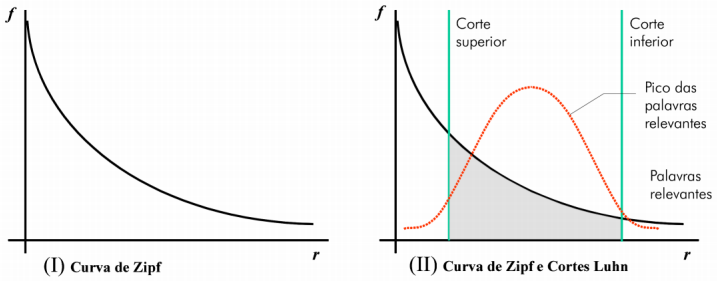


Figura 3.9: A curva de Zipf e os cortes de Luhn (Matsubara e Monard, 2003).

Assim, Luhn propôs uma técnica para encontrar termos relevantes, assumindo que os

termos mais significativos para discriminar o conteúdo do documento estão em um pico imaginário posicionado no meio dos dois pontos de corte. Porém, uma certa arbitrariedade está envolvida na determinação dos pontos de corte, bem como na curva imaginária, os quais são estabelecidos por tentativa e erro.

3.8 Considerações Finais

Neste capítulo foram apresentados métodos diferentes usados para atingir o objetivo deste trabalho. Foram mostrados os trabalhos mais relevantes de cada área bem como o atual estado da arte em Aprendizado Semissupervisionado, Aprendizado Ativo, Seleção de Atributos e Filtros de Rede.

Também foram explicados alguns métodos usados no trabalho e na avaliação experimental, como os conceitos de Análise ROC, *Rankings* e o funcionamento das *bag of words*.

No próximo capítulo será tratado sobre o sistema desenvolvido neste trabalho, seu funcionamento e como foram usados os métodos e conceitos apresentados no capítulo atual.

Capítulo 4

SmartSaint

4.1 Introdução

Crianças e adolescentes são expostos precocemente a informações indevidas para sua faixa etária. Algumas famílias simplesmente proíbem totalmente o acesso e, em muitas outras, o acesso é totalmente liberado. Há ainda instituições que gostariam que seus alunos ou colaboradores mantivessem o foco no que é realmente importante. Por isso é necessário que haja um sistema de moderação de conteúdo que possa ser personalizado para cada situação.

O objetivo deste trabalho é projetar e desenvolver um *sistema moderador de conteúdo* que atue como um filtro de internet inteligente, fazendo a indução de um classificador com o uso de aprendizado semissupervisionado ativo, denominado SmartSaint. O SmartSaint faz a moderação de acesso a sítios baseada no conteúdo textual. O filtro pode ser usado tanto para controle dos pais, como instalado em servidores de instituições para restringir o acesso a internet. No entanto, como é inviável analisar e processar toda a internet, especialmente porque o conteúdo é alterado todos os dias, o SmartSaint funciona sob demanda: conforme os usuários navegam pela web, o conteúdo acessado é processado para verificar se o acesso ao conteúdo é autorizado. Dessa maneira, o sistema se mantém atualizado e preciso.

Na seção seguinte será explicado como foi feita a implementação do pré-processador utilizado no SmartSaint.

4.2 BagMaker

A representação de documentos textuais em um formato estruturado, para o processo de Mineração de Textos (MT), tem uma influência fundamental em quão bem um algoritmo de aprendizado poderá generalizar. A abordagem *bag-of-words* é uma das representações estruturadas mais simples, mais utilizada e que tem obtido um bom desempenho no processo de Mineração de Textos. No entanto, essa abordagem é caracterizada pela alta dimensionalidade e por valores esparsos na representação dos textos, visto que cada pa-

lavra é um possível atributo nessa representação. São necessárias, portanto, ferramentas computacionais que possam realizar, de forma automática, a transformação dos documentos em uma representação estruturada e que, ao mesmo tempo, auxiliem na redução da dimensionalidade. Nesta seção é descrita a ferramenta BagMaker que tem essas características, bem como diversas outras funcionalidades que a distingue de outras ferramentas existentes. Quando for falado de documento (termo usado na área de MT) em relação ao BagMaker entenda sítio da internet.

O BagMaker é uma ferramenta computacional desenvolvida na linguagem de programação Python (van Rossum e de Boer, 1991), que tem como objetivo realizar pré-processamento de textos HTML e transformar esses textos em um formato estruturado, utilizado pela maioria dos algoritmos de Aprendizado de Máquina. Uma de suas principais funcionalidades é transformar sítios da internet, escritos em português ou inglês, em *stems*. A ferramenta usa o algoritmo do Porter para efetuar o *stemming* do texto. Além disso, a ferramenta apresenta facilidades para reduzir a dimensionalidade do conjunto de atributos, usando a lei de Zipf e os cortes de Luhn.

Atualmente a ferramenta somente faz a construção de *stems* usando 1-*gram*, porém pode ser adaptado para 2 ou 3-*gram*. O termo 1-*gram* se refere a um *stem* simples, enquanto 2 e 3-*gram* referem-se a 2 ou 3 *stems*, cujas palavras ocorrem sequencialmente no documento. A utilização de mais de um *gram* permite que palavras que aparecem sequencialmente no documento, como "inteligência artificial", "aprendizado de máquina" e "mineração de texto", que são mais representativas conceitualmente quando utilizadas juntas, possam ser utilizadas no BagMaker.

No BagMaker, a tabela atributo-valor é a principal estrutura de armazenamento de dados. Essa estrutura foi projetada para ser manipulada por outras classes. Normalmente, uma matriz é utilizada para representar uma estrutura de tabela atributo-valor. Para Mineração de Textos, muitas vezes essa não é a melhor estrutura de armazenamento. Isso ocorre devido a grande dimensão dos vetores na representação *bag-of-words*. Deve ser observado que, a dimensão dos vetores que representam cada elemento da coleção de documentos, é a mesma para qualquer documento na coleção, é dada pelo número de termos em toda a coleção de documentos. Desse modo, é gerada uma matriz de alta dimensão com poucos valores não nulos.

Suponha uma coleção de 100 documentos com 1.000 termos cada, no qual, todos os termos são disjuntos, i.e., não há termos em comum. Com a representação de tabela atributo-valor como uma matriz, cada documento é representado com um vetor de tamanho 100.000, mas tendo apenas 10% de seus valores não nulos. Essa representação, na maioria das vezes, quando não tratada, desperdiça muita memória. No BagMaker, ao invés dessa representação, pelo menos durante o armazenamento, é utilizado um dicionário para a coleção de documentos. Desse modo, no exemplo apresentado, é utilizado uma estrutura que equivaleria aproximadamente a um vetor de tamanho 1.000 (apenas os índices das palavras no dicionário e sua quantidade no documento), ao invés de 100.000.

O texto contido no sítio HTML, ao ser pré-processado pela ferramenta BagMaker, é transformado em uma tabela atributo-valor. O BagMaker transforma as palavras do texto em *stems*, removendo qualquer *stopword* a partir de uma *stoplist* interna. Os *stems* irão constituir os possíveis atributos utilizados na classificação do sítio.

Para calcular o valor desses atributos foi utilizada a medida *tf* (frequência de ocorrência do *stem*). Em seguida, foram aplicados os cortes de Luhn (Luhn, 1958), nos quais é possível especificar a frequência mínima e máxima dos *stems* da coleção. Neste trabalho, foi utilizado o valor 2 para realizar o corte mínimo em ambas descrições, isto é, todos os *stems* que aparecem apenas 1 vez, em toda a coleção, foram retirados da tabela atributo-valor que representa os documentos. O número máximo de aparições de um *stem* não foi limitado.

Após montada a tabela atributo-valor, o BagMaker irá filtrar os atributos, baseado na seleção de atributos feita previamente pelo classificador.

Na próxima seção é explicado o funcionamento do SmartSaint e como o BagMaker se integra com ele.

4.3 Funcionamento

A ideia por trás do SmartSaint é simples. Caso os usuários naveguem apenas por conteúdos permitidos, a presença dele nem deve ser notada. O usuário só tomará conhecimento da existência do SmartSaint caso tente acessar algum conteúdo proibido.

Uma visão geral do funcionamento do SmartSaint é ilustrado pela Figura 4.1 e obedece os seguintes passos:

1. O usuário tenta acessar um conteúdo de internet a partir de seu computador;
2. O SmartSaint, integrado ao *proxy*, verifica se o conteúdo já foi analisado e está em *cache*. Caso não tenha sido, faz o download do sítio para classificação;
3. O SmartSaint classifica o sítio e, se o conteúdo for classificado como autorizado, permite o acesso enviando o conteúdo ao cliente. Caso contrário, uma página de acesso negado será enviada como resposta.

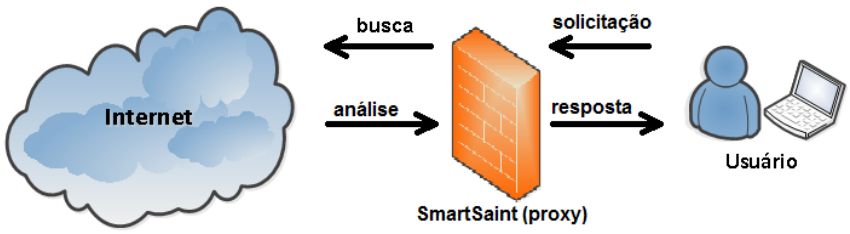


Figura 4.1: Visão Geral do Funcionamento

À direita da Figura 4.1 está o usuário que faz a solicitação de um sítio. Caso o *proxy* não tenha esse sítio em *cache*, ele faz a solicitação à internet, representada pela nuvem

à esquerda. A resposta vinda da internet é então recebida pelo *proxy*, que a submete a uma análise pelo SmartSaint. O SmartSaint filtra o conteúdo, apresentando como resposta ao usuário uma página padrão de acesso negado, quando o conteúdo não for autorizado. Analogamente, quando o conteúdo for autorizado é apresentada ao usuário a página do sítio sem modificações. A seguir será explicado como funciona esse filtro e os processos internos do SmartSaint.

Processos Internos

Sempre que o usuário solicitar um sítio, o *proxy* perguntará ao SmartSaint a classificação do mesmo. Para isso, o *proxy* submete ao SmartSaint o endereço (URL) do sítio e o seu conteúdo.

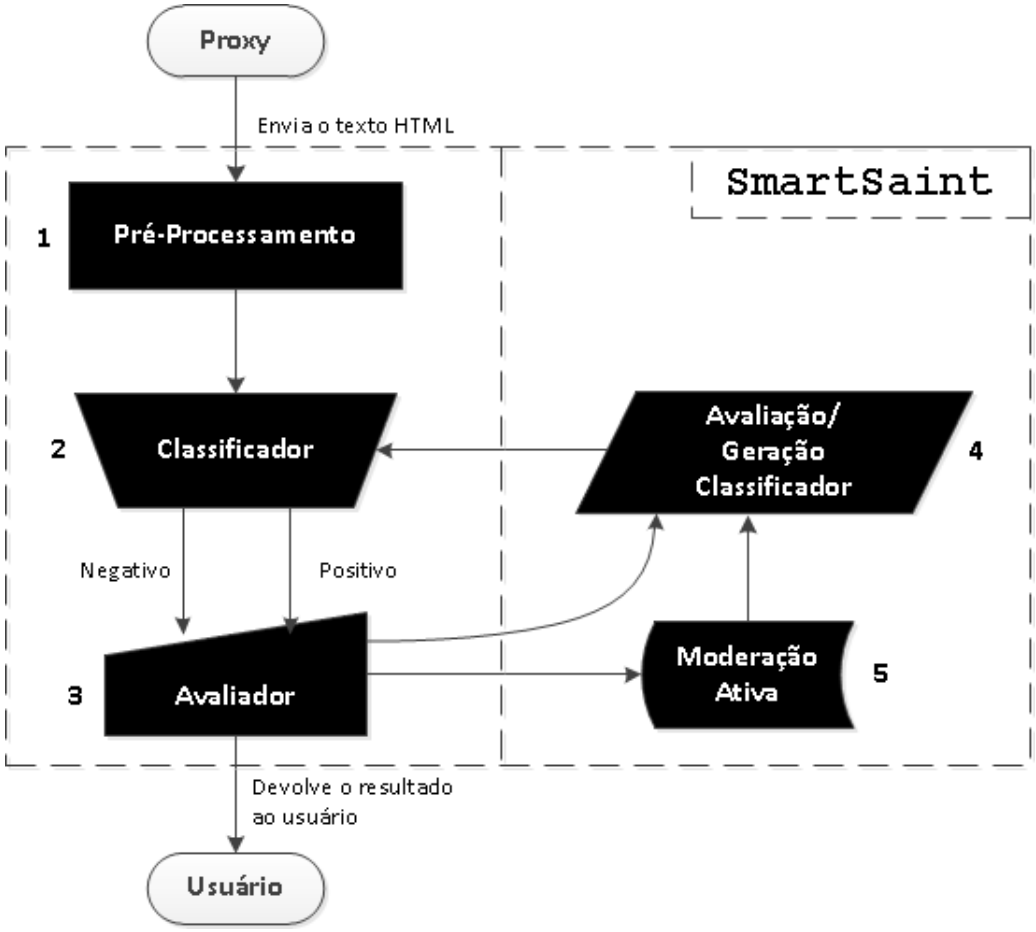


Figura 4.2: Visão Detalhada do Funcionamento do SmartSaint

A Figura 4.2 detalha o funcionamento interno do sistema. No topo o *proxy* submete o sítio para análise pelo SmartSaint. Quando o SmartSaint recebe o texto HTML do *proxy*, o primeiro passo é enviá-lo para Pré-Processamento (1) no BagMaker. No Pré-Processamento o HTML é convertido para o formato *bag-of-words*, conforme explicado na Seção 4.2. Após o Pré-Processamento, a *bag-of-words* é enviada ao Classificador (2). O Classificador faz a análise do conteúdo, classificando-o como negativo ou positivo e, em seguida, envia o resultado ao Avaliador (3). O Avaliador repassa o conteúdo ao

usuário caso seja classificado como positivo. Caso o conteúdo tenha sido classificado como negativo, o Avaliador envia ao usuário uma página de informação de acesso negado. Ao mesmo tempo, o Avaliador decide se aquele conteúdo se enquadra nos critérios para ser enviado para Moderação Ativa (pontos de contenção). Na Moderação Ativa (5), a classificação deve ser revista manualmente por um moderador e posteriormente encaminhada para o Gerador (4). Caso o conteúdo não vá para Moderação Ativa, o Avaliador o envia diretamente para o Gerador, que avalia a qualidade do Classificador, baseado nas últimas classificações e conteúdos recebidos, e gera um novo classificador caso seja necessário.

Como ilustrado na Figura 4.2, o SmartSaint pode ser dividido em dois blocos. O bloco da esquerda é o fluxo do dia-a-dia, no qual o usuário faz o acesso aos sítios permitidos. Esse bloco deve ser o mais rápido possível, pois o usuário não pode ficar esperando para receber o resultado da requisição. O bloco da direita é a geração do classificador, este bloco é usado com menor frequência e controlado pelo administrador do sistema. Esse bloco não precisa ser rápido e é onde é gasto mais tempo para fazer otimizações no classificador, de maneira que ele seja rápido. Algumas otimizações no classificador incluem a seleção de atributos e a moderação ativa. A seguir é visto com mais detalhe o funcionamento de cada um dos blocos.

Classificador

Algoritmo 2 `Classifica`

Entrada: host, texto
Saída: autorizado
  Buscar classificação prévia de **host**
  Se Não Classificado **ou** Classificação Antiga **então**
    bag-of-words ← BagMaker (**texto**)
    classe1 ← `MultinomialNB`(*bag-of-words*)
    classe2 ← `LogReg`(*bag-of-words*)
    Se classe1 = classe2 então
      classe de **host** ← **classe1**
    Senão {algoritmos discordam}
      Marca **host** para Moderação Ativa
      Se Confiança `MultinomialNB` > Confiança `LogReg` **então**
        classe de **host** ← **classe1**
      Senão
        classe de **host** ← **classe2**
      Fim Se
    Fim Se
  Fim Se
  autorizado ← classe de **host** permitida ?
  Retorna autorizado

A maneira como o SmartSaint efetua a classificação é mostrado no Algoritmo 2. O algoritmo recebe como entrada o endereço do sítio (**host**) e o conteúdo do sítio (**texto**), ambos fornecidos pelo *proxy*. Como saída, o algoritmo retorna um booleano (**autorizado**)

indicando se o sítio foi autorizado ou não. Dependendo do valor do booleano retornado, o *proxy* sabe se deve enviar o conteúdo do sítio ao usuário, ou se deve enviar o conteúdo de página não autorizada.

O algoritmo `Classifica` faz uso da classe BagMaker, que retornará os atributos mais relevantes do texto na forma de *bag-of-words*. A *bag-of-words* será passada para dois classificadores do `SciKit-learn` : `MultinomialNB` e `LogReg`. Caso os dois algoritmos concordem na classificação, será retornado o resultado para que o *proxy* envie a resposta ao usuário. Caso os algoritmos discordem, o usuário receberá a resposta do algoritmo com maior confiança e o **host** será marcado para Moderação Ativa.

Moderação Ativa e Gerador

Algoritmo 3 CoLabeling

Entrada: L, U, k
  Para $it = 1$ **até** k **faça**
    seleção de atributos sobre L
    obter o classificador h_1 usando o algoritmo 1
    obter o classificador h_2 usando o algoritmo 2
    $R_1 =$ todos os exemplos de U rotulados por h_1
    $R_2 =$ todos os exemplos de U rotulados por h_2
    $PC = \{(x_i) | h_1(x_i) \neq h_2(x_i) \wedge (x_i, h_1(x_i)) \in R_1 \wedge (x_i, h_2(x_i)) \in R_2\}$
    Se $PC = \emptyset$ **então**
      Retorna h_1, h_2, L
    Fim Se
    $R_o =$ exemplos rótulados pelo oráculo a partir de PC
    $L = L \cup R_o$
    $U = U - R_o$
  Fim Para
  Retorna h_1, h_2, L

Nesse segundo bloco do SmartSaint é feita a geração do classificador baseada na moderação ativa, conforme mostrado no Algoritmo 3. O CoLabeling usa dois classificadores h_1 e h_2, gerados a partir de dois algoritmos de classificação diferentes, usando o mesmo conjunto de dados. Para o SmartSaint foram escolhidos os classificadores `MultinomialNB` e `LogReg`, conforme será explicado na Seção 5.2. O algoritmo recebe como entrada os conjuntos de exemplos rotulados L e de exemplos não rotulados U, e ainda um inteiro k, que define o número máximo de iterações do algoritmo.

Antes de induzir os classificadores é feita uma seleção de atributos sobre L, visando melhorar a precisão e eficiência dos classificadores. Com os classificadores induzidos é feita a classificação de todos exemplos de U por h_1 e h_2. Após todos os exemplos terem sido rotulados, o conjunto PC é construído de maneira a conter os exemplos que foram rotulados com classes diferentes por cada classificador. Se $PC = \emptyset$ o processo termina, caso contrário, no próximo passo os exemplos pertencentes ao conjunto PC são submetidos ao oráculo, que pode rotular um número arbitrário de exemplos. Os exemplos x_i, onde

i é o índice dos exemplos com $i = 1, .., n_{\text{ex}}$, rotulados pelo oráculo, são retirados de U e inseridos no conjunto L de exemplos rotulados. O processo é repetido até k vezes caso não se verifique $PC = \emptyset$.

Na próxima seção é explicado a documentação do SmartSaint.

4.4 Documentação

O projeto foi desenvolvido em `Python` usando como base o *framework* `SciKit-learn` (Pedregosa et al., 2011) e diversas de suas ferramentas. Para melhor organização, o código foi dividido em diversas classes, que serão melhor explicadas a seguir: Main, piMiProxy, SmartSaint, BagMaker e CoLabeling.

A classe Main centraliza a execução do código, habilitando o *proxy* (piMiProxy) e o instruindo a passar todas as requisições ao SmartSaint. O piMiProxy foi baseado na ferramenta disponibilizada por Allfro (2013), que permite que todo conteúdo seja interceptado, analisado e modificado, se necessário.

O ciclo de vida do SmartSaint tem inicio quando é recebida a requisição do *proxy* e, em seguida, verifica em sua base se o sítio já foi classificado. Caso o sítio ainda não tenha sido classificado, o SmartSaint pega o conteúdo HTML do sítio e o fornece ao BagMaker (Pré-Processamento), que transforma o sítio em palavras chaves (*keywords*) no formato de *bag-of-words*. A *bag-of-words* é enviada para o Classificador (CoLabeling) fazer a classificação do conteúdo do sítio, obtendo o rótulo autorizado ou proibido. Se o conteúdo for rotulado como autorizado, o acesso é liberado e o conteúdo é repassado ao usuário. Caso o conteúdo seja rotulado como proibido, será exibido ao usuário uma página padrão de acesso negado.

O SmartSaint armazena várias informações a respeito de cada sítio acessado (e classificado). Entre essas informações estão as datas de quando ele foi acessado e a data em que foi classificado. Essas informações são importantes para que o sítio possa ser reavaliado de tempos em tempos, conforme um período pré-definido. Assim, caso o conteúdo do sítio seja modificado, passando a conter algum conteúdo não autorizado, o sítio receberá novo rótulo de proibido. O mesmo ocorre no sentido inverso, caso um sítio que era proibido tenha seu conteúdo modificado, de maneira a ter somente conteúdos autorizados, então ele receberá o novo rótulo autorizado.

O SmartSaint possui um *ranking* de todos os sítios classificados. Ele utiliza-se de um limiar pré-definido, e que pode ser ajustado pelo moderador, para permitir ou bloquear o acesso aos sítios, analisando se o *ranking* do sítio está abaixo ou acima do limiar. Conforme novas classificações sejam realizadas, a posição de um sítio pode subir ou descer no *ranking* e o sítio ser rotulado como autorizado ou proibido, conforme as alterações de conteúdo do mesmo e dos ajustes no limiar do classificador. Um exemplo de *ranking* de sítios, ranqueados como adultos por Alexa (2012), pode ser visto na Figura 4.3. Sempre que há uma atualização no classificador ou uma alteração de limiar, assim como alteração

do conteúdo do sítio, os sítios poderão receber novos rótulos (autorizado / proibido). A cada nova classificação de um sítio, ele pode ter seu *ranking* alterado. Na Figura 4.3 são exibidos apenas os sítios cujo *ranking* os classifica como proibido. A coluna "Posição" mostra a posição atual do sítio no *ranking*. A coluna "*Ranking*" mostra a classificação atual do sítio, enquanto as colunas "-7", "-30", "-90" exibem qual era a classificação do sítio 7, 30 e 90 dias atrás, respectivamente. Essas informações são abstraídas para uma interface mais intuitiva para o usuário, conforme ilustrado na seção 4.5.

Site	Posição	Ranking	-7	-30	-90
livejasmin.com	1	3.02	2.686	2.395	1.962
youporn.com	2	1.04	0.929	0.916	0.918
xnxx.com	3	0.74	0.658	0.661	0.657
adultfriendfinder.com	4	0.58	0.524	0.536	0.529
streamate.com	5	0.123	0.11	0.136	0.1505
imlive.com	6	0.114	0.117	0.148	0.0837
freeones.com	7	0.126	0.109	0.1097	0.1122
literotica.com	8	0.06	0.052	0.0543	0.0557
voyeurweb.com	9	0.033	0.034	0.0345	0.0363
playboy.com	10	0.0837	0.045	0.046	0.0416

Figura 4.3: Ranqueamento de sítios Adultos. Alexa (2012) com adaptações.

Como a internet está em constante transformação, é necessário um filtro que seja dinâmico. Essa é a principal vantagem do SmartSaint sobre os filtros existentes. Vários sítios são criados e outros saem do ar todos os dias. Um filtro estático, no qual o moderador seleciona sítio por sítio para bloquear, não parece ser uma alternativa viável. Assim, no SmartSaint, se o moderador decide bloquear sítios de relacionamento, ele poderá dar exemplos como *MySpace* e *Orkut*, e o SmartSaint se encarregará de bloquear também sítios como *Facebook* e o *Sonico*.

A implementação do SmartSaint utiliza os conceitos de aprendizado semissupervisionado ativo e *rankings* descritos no Capítulo 3. O SmartSaint baseia-se ainda em ferramentas como o `SciKit-learn` e o piMiProxy. O `SciKit-learn` integra algoritmos de Aprendizado de Máquina no mundo científico do `Python` , construído sobre outras ferramentas `Python` como: `numpy, scipy`, e `matplotlib`. Como um módulo de Aprendizado de Máquina, o `SciKit-learn` provê ferramentas versáteis para mineração de dados e análise em qualquer campo da ciência ou engenharia. Ele tenta ser simples e eficiente, acessível a todos, e reutilizável em vários contextos. O piMiProxy é um *proxy* pequeno e leve capaz de inspecionar conteúdo sobre HTTP e HTTPS (ou SSL). Ele também é extensível provendo dois meios para isso: sobrecarga de métodos e uma interface conectável. Neste projeto foi utilizado a interface conectável, na qual o Main conecta o piMiProxy e o SmartSaint. O piMiProxy é ideal para situações nas quais você deseja analisar o conteúdo de entrada e saída HTTP e modificá-lo quando necessário.

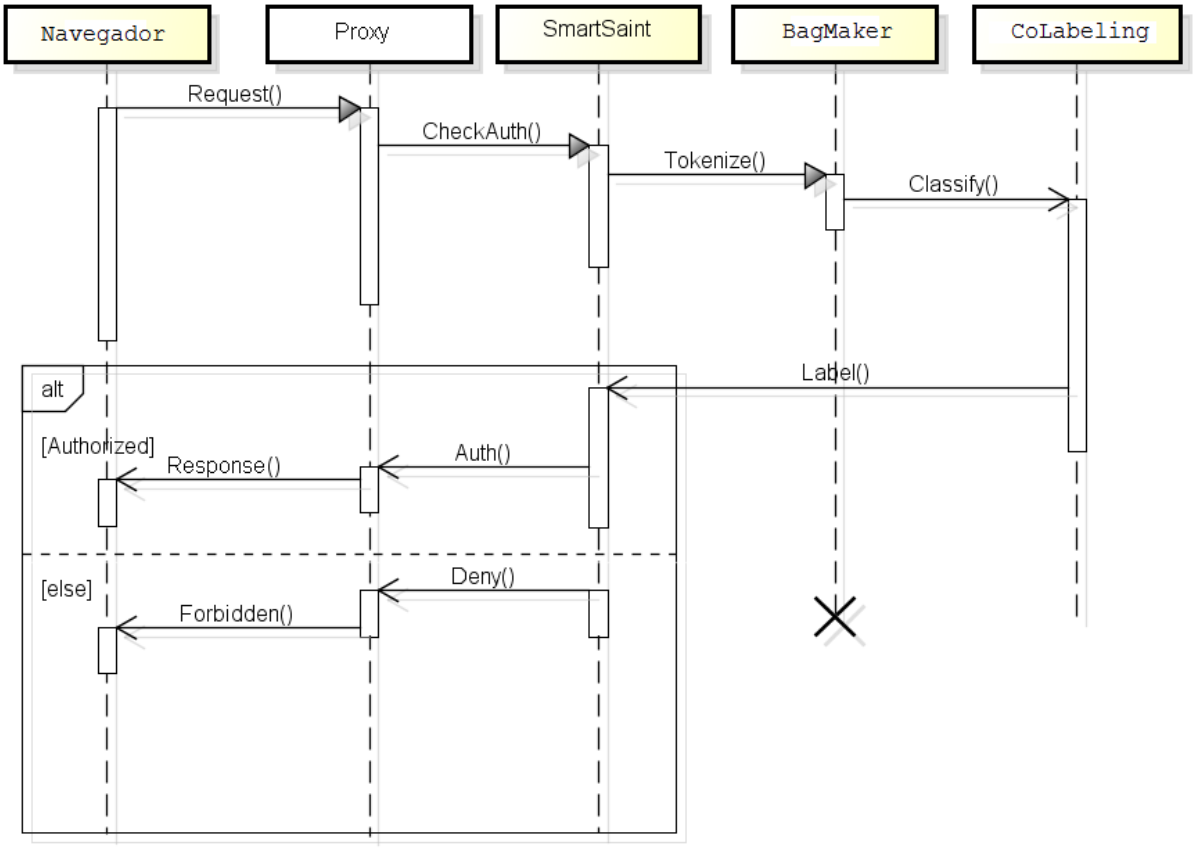


Figura 4.4: Diagrama de Sequência

Na Figura 4.4 é apresentado o diagrama de sequência que mostra o funcionamento e troca de mensagens entre os componentes do sistema. Inicialmente, o navegador de internet faz uma requisição (*Request*) ao *Proxy*. Caso o *Proxy* ainda não tenha o conteúdo do sítio em *cache*, ele irá fazer a copia do conteúdo. Em seguida, o *Proxy* verifica com o SmartSaint se o acesso àquele determinado sítio está autorizado (*CheckAuth*). Caso o SmartSaint já tenha classificado a página em questão recentemente, se ela for autorizada (*Auth*) o *proxy* entrega o conteúdo do sítio (*Response*). Caso não seja autorizada (*Deny*) será entregue ao usuário o conteúdo de uma página de acesso negado (*Forbidden*).

Caso seja a primeira vez que o SmartSaint recebe uma requisição daquele sítio, ou caso ele necessite uma nova classificação, o SmartSaint pegará o conteúdo do sítio copiado recentemente pelo *proxy* para análise. Esse conteúdo é passado inicialmente ao BagMaker (*Tokenize*), que irá convertê-lo para o formato *bag-of-words*, e posteriormente enviá-lo para o CoLabeling (*Classify*). O CoLabeling procederá com a análise e classificação do sítio e devolverá o resultado ao SmartSaint (*Label*). A partir desse instante a lógica segue conforme explicada anteriormente.

4.5 Interfaces

Para demonstrar como seria a utilização do sistema por parte do moderador, nesta seção são ilustrados alguns protótipos das interfaces do SmartSaint.

Site	Posição	Ranking	-7	-30	-90
battle.net	1	0.426	0.407	0.532	0.3306
ign.com	2	0.299	0.301	0.32	0.314
pch.com	3	0.161	0.187	0.1895	0.1903
williamhill.com	4	0.26	0.278	0.262	0.2068
gamespot.com	5	0.165	0.195	0.2	0.1839
miniclip.com	6	0.189	0.209	0.202	0.2027
888.com	7	0.255	0.259	0.265	0.3668
pogo.com	8	0.154	0.147	0.1446	0.1499
gamefaqs.com	9	0.119	0.13	0.1217	0.1223
betfair.com	10	0.11	0.121	0.1072	0.1129

Figura 4.5: Ranqueamento de sítios de Jogos. Alexa (2012) com adaptações.

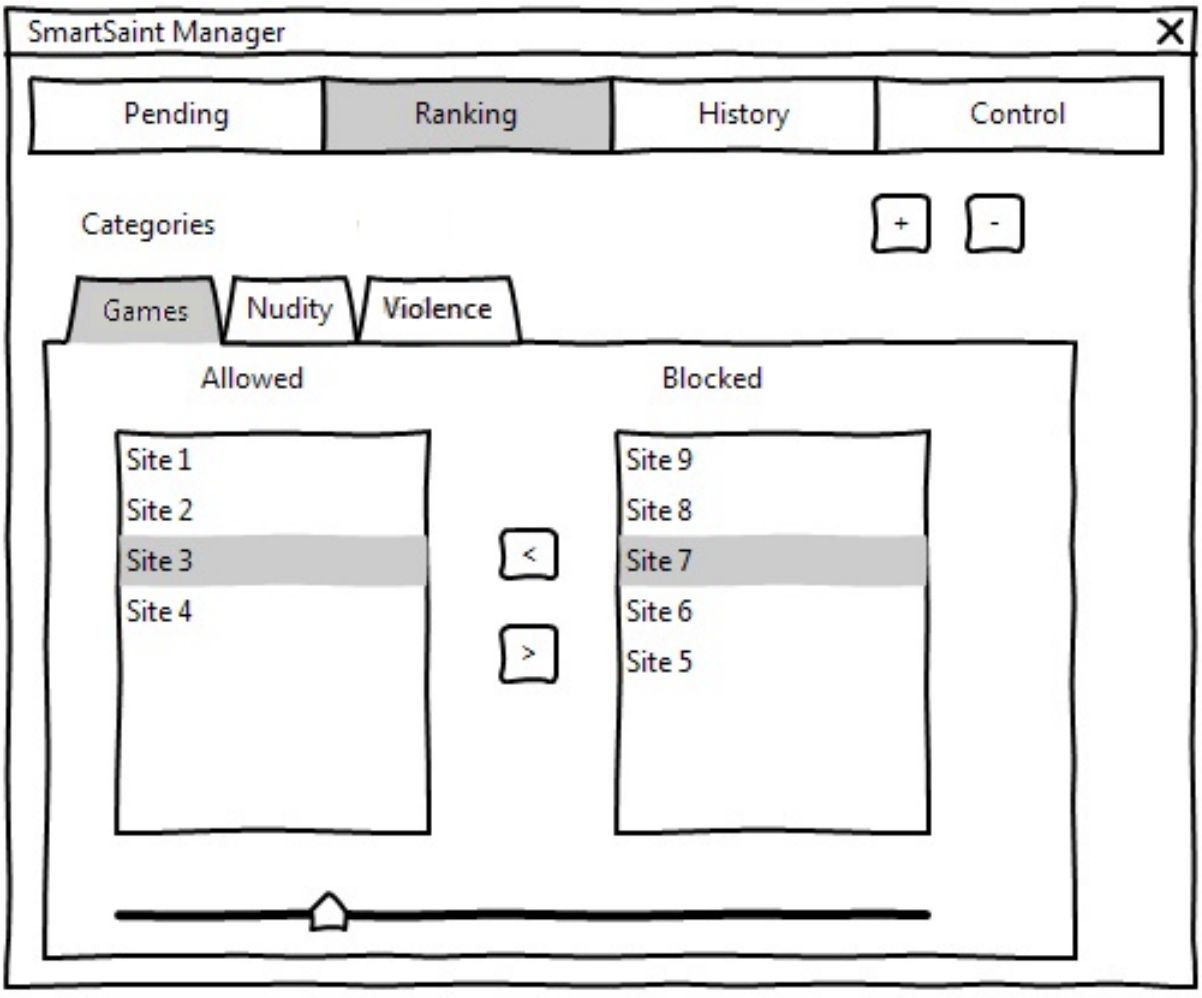


Figura 4.6: Interface de Ranqueamento

Na Figura 4.6 existem quatro seções *Pending*, *Ranking*, *History*, *Control*, sendo que a seção selecionada é a *Ranking*. A tela da seção de Ranqueamento (*Ranking*) exibe as categorias em que os sítios são classificados. Nesse caso *Games*, *Nudity* e *Violence*. O SmartSaint contém categorias pré-cadastradas que podem ser ativadas ou não. O

moderador pode ainda cadastrar novas categorias. Para cadastrar uma categoria, ele deve clicar no botão "+" e informar um nome para a categoria. Após deve adicionar alguns exemplos de sítios que devem ser proibidos por esta classificação e alguns exemplos de sítios que devem ser autorizados por esta classificação. Para apagar uma categoria é necessário seleciona-la e clicar no botão "-", que irá solicitar uma confirmação. Após a confirmação o classificador da categoria e todos os seus dados serão apagados. As categorias pré-cadastradas não podem ser apagadas, apenas serão desativadas apagando os dados do usuário, mas são mantidos os classificadores mesmo que não sejam usados. Cada categoria possui o seu classificador, assim, cada categoria contém um *Ranking* diferente. Os sítios possuem um *score* diferente para cada categoria. O *score* associado ao sítio o classifica como permitido (rótulo autorizado) ou não (rótulo proibido), de acordo com o ranqueamento e o limiar da categoria. O sítio só poderá ser visualizado pelo usuário se o mesmo for rotulado como autorizado em todas as categorias ativas.

Site	Posição	Ranking	-7	-30	-90
google.com	1	46.18	48.69	49.02	49.66
facebook.com	2	45.17	45.29	45.21	44.719
youtube.com	3	34.07	32.82	32.91	32.754
yahoo.com	4	20.77	21.33	21.58	21.885
baidu.com	5	11.67	11.6	11.49	11.306
wikipedia.org	6	13.09	13.93	13.954	13.887
live.com	7	9.65	10.22	10.225	10.343
twitter.com	8	8.64	8.81	9.273	9.231
qq.com	9	7.69	7.66	7.582	7.455
amazon.com	10	5.69	5.94	6.022	6.129

Figura 4.7: Ranqueamento de sítios Autorizados. Alexa (2012) com adaptações.

Para cada categoria selecionada são exibidas duas listagens de sítios. Conforme observa-se na Figura 4.6, a categoria *Games* está selecionada. Na listagem da esquerda estão os sítios rotulados como "autorizado", e na listagem da direita, estão os sítios rotulados como "proibido" para a categoria selecionada.

Nas Figuras 4.3 e 4.5 pode-se observar o que seria a listagem de sítios proibidos para as categorias *Nudity* e *Games*, respectivamente. A Figura 4.7 ilustra o que seria a listagem de sítios autorizados para ambas categorias.

O moderador tem a opção de alterar o rótulo de um sítio, para isso ele pode arrastar um sítio de uma lista para outra, ou então selecionar o sítio e pressionar o botão "<" para autorizar o acesso ao sítio, e o botão ">" para bloquear o acesso ao sítio. A alteração de rótulos fará com que o SmartSaint gere um novo classificador. Abaixo da tela de *Ranking* há um *slider*, onde o moderador poderá aumentar ou diminuir o limiar do ranqueamento. Alterando o limiar do ranqueamento, a moderação da categoria se tornará mais branda ou mais severa. A alteração do limiar fará com que os sítios próximo a borda do limiar

tenham suas categorias alteradas. A alteração de limiar não gera um novo classificador.

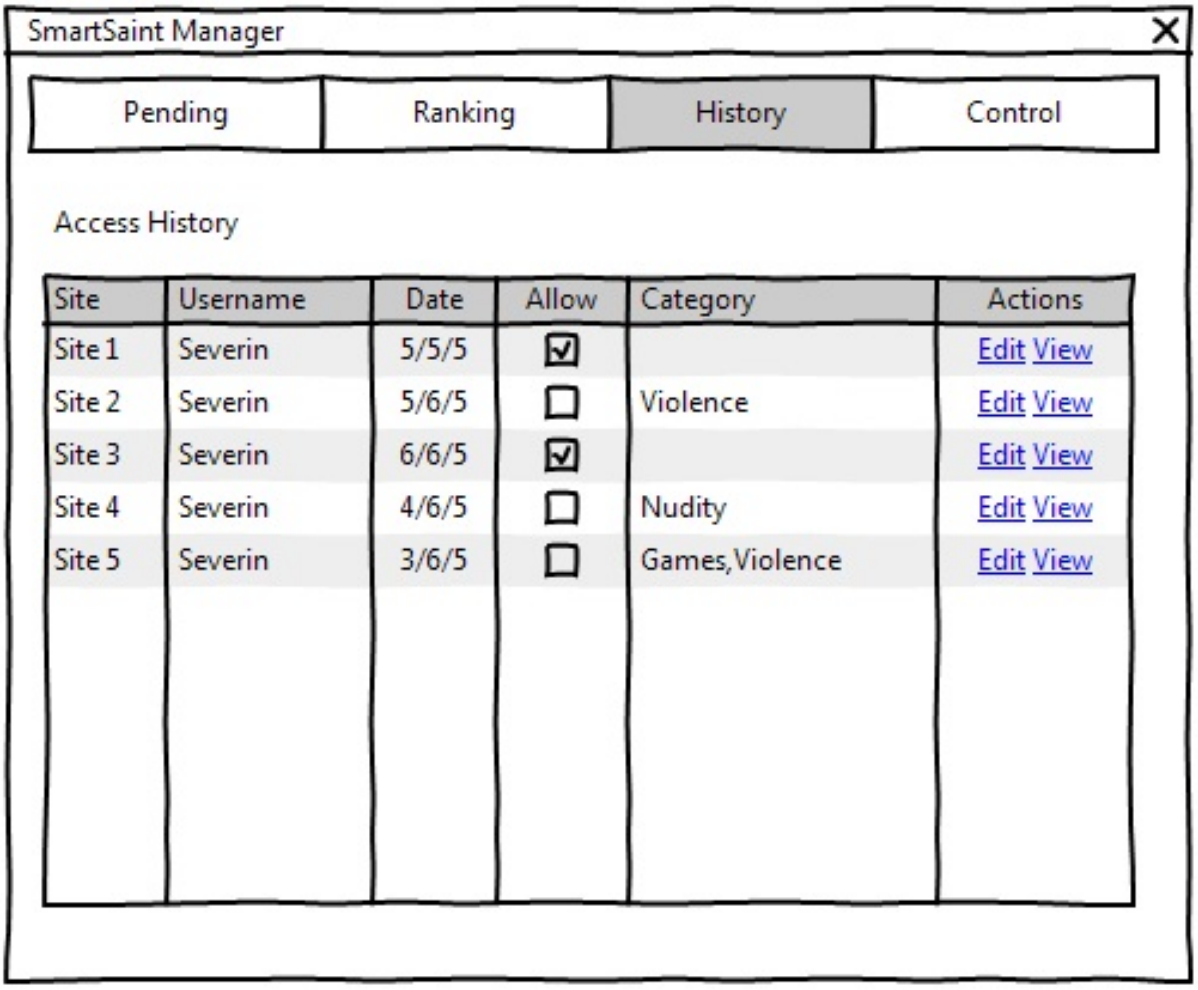


Figura 4.8: Interface de Histórico

Na Figura 4.8 é apresentada a prototipagem da tela da seção *History*, onde tem-se o histórico de todos os sítios acessados pelos usuários. Essa tela é formada basicamente por uma tabela, onde são listadas diversas informações sobre cada acesso feito pelos usuários. Os dados exibidos na tabela, da esquerda para direita, são o nome do sítio requisitado, o usuário que fez a requisição, a data da requisição, se o acesso foi autorizado ou não, as categorias pela qual o sítio foi proibido e, por último, as ações disponíveis para cada requisição. Quando o acesso é autorizado, a coluna *Allow* aparecerá com uma caixa marcada, enquanto se o acesso for negado será exibida uma caixa desmarcada. A coluna de categorias (*Category*) exibirá uma ou mais categorias pelas quais o sítio foi proibido ou ficará em branco, caso o sítio tenha sido autorizado.

O moderador pode executar duas ações sobre cada requisição constante no histórico do SmartSaint. Essas ações são comandadas por meio de 2 atalhos: o primeiro para editar (*Edit*) e alterar a classificação do sítio requisitado; o segundo para visualizar (*View*) o sítio requisitado no navegador padrão do sistema. A Figura 4.9 apresenta a tela de edição quando se pressiona o atalho *Edit* da tela de histórico.

Conforme ilustrado na Figura 4.9, a tela de edição exibe as informações do sítio como

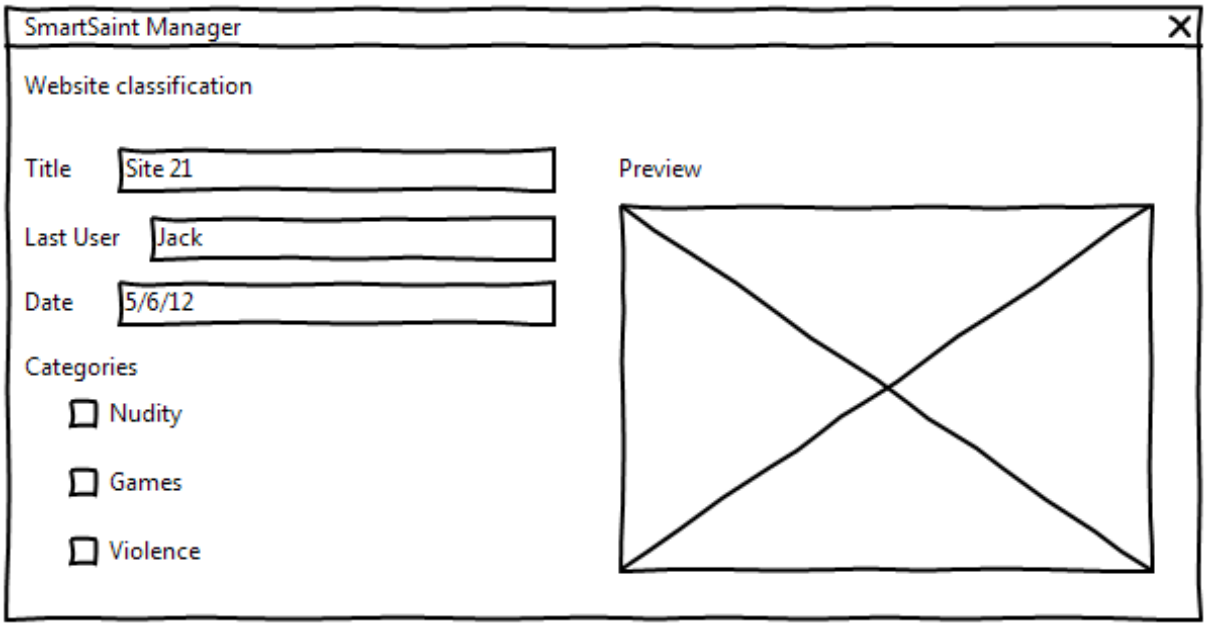


Figura 4.9: Interface de Edição

nome (*Title*), nome do último usuário que requisitou o sítio (*Last User*), a data da última requisição ao sítio (*Date*) e em quais categorias (*Categories*) o sítio foi enquadrado. À direita há ainda uma visualização do sítio (*Preview*). As categorias em que o sítio foi rotulado como proibido aparecerão marcadas, enquanto as categorias em que o sítio foi rotulado como autorizado aparecerão desmarcadas. O moderador poderá sobrescrever a classificação do SmartSaint apenas marcando ou desmarcando as caixas das categorias, fazendo assim com que os classificadores das categorias sejam recalibrados.

Semelhante à interface de histórico, a tela da seção *Pending* (Pendências) é ilustrada na Figura 4.10 e corresponde ao Aprendizado Ativo (*Active Learning*) do SmartSaint. Nessa tela são exibidos sítios pré-selecionados pelo SmartSaint para que o moderador faça a classificação manualmente. As categorias estão seguidas de um ponto de interrogação, o que significa que o SmartSaint sugere estas categorias, porém as mesmas devem ser auditadas pelo moderador. Esses sítios serão selecionados por diversos critérios objetivando manter o classificador eficiente. Dentre os critérios, pode-se citar novamente alguns comentados na seção 3.2:

1. O sítio foi classificado muito próximo à borda de decisão;
2. O sítio está contido em um *cluster* com muitos exemplos heterogêneos;
3. Há um conflito de rotulamento entre a atribuição dos classificadores;
4. Sítios rotulados com alta confiança.

Note que na tela exibida na Figura 4.10, diferentemente da tela de histórico, apesar de o SmartSaint estar questionando se alguns sítios pertencem a uma determinada categoria, ele pode ter autorizado o acesso a ela. Nessa tela de pendências o moderador deve obrigatoriamente clicar na ação *Edit* e confirmar ou desmarcar a categoria do sítio.

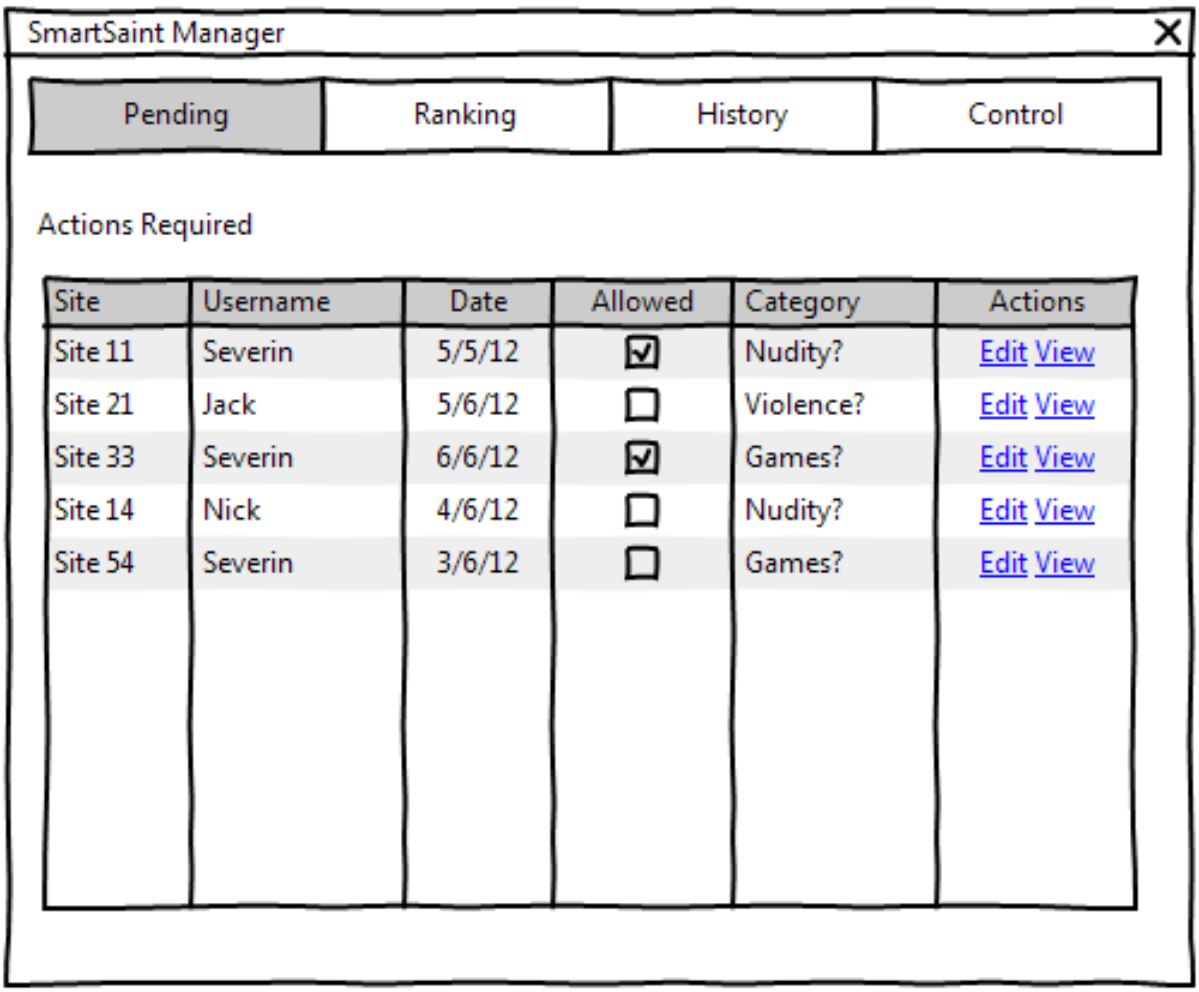


Figura 4.10: Interface de Aprendizado Ativo

Por último, na Figura 4.11 é exibida uma tela referente a seção *Control* (controle). Neste seção o moderador poderá ativar ou desativar a moderação da navegação. Ainda é exibida uma terceira opção: *Restart*. A opção *Restart* fará com que o SmartSaint seja reiniciado, recalibrando os classificadores e corrigindo qualquer inconsistência. Durante a reinicialização do SmartSaint, o acesso a todos os sítios é negado.

A seguir é explicado como foi realizada a análise experimental do SmartSaint, a escolha os algoritmos de classificação usados e a obtenção das bases de dados para os testes.

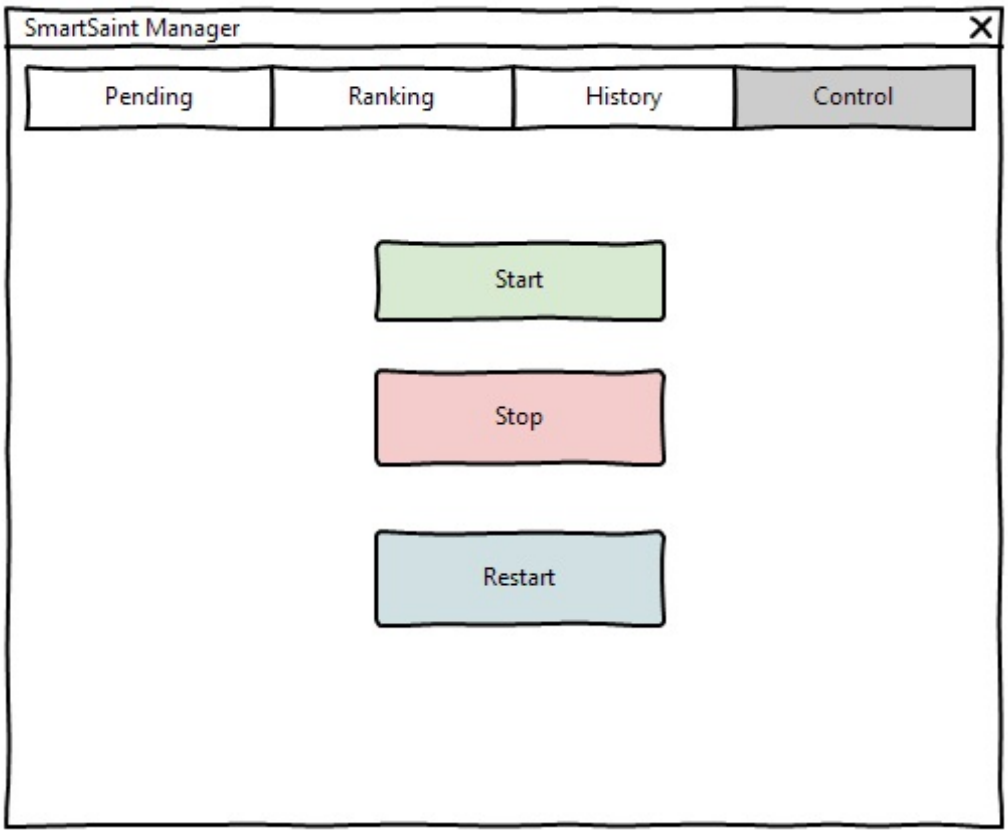


Figura 4.11: Interface de Controle

Capítulo 5

Avaliação Experimental

De maneira a avaliar a performance de classificação e evitar uma modelagem incorreta, primeiro foi conduzido um conjunto de experimentos buscando por classificadores com um alto desempenho para o conjunto de dados. Para chegar a classificadores ainda melhores e reduzir a dimensionalidade do conjunto de dados, também foi realizada a seleção de atributos nos conjuntos de dados.

A seguir, foi implementado para o SmartSaint uma variante do algoritmo Co-Testing. Porém, ao invés de usar duas visões diferentes, são usados dois algoritmos de aprendizado diferentes no conjunto de dados. As seções seguintes descrevem como foram feitos os experimentos e os resultados da avaliação experimental.

5.1 Conjuntos de Dados

Como o objetivo é desenvolver um filtro de internet voltado para famílias interessadas em proteger suas crianças de conteúdo potencialmente perigoso, foram garimpadas bases de dados pensando nas categorias que deveriam ser analisadas para proteger as famílias (crianças). As bases de dados foram coletadas de conteúdos em inglês devido a fonte de dados utilizada, porém as aplicações podem ser expandidas para outros idiomas.

Para compor as bases de dados foram selecionadas as seguintes categorias: *drugs, porn, alcohol, games, dating, medical, banking, news* e *celebrity*. As quatro primeiras categorias estão relacionadas a conteúdos que se deseja filtrar e as últimas cinco a conteúdos comuns em sítios em geral.

As *urls* dos sítios foram coletadas a partir de um sítio que faz catálogos chamado **urlblacklist** (`http://urlblacklist.com`). Foram coletados e analisados aproximadamente mil sítios de cada categoria. A coleta e análise consiste na cópia do conteúdo textual (somente código HTML e textos), remoção das *tags* do HTML, conversão para o formato *bag-of-words* e a remoção das *stopwords*. O conjunto de dados convertido está disponível em `http://lia.facom.ufms.br/wiki/databases`.

Devido a restrições de hardware, os testes foram limitados a não mais que 1500 exemplos com 40 mil atributos. O problema multiclasse foi convertido para vários problemas

binários, usando a estratégia um contra todos.

Os conjuntos de dados são construídos com 300 exemplos de uma das categorias (por exemplo *drugs*), a qual dá o nome ao conjunto de dados, formando os exemplos positivos. Os exemplos negativos são constituídos por 100 exemplos de cada uma das outras categorias (aquelas diferentes da positiva). Isso equivale a 300 exemplos positivos e 800 exemplos negativos no conjunto experimental usado. Foram utilizados 300 exemplos positivos ao invés de 100 (número de exemplos de cada uma das outras categorias) para diminuir o desbalanceamento dos rótulos.

Para cada conjunto de dados, o número de atributos varia, devido a diferença de conteúdo de cada sítio, o que pode ser verificado na Tabela 5.1.

Tabela 5.1: Descrição dos Conjuntos de Dados

Nome do Conjunto	# Atributos	Categorias	Rótulos	% Rótulos
porn	22.309	porn	positiva $\oplus$	27%
		outras(8)	negativa $\ominus$	73%
drugs	25,052	drugs	positiva $\oplus$	27%
		outras(8)	negativa $\ominus$	73%
dating	26.986	dating	positiva $\oplus$	27%
		outras(8)	negativa $\ominus$	73%
alcohol	23.340	alcohol	positiva $\oplus$	27%
		outras(8)	negativa $\ominus$	73%
celebrity	27.646	celebrity	positiva $\oplus$	27%
		outras(8)	negativa $\ominus$	73%
banking	24.801	banking	positiva $\oplus$	27%
		outras(8)	negativa $\ominus$	73%
games	25.615	games	positiva $\oplus$	27%
		outras(8)	negativa $\ominus$	73%
medical	24,010	medical	positiva $\oplus$	27%
		outras(8)	negativa $\ominus$	73%
news	26.996	news	positiva $\oplus$	27%
		outras(8)	negativa $\ominus$	73%

5.2 Classificadores Base

Inicialmente foi feito um teste de performance da classificação de diversos algoritmos e suas variações, usando validação cruzada de 10 partições. Os algoritmos testados incluem:

- k-Nearest Neighbor (KNN)
- Support Vector Machine (SVM)
- Stochastic Gradient Descent (SGD)
- Logistic Regression (LogReg)
- Naive Bayes (NB)

Os algoritmos analisados e suas variações foram obtidos do *framework* `SciKit-learn` (Pedregosa et al., 2011) em **python**. Os algoritmos foram analisados com diferentes parâmetros, de maneira a descobrir como poderia se obter um melhor resultado em termos de AUC ROC (Fawcett, 2006). A Tabela 5.2 apresenta o AUC ROC dos algoritmos de aprendizado de máquina selecionados. Os parâmetros testados para o algoritmo SVM incluem: os valores 0,1, 1 e 10 para o parâmetro de penalidade C e os *kernels* linear, poly e RBF, como sugerido em Forman (2003). Também foi testado uma variação disponível no `SciKit-learn` chamada *LinearSVM*. Para o conjunto de dados coletado, o SVM teve uma performance melhor com parâmetros de $C = 0,1$ e *kernel* linear. Também tentou-se diferentes parâmetros para o algoritmo KNN, como K igual a 3, 5, 7 e 10, assim como sua variação *NearestCentroid*. O KNN se saiu melhor com $K = 3$. Para o algoritmo Naive Bayes testou-se as variações Gaussian, Multinomial e Bernoulli, sendo que a Multinomial foi a que obteve melhores resultados.

Tabela 5.2: Desempenho de Classificação em termos de AUC ROC para cada conjunto de dados e o desvio padrão (entre parênteses)

	LinearSVM	SVM	3NN	SGD	LogReg	MNB
porn	0,937 (0,004)	0,941 (0,005)	0,891 (0,004)	0,913 (0,011)	0,948 (0,003)	0,940 (0,003)
drugs	0,699 (0,008)	0,691 (0,010)	0,657 (0,009)	0,735 (0,008)	0,720 (0,009)	0,773 (0,006)
dating	0,793 (0,009)	0,799 (0,007)	0,726 (0,006)	0,818 (0,006)	0,819 (0,006)	0,823 (0,007)
alcohol	0,823 (0,008)	0,818 (0,004)	0,769 (0,005)	0,827 (0,006)	0,847 (0,004)	0,909 (0,004)
celebrity	0,761 (0,007)	0,775 (0,005)	0,645 (0,009)	0,776 (0,008)	0,786 (0,007)	0,810 (0,010)
banking	0,852 (0,005)	0,851 (0,005)	0,805 (0,006)	0,871 (0,006)	0,870 (0,006)	0,899 (0,004)
games	0,811 (0,007)	0,815 (0,009)	0,763 (0,012)	0,814 (0,010)	0,828 (0,009)	0,868 (0,006)
medical	0,780 (0,005)	0,784 (0,009)	0,661 (0,008)	0,811 (0,010)	0,807 (0,009)	0,863 (0,007)
news	0,839 (0,006)	0,838 (0,007)	0,759 (0,011)	0,841 (0,006)	0,844 (0,009)	0,882 (0,005)

É interessante notar que na média, o SVM não obteve os maiores valores AUC. Foram testados diferentes *kernels* e diferentes valores de C, mas não foram encontrados os parâmetros corretos para que ele se sobressaísse. Os desvios padrões são, na maioria das vezes, menores que 0,01.A partir dos dados da Tabela 5.2 foi executado o teste estatístico

não paramétrico de Friedman, para verificar qual o melhor algoritmo. Na Tabela 5.3 pode ser visto o resultado do ranqueamento dos algoritmos. Os valores representam a posição em termos de *Ranking* e os valores entre parênteses representam a média de AUC ROC de cada algoritmo para o conjunto de dados.

Tabela 5.3: Ranqueamento dos resultados pelo Teste de Friedman e média AUC do classificador (entre parênteses)

	LinearSVM	SVM	3NN	SGD	LogReg	MNB
porn	4,0 (0,937)	2,0 (0,941)	6,0 (0,891)	5,0 (0,913)	1,0 (0,948)	3,0 (0,940)
drugs	4,0 (0,699)	5,0 (0,691)	6,0 (0,657)	2,0 (0,735)	3,0 (0,720)	1,0 (0,773)
dating	5,0 (0,793)	4,0 (0,799)	6,0 (0,726)	3,0 (0,818)	2,0 (0,819)	1,0 (0,823)
alcohol	4,0 (0,823)	5,0 (0,818)	6,0 (0,769)	3,0 (0,827)	2,0 (0,847)	1,0 (0,909)
celebrity	5,0 (0,761)	4,0 (0,775)	6,0 (0,645)	3,0 (0,776)	2,0 (0,786)	1,0 (0,810)
banking	4,0 (0,852)	5,0 (0,851)	6,0 (0,805)	2,0 (0,871)	3,0 (0,870)	1,0 (0,899)
games	5,0 (0,811)	3,0 (0,815)	6,0 (0,763)	4,0 (0,814)	2,0 (0,828)	1,0 (0,868)
medical	5,0 (0,780)	4,0 (0,784)	6,0 (0,661)	2,0 (0,811)	3,0 (0,807)	1,0 (0,863)
news	4,0 (0,839)	5,0 (0,838)	6,0 (0,759)	3,0 (0,841)	2,0 (0,844)	1,0 (0,882)
rank médio	4,444	4,111	6,000	3,000	2,222	1,222

O valor crítico a partir da estatística F é de 40,05. O valor crítico da estatística F com 5 e 40 graus de liberdade e a 95% é de 2,45. De acordo com o teste de Friedman, usando o valor da estatística F, é rejeitada a hipótese-nula de que todos os algoritmos se comportam da mesma forma, pois o valor de 40,05 está acima de 2,45.

Executando o teste Nemenyi *post-hoc* para verificar se é possível identificar diferença entre os algoritmos, o valor crítico para comparação da média dos *rankings* de dois algoritmos diferentes a 95% é de 2,51. As diferenças de média dos *rankings* acima desse valor são significativas. Na Tabela 5.4 é exibida uma comparação de média dos *rankings* dos algoritmos levando em conta esse valor crítico.

Tabela 5.4: Algoritmos que tiveram uma melhor performance são marcados com +, e os piores com −, e onde não é possível detectar diferença está marcado com *o*.

	MNB	LogReg	SGD	SVM	LinearSVM	3NN
MNB	*o*	*o*	*o*	+	+	+
LogReg	*o*	*o*	*o*	*o*	*o*	+
SGD	*o*	*o*	*o*	*o*	*o*	+
SVM	−	*o*	*o*	*o*	*o*	*o*
LinearSVM	−	*o*	*o*	*o*	*o*	*o*
3NN	−	−	−	*o*	*o*	*o*

Com base nessas análises, escolheu-se os dois algoritmos que tiveram uma melhor performance, com estratégias diferentes: um com característica gerativa (Multinomial Naive Bayes) e outro com característica discriminativa (Logistic Regression).

Para resultados mais detalhados, veja `http://lia.facom.ufms.br/wiki/smartsaint`. Na próxima seção é explicado como procederam os experimentos com o algoritmo do SmartSaint.

5.3 Classificação de Sítios

Como mencionado, na classificação de textos um grande problema é a alta dimensão dos conjuntos de atributos. É muito comum nesses problemas que haja dezenas de milhares de atributos. A maioria desses atributos não são relevantes ou vantajosos para a tarefa de classificação de textos. Ainda, alguns atributos representam ruídos e podem reduzir a precisão da classificação. Além disso, um alto número de atributos pode deixar o processo de classificação lento ou ainda fazer com que alguns classificadores se tornem inaplicáveis. Por isso a seleção de atributos é comumente usada na classificação de textos, para reduzir a dimensionalidade do domínio de atributos e aumentar a eficiência e precisão dos classificadores (Chen et al., 2009).

Dois métodos de seleção de atributos foram analisados. O método de seleção de atributos Chi2, muito usado na literatura, é fornecido pelo `SciKit-learn` . Ainda foi implementado um outro método de seleção de atributos indicado na literatura pelo bom desempenho na classificação de textos, chamado Bi-Normal Separation (BNS) (Forman, 2003). Cada seletor possui o nome do valor estatístico que ele calcula para cada combinação de classe/atributo. O seletor Chi2 calcula o valor estatístico chi-quadrado, que é um valor da dispersão para duas variáveis de escala nominal. O seletor BNS calcula o valor da separação Bi-Normal.

Os experimentos seguintes foram conduzidos usando validação cruzada de 10 partições e 15 passos de execução. As execuções foram limitadas em 15 passos, pois os dados passavam a convergir a partir de 12º passo. No passo inicial de execução o conjunto de exemplos rotulados L contém 50 exemplos, número escolhido por ter uma boa diversidade de sítios e uma seleção de atributos satisfatória. A cada passo (iteração) da execução, até 10 exemplos são enviados a um oráculo (o moderador do aprendizado ativo) e adicionados ao conjunto de exemplos rotulados. Foram usados diferentes métodos de escolha para decidir quais exemplos seriam adicionados ao conjunto de rotulados. Nos testes executados, os métodos conseguiram obter 10 exemplos para serem enviados ao oráculo em todas as iterações. A seguir são descritos os 3 métodos de escolha de exemplos que tiveram melhor desempenho entre os analisados:

1. Estratégia 1 (MNB_1 e LogReg_1): a maior diferença de *score* (confiança na classificação) dos algoritmos. Isso é equivalente a selecionar os exemplos com maior discordância entre os algoritmos, isto é, ambos os classificadores tem alta confiança, mas discordam na classificação;

2. Estratégia 2 (MNB_2 e LogReg_2): discordância na classificação, sem considerar o nível de confiança (*score*);

3. Estratégia 3 (MNB_3 e LogReg_3): aleatório. São escolhidos 10 exemplos ao acaso (usado como *baseline*).

Além dessas três estratégias são exibidos valores chamados de MNB e LogReg, que representam a média de AUC da validação cruzada de 10 partições de um algoritmo supervisionado puro, usado para comparação.

A seguir são mostrados os resultados dos experimentos realizados para escolher os melhores algoritmos e estratégias para uso com o SmartSaint.

Experimento 1: Vale a pena utilizar aprendizado semissupervisionado no problema apresentado?

A comparação a ser realizada para responder a essa pergunta consiste em comparar: resultado da classificação sem aprendizado semissupervisionado e com semissupervisionado. Assume-se que existam 50 exemplos rotulados disponíveis de cada classe. Pelo gráfico de desempenho, na maioria dos experimentos executados o aprendizado semissupervisionado estabilizava o seu resultado por volta da 12ª iteração. Assim, o aprendizado semissupervisionado foi executado 14 vezes (iterações) e a cada iteração são incrementados 10 exemplos utilizando a Estratégia 3.

Na Tabela 5.5 são apresentados os valores de AUC ROC dos algoritmos Multinomial Naive Bayes (MNB) e Regressão Logística (LogReg). Os valores entre parênteses indicam o desvio padrão. Pode-se notar que, com aprendizado semissupervisionado, os valores AUC são sempre maiores nos conjuntos de dados e algoritmos testados. Para verificar se existe diferença significativa os valores obtidos foram submetidos ao teste t-pareado com 95% de intervalo de confiança. A última coluna da tabela expressa se existe diferença significativa.

A análise utilizando teste t-pareado para comparar as duas amostras, que serve para verificar a diferença entre os resultados, indicou que na maioria dos casos há diferença significativa. Porém esse teste não pode ser utilizado para comparar a abordagem com e sem aprendizado semissupervisionado devido a correção de bonferroni. Para realizar a comparação entre as duas abordagens foi realizado o teste de Wilcoxon.

A soma dos rankings com aprendizado semissupervisionado é de 171. A soma dos rankings sem aprendizado semissupervisionado é zero. Assim, de acordo com o teste de Wilcoxon, o p-valor é 40 com 95% de confiança. Assim, pode-se rejeitar a hipótese nula, ou seja, existe diferença significativa entre os resultados.

Com as análises dos testes, pode-se dizer que existe diferença significativa com um grau de 95% de confiança. Então o aprendizado semissupervisionado apresenta um maior valor de AUC. Assim, a recomendação a partir deste experimento é que seja utilizado aprendizado semissupervisionado com melhora significativa dos resultados.

Após concluir que o uso do aprendizado semissupervisionado era adequado, precisou-se analisar qual estratégia de escolha deveria ser usado para enviar exemplos para a moderação ativa. A seguir são apresentados os resultados desse experimento.

Tabela 5.5: AUC ROC sem e com aprendizado semissupervisionado e a diferença significativa entre eles. Os valores entre parênteses indicam os desvios padrões.

conjunto	algoritmo	sem semissup.	com semissup.	dif. sig.
alcohol	MNB	0,725 (0,073)	0,833 (0,054)	sim
	LogReg	0,800 (0,039)	0,888 (0,026)	sim
porn	MNB	0,899 (0,050)	0,929 (0,021)	não
	LogReg	0,924 (0,042)	0,973 (0,020)	sim
drugs	MNB	0,703 (0,058)	0,775 (0,032)	sim
	LogReg	0,688 (0,034)	0,784 (0,042)	sim
celebrity	MNB	0,685 (0,086)	0,770 (0,065)	não
	LogReg	0,667 (0,070)	0,781 (0,040)	sim
banking	MNB	0,828 (0,058)	0,908 (0,028)	sim
	LogReg	0,853 (0,070)	0,904 (0,046)	não
games	MNB	0,774 (0,060)	0,857 (0,049)	sim
	LogReg	0,777 (0,051)	0,847 (0,054)	sim
news	MNB	0,756 (0,060)	0,868 (0,030)	sim
	LogReg	0,678 (0,095)	0,856 (0,030)	sim
dating	MNB	0,744 (0,079)	0,840 (0,052)	sim
	LogReg	0,726 (0,103)	0,835 (0,041)	sim
medical	MNB	0,697 (0,087)	0,836 (0,040)	sim
	LogReg	0,733 (0,070)	0,834 (0,066)	sim

Experimento 2: Existe diferença significativa entre as estratégias 1, 2 e 3 de escolha de exemplos para rotular no CoLabeling?

Para verificar a diferença entre as estratégias 1, 2 e 3, o CoLabeling foi executado comparando o AUC ROC na iteração 14 de todos os conjuntos de dados. Para verificar diferença significativa foi utilizado o teste de Friedman. Para uma análise mais criteriosa, o desempenho dos algoritmos Regressão Logística e Multinomial Naive Bayes são verificados separadamente. A Tabela 5.6 apresenta os *Rankings* das estratégias utilizadas para escolher os exemplos a serem rotulados pelo oráculo para o Algoritmo Regressão Logística e, entre parênteses, é mostrado o valor AUC para o conjunto de dados.

Tabela 5.6: *Ranking* das estratégias para o Algoritmo Regressão Logística e entre parênteses o valor AUC

	Estratégia 1	Estratégia 2	Estratégia 3
alcohol	2,0 (0,905)	1,0 (0,910)	3,0 (0,888)
porn	3,0 (0,970)	1,0 (0,984)	2,0 (0,973)
drugs	1,0 (0,801)	2,0 (0,800)	3,0 (0,784)
celebrity	1,0 (0,845)	2,0 (0,807)	3,0 (0,781)
banking	2,0 (0,909)	1,0 (0,931)	3,0 (0,904)
games	1,0 (0,903)	2,0 (0,877)	3,0 (0,847)
news	1,0 (0,919)	2,0 (0,888)	3,0 (0,856)
dating	1,0 (0,875)	2,0 (0,853)	3,0 (0,835)
medical	2,0 (0,867)	1,0 (0,868)	3,0 (0,834)
rank médio	1,556	1,556	2,889

A estatística F para o conjunto de dados classificado pelo algoritmo Regressão Logística (apresentados na Tabela 5.6) é de 11,64. O valor crítico da estatística F para 2 e 16 graus

de liberdade é de 3,63 com 95% de confiança. Assim, de acordo com o teste de Friedman, pode-se rejeitar a hipótese nula que os algoritmos possuem resultados similares.

Prosseguindo com a análise com um teste *post-hoc* para verificar se existe diferença significativa entre as estratégias foi aplicado o teste de Nemenyi. A Tabela 5.7 mostra que as estratégias 1 e 2 tiveram desempenhos semelhantes, enquanto foram significativamente melhores que a estratégia 3.

Tabela 5.7: Estratégias que tiveram melhor performance (com o Algoritmo Regressão Logística) são marcadas com +, e as piores com −, e onde não é possível detectar diferença está marcado com *o*.

	rank médio	Estratégia 2	Estratégia 1	Estratégia 3
Estratégia 2	1,556	*o*	*o*	+
Estratégia 1	1,556	*o*	*o*	+
Estratégia 3	2,889	−	−	*o*

A Tabela 5.8 apresenta os *Rankings* das estratégias utilizadas para escolher os exemplos a serem rotulados pelo oráculo para o Algoritmo Multinomial Naive Bayes e, entre parênteses, é mostrado o valor AUC para o conjunto de dados.

Tabela 5.8: *Ranking* das estratégias para o Algoritmo Multinomial Naive Bayes e entre parênteses o valor AUC

	Estratégia 1	Estratégia 2	Estratégia 3
alcohol	2,0 (0,905)	1,0 (0,910)	3,0 (0,888)
porn	3,0 (0,970)	1,0 (0,984)	2,0 (0,973)
drugs	1,0 (0,801)	2,0 (0,800)	3,0 (0,784)
celebrity	1,0 (0,845)	2,0 (0,807)	3,0 (0,781)
banking	2,0 (0,909)	1,0 (0,931)	3,0 (0,904)
games	1,0 (0,903)	2,0 (0,877)	3,0 (0,847)
news	1,0 (0,919)	2,0 (0,888)	3,0 (0,856)
dating	1,0 (0,875)	2,0 (0,853)	3,0 (0,835)
medical	2,0 (0,867)	1,0 (0,868)	3,0 (0,834)
rank médio	1,556	1,500	2,944

A estatística F para o conjunto de dados classificado pelo algoritmo Naive Bayes (apresentados na Tabela 5.8) é de 16,22. O valor crítico da estatística F para 2 e 16 graus de liberdade é de 3,63 com 95% de confiança. Assim, de acordo com o teste de Friedman, pode-se rejeitar a hipótese nula que os algoritmos possuem resultados similares.

Prosseguindo com a análise com um teste *post-hoc* para verificar se existe diferença significativa entre as estratégias, foi aplicado o teste de Nemenyi. A Tabela 5.9 mostra que as estratégias 1 e 2 tiveram desempenhos semelhantes, enquanto foram significativamente melhores que a estratégia 3.

Assim, a partir deste experimento verificou-se que uma estratégia de escolha de exemplos a rotular deve ser mantida. No SmartSaint é usado a estratégia 1 por ser ligeiramente melhor, mas sem diferença significativa, que a estratégia 2.

Tabela 5.9: Estratégias que tiveram uma melhor performance (com o Algoritmo Multinomial Naive Bayes) são marcadas com +, e as piores com −, e onde não é possível detectar diferença está marcado com *o*.

	rank médio	Estratégia 2	Estratégia 1	Estratégia 3
Estratégia 2	1,500	*o*	*o*	+
Estratégia 1	1,556	*o*	*o*	+
Estratégia 3	2,944	−	−	*o*

A seguir havia a necessidade de verificar a possibilidade de melhorar ainda mais a precisão de classificação e ainda deixar o classificador mais rápido. Para isso resolveu-se analisar o uso de seleção de atributos, conforme mostrado no próximo experimento.

Experimento 3: Comparação com e sem seleção de atributos

Para verificar o efeito da seleção de atributos sobre o desempenho do algoritmo, o CoLabeling foi executado comparando o AUC ROC na iteração 14 de todos os conjuntos de dados sem seleção de atributos, com o seletor Chi2 e com o seletor BNS. Como estratégia de escolha de exemplos a rotular, foi escolhido a estratégia 3 por ser a mais imparcial das 3, pois escolhe os exemplos aleatoriamente. Para verificar diferença significativa foi utilizado o teste de Friedman. Para uma análise mais criteriosa é verificado o desempenho dos algoritmos Regressão Logística e Multinomial Naive Bayes separadamente. A Tabela 5.10 apresenta os *Rankings* dos métodos de seleção de atributos utilizados para o Algoritmo Regressão Logística e, entre parênteses, é mostrado o valor AUC para o conjunto de dados.

Tabela 5.10: *Ranking* dos seletores para o Algoritmo Regressão Logística e entre parênteses o valor AUC

	Sem seleção	Seletor Chi2	Seletor BNS
alcohol	3,0 (0,888)	2,0 (0,891)	1,0 (0,900)
porn	3,0 (0,973)	1,0 (0,978)	2,0 (0,977)
drugs	1,0 (0,784)	2,0 (0,767)	3,0 (0,764)
celebrity	2,0 (0,781)	3,0 (0,769)	1,0 (0,792)
banking	3,0 (0,904)	2,0 (0,909)	1,0 (0,925)
games	3,0 (0,847)	2,0 (0,850)	1,0 (0,864)
news	3,0 (0,856)	2,0 (0,859)	1,0 (0,869)
dating	3,0 (0,835)	2,0 (0,847)	1,0 (0,855)
medical	2,5 (0,834)	2,5 (0,834)	1,0 (0,843)
rank médio	2,611	2,056	1,333

A estatística F para o conjunto de dados classificado pelo algoritmo Regressão Logística (apresentados na Tabela 5.10) é de 5,57. O valor crítico da estatística F para 2 e 16 graus de liberdade é de 3,63 com 95% de confiança. Assim, de acordo com o teste de Friedman, pode-se rejeitar a hipótese nula que os seletores possuem resultados similares.

Prosseguindo com a análise com um teste *post-hoc* para verificar se existe diferença significativa entre os seletores foi aplicado o teste de Nemenyi. A Tabela 5.11 mostra que

o seletor BNS teve uma performance melhor que quando não há seleção de atributos e uma performance similar a do Chi2.

Tabela 5.11: Seletores que tiveram uma melhor performance (com o Algoritmo Regressão Logística) são marcados com +, e os piores com −, e onde não é possível detectar diferença está marcado com *o*.

	rank médio	BNS	Chi2	Nenhum
BNS	1,333	*o*	*o*	+
Chi2	2,056	*o*	*o*	*o*
Nenhum	2,611	−	*o*	*o*

A Tabela 5.12 apresenta os *Rankings* métodos de seleção de atributos utilizados para o Algoritmo Multinomial Naive Bayes e, entre parênteses, é mostrado o valor AUC para o conjunto de dados.

Tabela 5.12: *Ranking* dos seletores para o Algoritmo Multinomial Naive Bayes e entre parênteses o valor AUC

	Sem seleção	Seletor Chi2	Seletor BNS
alcohol	3,0 (0,888)	2,0 (0,891)	1,0 (0,900)
porn	3,0 (0,973)	1,0 (0,978)	2,0 (0,977)
drugs	1,0 (0,784)	2,0 (0,767)	3,0 (0,764)
celebrity	2,0 (0,781)	3,0 (0,769)	1,0 (0,792)
banking	3,0 (0,904)	2,0 (0,909)	1,0 (0,925)
games	3,0 (0,847)	2,0 (0,850)	1,0 (0,864)
news	3,0 (0,856)	2,0 (0,859)	1,0 (0,869)
dating	3,0 (0,835)	2,0 (0,847)	1,0 (0,855)
medical	2,5 (0,834)	2,5 (0,834)	1,0 (0,843)
rank médio	2,611	2,056	1,333

A estatística F para o conjunto de dados classificado pelo algoritmo Naive Bayes (apresentados na Tabela 5.12) é de 5,57. O valor crítico da estatística F para 2 e 16 graus de liberdade é de 3,63 com 95% de confiança. Assim, de acordo com o teste de Friedman, pode-se rejeitar a hipótese nula que os algoritmos possuem resultados similares.

Prosseguindo com a análise com um teste *post-hoc* para verificar se existe diferença significativa entre os seletores foi aplicado o teste de Nemenyi. A Tabela 5.13 mostra que o seletor BNS teve uma performance melhor que quando não há seleção de atributos e uma performance similar a do Chi2.

Os resultados são promissores pois indicam que a seleção de atributos pode de fato melhorar a precisão do aprendizado semissupervisionado. No SmartSaint é usado a seleção de atributos pelo seletor BNS por apresentar melhor desempenho.

A seguir é feito uma análise de desempenho de todos os algoritmos a cada iteração da execução.

Tabela 5.13: Seletores que tiveram uma melhor performance (com o Algoritmo Multinomial Naive Bayes) são marcados com +, e os piores com −, e onde não é possível detectar diferença está marcado com *o*.

	rank médio	BNS	Chi2	Nenhum
BNS	1,333	*o*	*o*	+
Chi2	2,056	*o*	*o*	*o*
Nenhum	2,611	−	*o*	*o*

Experimento 4: Desempenho dos algoritmos em cada iteração

Nos experimentos anteriores, foram analisados apenas a primeira e a última iteração. Nesse experimento será feito uma análise mais detalhada do desempenho dos algoritmos. Para isso serão analisados todos os algoritmos, com diferentes estratégias de escolha de exemplos e diferentes seletores para seleção de atributos, ao longo de todas as iterações da execução. Devido ao grande volume de dados gerado nesse experimento serão analisados a seguir apenas os resultados mais relevantes.

A Tabela 5.14 apresenta o resultado da execução sem seleção sobre o conjunto de dados nomeado *dating*, no qual *dating* é a classe positiva e as outras categorias formam a classe negativa, conforme explicado na Seção 5.1. Nessa tabela cada coluna representa o valor de AUC para cada algoritmo após cada iteração, conforme explicado no início da seção.

Na Figura 5.1 item (a) podem ser vistos os dados da tabela plotados para melhor visualização. O eixo x representa os valores de AUC ROC de cada um dos algoritmos e o eixo y representa os passos (iterações) da execução. Em $\mathbf{MNB}_1$ é ilustrada a evolução dos valores de AUC do algoritmo Multinomial Naive Bayes, usando a maior diferença de *score* entre ele e o algoritmo Logistic Regression (estratégia 1). Analogamente em $\mathbf{LogReg}_1$ é ilustrada a evolução da performance do algoritmo Logistic Regression, usando a maior diferença de *score* entre ele e o algoritmo Multinomial Naive Bayes. Em $\mathbf{MNB}_2$ e $\mathbf{LogReg}_2$, é ilustrada a performance dos algoritmos quando a escolha dos novos elementos a serem rotulados pelo oráculo é feita pela discordância dos algoritmos sem considerar o *score* (estratégia 2). Em $\mathbf{MNB}_3$ e $\mathbf{LogReg}_3$ é ilustrada a performance dos algoritmos usando a escolha de novos elementos de maneira aleatória para referência. E finalmente, abaixo da tabela são exibidos os valores de **MNB** e **LogReg**, que são a versão supervisionada do algoritmo, usando todos os elementos rotulados.

Tabela 5.14: Valores AUC ROC para classificação do conjunto de dados *dating* sem seleção.

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,729	0,742	0,795	0,803	0,819	0,837	0,835	0,840	0,844	0,848	0,854	0,856	0,860	0,863	0,863
$LogReg_1$	0,741	0,764	0,789	0,819	0,833	0,839	0,844	0,845	0,840	0,840	0,851	0,861	0,865	0,867	0,874
MNB_2	0,737	0,812	0,832	0,815	0,841	0,836	0,844	0,843	0,842	0,836	0,846	0,851	0,856	0,849	0,856
$LogReg_2$	0,761	0,802	0,826	0,834	0,841	0,843	0,846	0,847	0,850	0,854	0,859	0,855	0,861	0,857	0,852
MNB_3	0,743	0,763	0,772	0,781	0,796	0,807	0,820	0,830	0,827	0,834	0,842	0,842	0,840	0,838	0,839
$LogReg_3$	0,726	0,740	0,747	0,766	0,788	0,794	0,804	0,821	0,822	0,824	0,838	0,832	0,831	0,836	0,834

MNB supervisionado obteve 0,861 e LogReg supervisionado obteve 0,909.

Tabela 5.15: Valores AUC ROC para classificação do conjunto de dados *dating* usando o seletor chi2

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,740	0,748	0,777	0,789	0,818	0,834	0,830	0,842	0,849	0,851	0,858	0,859	0,859	0,862	0,861
$LogReg_1$	0,729	0,758	0,782	0,809	0,823	0,834	0,845	0,850	0,850	0,852	0,858	0,859	0,866	0,872	0,874
MNB_2	0,737	0,785	0,801	0,800	0,820	0,816	0,828	0,831	0,836	0,831	0,837	0,844	0,846	0,847	0,853
$LogReg_2$	0,739	0,777	0,793	0,808	0,816	0,821	0,826	0,831	0,832	0,839	0,840	0,841	0,855	0,849	0,848
MNB_3	0,776	0,781	0,782	0,793	0,802	0,810	0,817	0,824	0,825	0,828	0,831	0,837	0,832	0,835	0,837
$LogReg_3$	0,740	0,755	0,766	0,780	0,792	0,799	0,814	0,822	0,822	0,829	0,836	0,840	0,838	0,849	0,846

MNB supervisionado obteve 0,852 e LogReg supervisionado obteve 0,915.

Tabela 5.16: Valores AUC ROC para classificação do conjunto de dados *dating* usando o seletor BNS

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,750	0,754	0,779	0,790	0,815	0,826	0,832	0,847	0,848	0,851	0,855	0,861	0,856	0,864	0,865
$LogReg_1$	0,747	0,765	0,791	0,811	0,829	0,833	0,850	0,859	0,852	0,856	0,860	0,863	0,864	0,877	0,879
MNB_2	0,761	0,795	0,808	0,810	0,824	0,829	0,838	0,840	0,845	0,842	0,850	0,854	0,853	0,855	0,860
$LogReg_2$	0,761	0,792	0,809	0,822	0,828	0,838	0,846	0,848	0,850	0,855	0,854	0,856	0,862	0,861	0,863
MNB_3	0,776	0,772	0,779	0,790	0,798	0,808	0,815	0,826	0,827	0,830	0,834	0,838	0,836	0,840	0,842
$LogReg_3$	0,758	0,764	0,781	0,791	0,805	0,813	0,826	0,831	0,834	0,840	0,845	0,850	0,850	0,855	0,854

MNB supervisionado obteve 0,874 e LogReg supervisionado obteve 0,919.

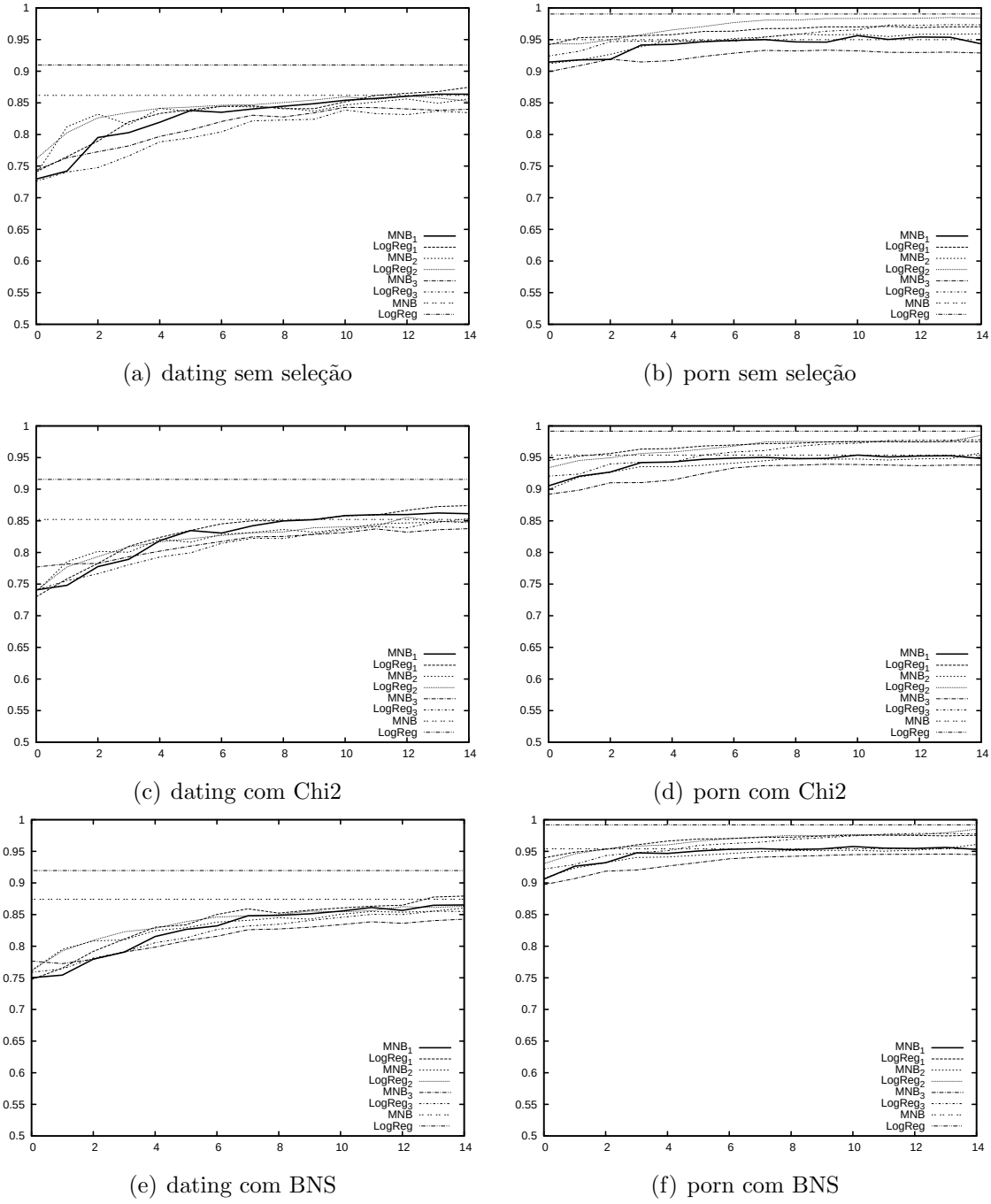
Os melhores resultados sem o uso de seleção de atributos podem ser observados na iteração 14 da Tabela 5.14 e na Figura 5.1 item (a). Eles são obtidos por **LogReg** (supervisionado puro) e $\mathbf{LogReg}_1$ (estratégia 1), seguido de $\mathbf{MNB}_1$ (estratégia 1). O que indica que, nesse caso, com uma fração de exemplos rotulados o semissupervisionado pode obter um resultado semelhante ao supervisionado puro.

A Tabela 5.15 apresenta os resultados do mesmo conjunto de dados, porém usando seleção de atributos com o seletor Chi2. Na Figura 5.1 item (c) são plotados esses dados. É interessante notar que o MNB_1 e $LogReg_1$ tem uma performance melhor que o MNB após a 9^a iteração. Note que o MNB é treinado com todo o conjunto de dados, enquanto o MNB_1 e $LogReg_1$ após a 9^a iteração é treinado apenas com 130 exemplos.

Por último, na Tabela 5.16 são apresentados os dados do conjunto *dating* usando o seletor BNS como método de seleção de atributos e plotados na Figura 5.1 item (e). O $LogReg_1$ tem o melhor desempenho entre os algoritmos semissupervisionados, ultrapassando o MNB na 14^a iteração. Vale ressaltar novamente que os algoritmos semissupervisionados ao final da 15^a iteração usaram apenas 190 exemplos rotulados, enquanto os algoritmos supervisionados usaram mais de mil exemplos para ter um desempenho semelhante.

Na Figura 5.1, ainda pode-se ver os dados do conjunto **porn** sem seleção de atributos (item (b)), com seleção de atributos usando o Chi2 (item(d)) e com seleção de atributos usando o BNS (item(f)). A categoria **porn** é uma categoria que os classificadores tem um alta taxa de precisão, sendo mais fácil separá-la das outras categorias. Como pode ser observado nas figuras, os valores de AUC de **porn** são mais altos que a categorias **dating**, sendo muito próximos de 1 (100% de acertos). Na Figura 5.1 item (b), sem seleção de atributos, o algoritmo $LogReg_1$ ultrapassa MNB logo após a 2^a iteração com apenas 60

Figura 5.1: Valores AUC ROC dos conjuntos de dados *dating* e *porn* ao longo das iterações



(a) dating sem seleção

(b) porn sem seleção

(c) dating com Chi2

(d) porn com Chi2

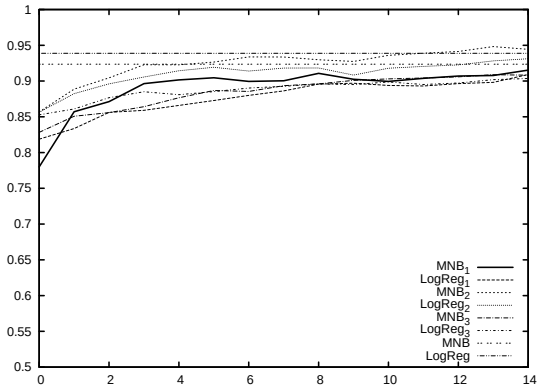
(e) dating com BNS

(f) porn com BNS

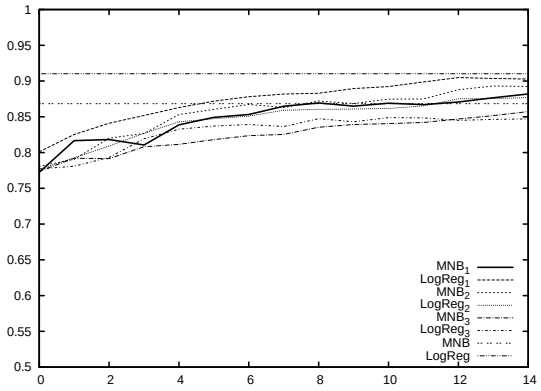
exemplos rotulados. Essa é uma categoria em que quase todos os classificadores semissupervisionados ultrapassam o MNB usando seleção de atributos ou não. As abordagens aleatórias (MNB_3 e $LogReg_3$) que representam um simples algoritmo semissupervisionado com aprendizado ativo, mostram que o aprendizado ativo pode de fato ajudar a melhorar a classificação.

Na Figura 5.2, é possível visualizar a plotagem de alguns dos experimentos com os valores AUC resultantes da classificação de conjuntos como **banking**, **games**, **drugs** e **celebrity** sem seleção de atributo e com seleção de atributos usando o Chi2. Na Figura 5.2 item (a), o algoritmo MNB_2 consegue empatar com o algoritmo supervisionado MNB na 4ª

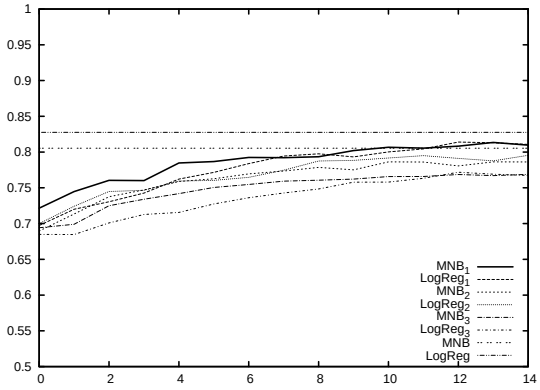
Figura 5.2: Valores AUC ROC de alguns resultados dos conjuntos de dados *banking*, *games*, *drugs* e *celebrity* ao longo das iterações



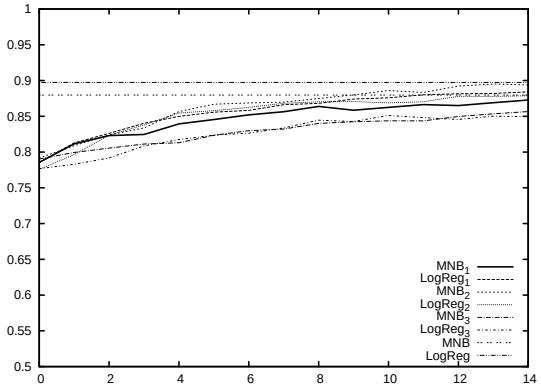
(a) banking sem seleção



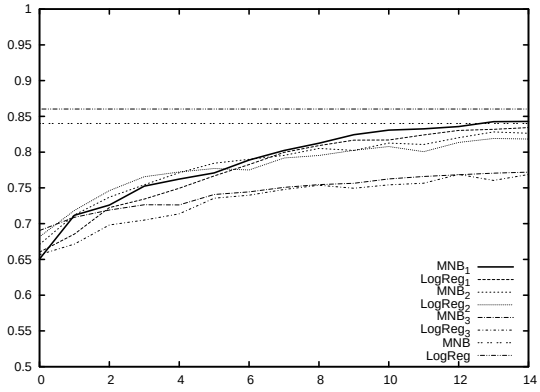
(b) games sem seleção



(c) drugs com Chi2



(d) games com Chi2



(e) celebrity com Chi2

iteração e ultrapassar os 2 algoritmos supervisionados (MNB e LogReg) na 14ª iteração. O algoritmo LogReg_2 consegue ultrapassar o MNB na 13ª iteração. Na diversidade da Figura 5.2 fica evidente que os algoritmos tem precisões diferentes, dependendo da categoria a ser classificada. Dentro desse conjunto o **drugs** fica com os menores valores (menor precisão) e o **banking** fica com os maiores valores (maior precisão). Os dados usados para gerar esses gráficos podem ser consultados no Apêndice C.

Os gráficos das outras categorias analisadas encontram-se no Apêndice B. No Apêndice A podem ser visualizadas as curvas ROC que resultaram nos valores AUC da última

iteração de execução para os conjuntos de dados. Os pontos onde pode-se ver mudanças de direção da curva ROC são usados pelo SmartSaint para que o usuário possa definir diferentes limiares de classificação.

Capítulo 6

Conclusões

Neste trabalho foi apresentado o sistema moderador de conteúdo SmartSaint, assim como sua implementação, seus algoritmos e a ferramenta BagMaker usada por ele. O SmartSaint adequa-se ao paradigma do Aprendizado Semissupervisionado, que é uma classe de problemas de crescente relevância no contexto do Aprendizado de Máquina: aquela onde há um grande quantidade de exemplos não rotulados e um número de exemplos rotulados escasso. Um dos motivos para escassez de exemplos rotulados deve-se especialmente ao alto custo de rotulação de exemplos, como é o caso da rotulação de sítios da internet. Em adição, o SmartSaint faz proveito do Aprendizado Ativo, que visa atingir altas taxas de acerto utilizando alguns exemplos rotulados por um especialista, desta forma, minimizando o custo da obtenção de dados rotulados. Assim facilitando a tarefa do administrador de redes ou pai de família e tornando acessível a moderação de conteúdo.

O uso conjunto de Aprendizado Semissupervisionado Ativo, aplicado a moderação de sítios é algo novo. É uma abordagem que se adapta a realidade da internet, que está em constante mudança. Após um aperfeiçoamento, o sistema poderá funcionar tanto em um computador pessoal, filtrando conteúdo para controle dos pais, quanto em instituições onde servidores filtram as requisições de milhares de usuários.

Neste trabalho foi realizado uma avaliação experimental que cobre os principais questionamentos levantados durante o projeto sobre o uso das técnicas de aprendizado semissupervisionado, aprendizado ativo e seleção de atributos. Inicialmente, após uma verificação de desempenho entre vários algoritmos supervisionados para conjuntos de dados da classe de problemas que este trabalho visa solucionar foram escolhidos os algoritmos *Multinomial Naive Bayes* (MNB) e *Logistic Regression* (LogReg) como algoritmo supervisionado base. Com isso foi constatado o que pode ser encontrado na literatura que o MNB consegue lidar bem com bases textuais, assim como o LogReg.

Foram exploradas e colocadas a prova algumas propostas encontradas na literatura. Os experimentos demonstraram que a abordagem proposta resulta em um classificador de textos rápido e preciso para o problema de Moderação de Sítios. Os resultados indicam que usando seleção de atributos e Aprendizado Semissupervisionado Ativo é possível reduzir

o número de exemplos necessários a 10% do utilizado por um algoritmo Supervisionado puro.

Os resultados são motivadores dado a fração do número de exemplos de treinamento necessários no Aprendizado Semissupervisionado Ativo, em comparação com os classificadores tradicionais.

Contribuições

Este trabalho contribuiu com a criação de uma aplicação do Aprendizado Semissupervisionado Ativo com o uso de seleção de atributos para o problema de moderação de sítios da internet.

Foi construído um banco de dados com dados reais de mais de 10 mil sítios divididos em 9 categorias, os quais podem ser usados por outros pesquisadores para análises futuros.

Ainda criou-se uma boa ferramenta de Pré-Processamento de textos HTML, o BagMaker, que pode ser usada e expandida para diversos outros problemas.

Trabalhos Futuros

Futuramente deve-se adicionar uma melhoria ao SmartSaint com o uso de Aprendizado Online para assegurar que os classificadores não reduzam sua performance com o uso de exemplos antigos não mais relevantes, e também para que possam detectar possíveis propagações de ruídos e erros, reduzindo ainda mais a interação do usuário.

Também será produzida no futuro uma interface amigável para controle do SmartSaint nos moldes do que foi proposto na Seção 4.5, de maneira que o usuário possa personalizar o SmartSaint e usá-lo mais facilmente.

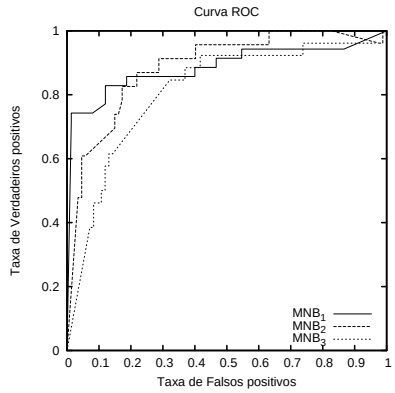
Outras melhorias que poderiam ser feitas incluem análise de links que a página referencia, o conteúdo das imagens do sítio, o conteúdo de outros tipos de arquivos como javascript ou css, entre outros.

Apêndice A

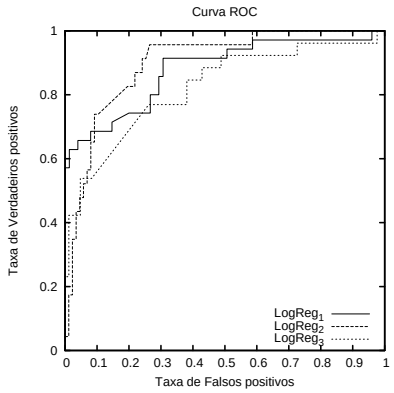
Curvas ROC

A seguir estão plotadas as curvas ROC da última iteração dos experimentos de classificação de sítios tratados na Seção 5.3. Os pontos onde pode-se ver mudança de direção da curva ROC são usados pelo SmartSaint para que o usuário possa definir diferentes limiares de classificação.

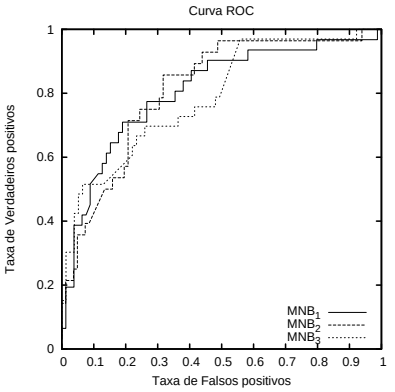
Figura A.1: Curvas ROC dos conjuntos de dados *dating* e *drugs*



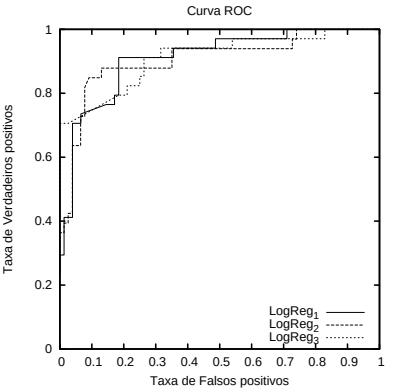
(a) dating MNB



(b) dating LogReg

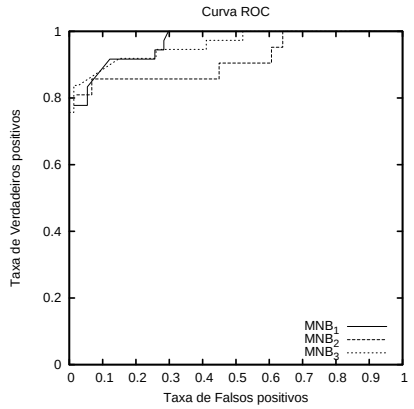


(c) drugs MNB

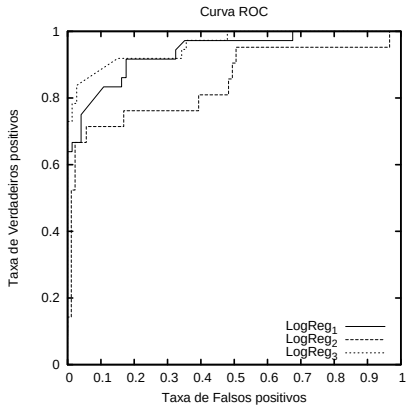


(d) drugs

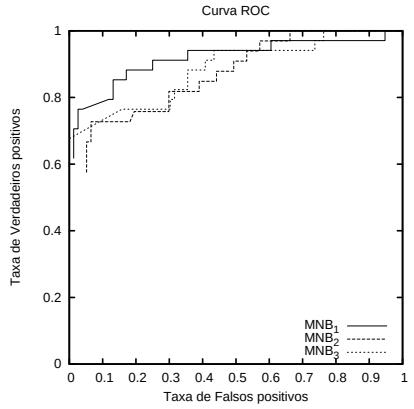
Figura A.2: Curvas ROC dos conjuntos de dados *banking*, *games*, *medical* e *porn*



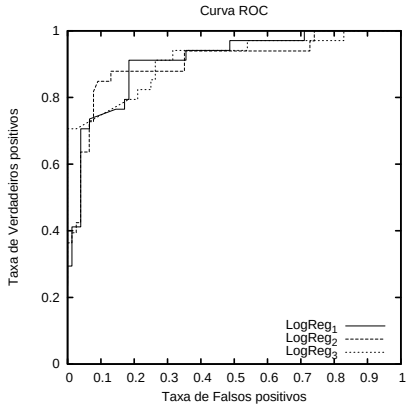
(a) banking MNB



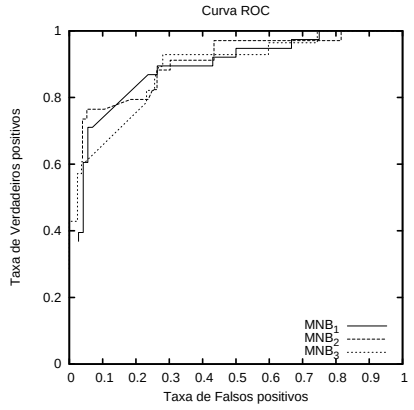
(b) banking LogReg



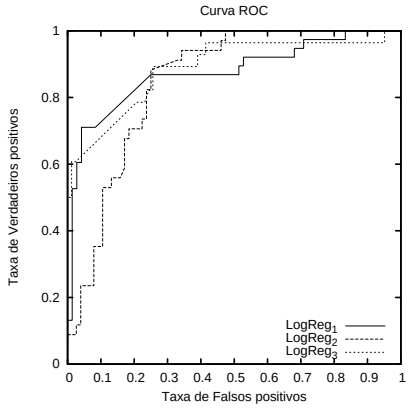
(c) games MNB



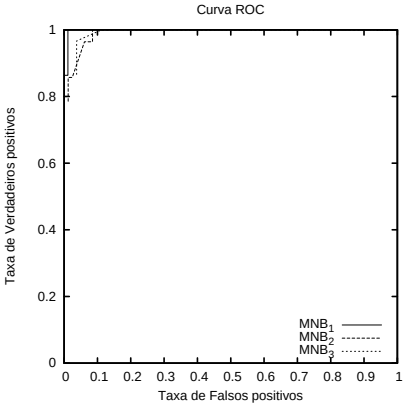
(d) games LogReg



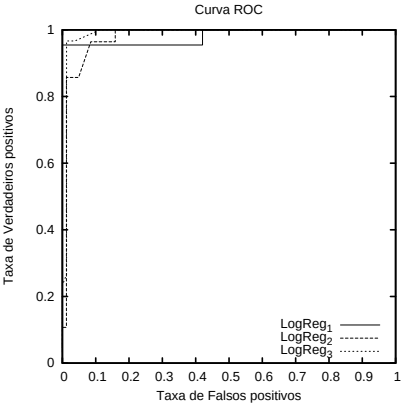
(e) medical MNB



(f) medical LogReg

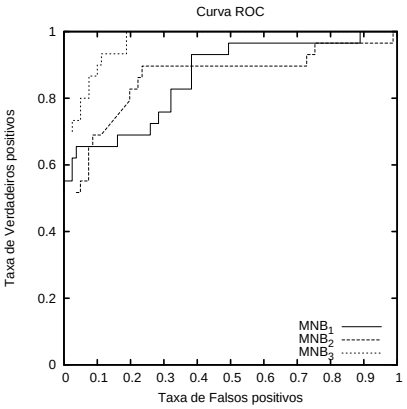


(g) porn MNB

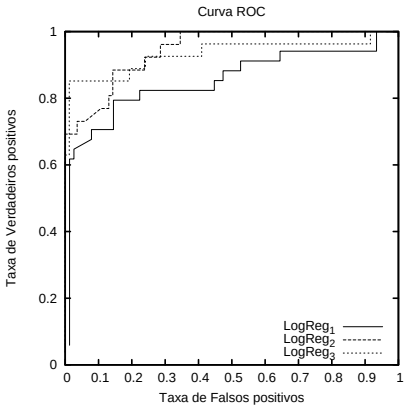


(h) porn LogReg

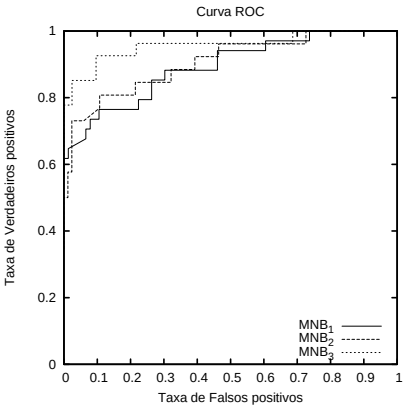
Figura A.3: Curvas ROC dos conjuntos de dados *celebrity*, *alcohol* e *news*



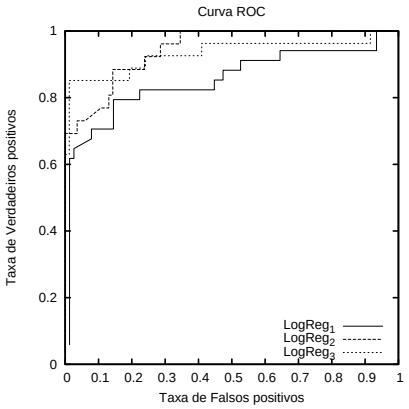
(a) celebrity MNB



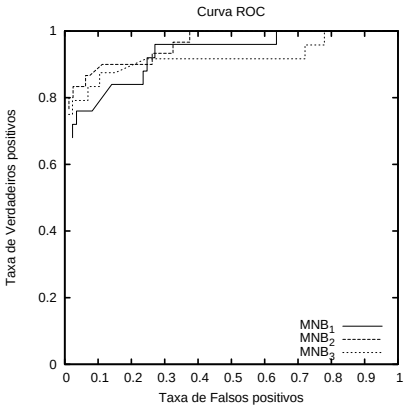
(b) celebrity LogReg



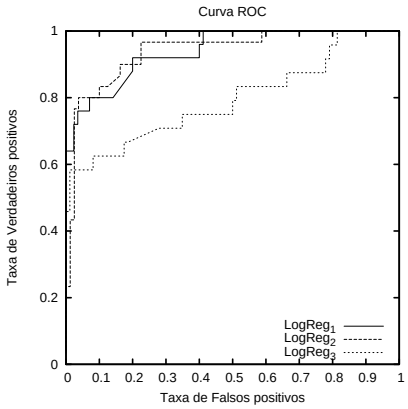
(c) alcohol MNB



(d) alcohol LogReg



(e) news MNB



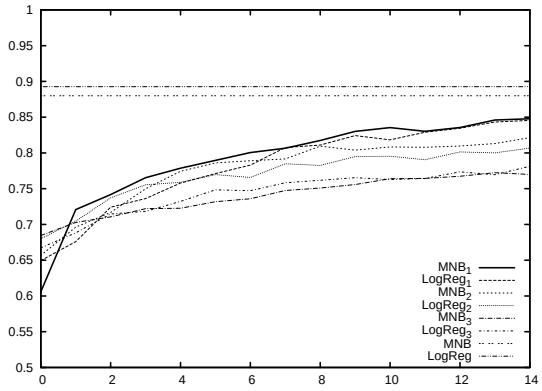
(f) news LogReg

Apêndice B

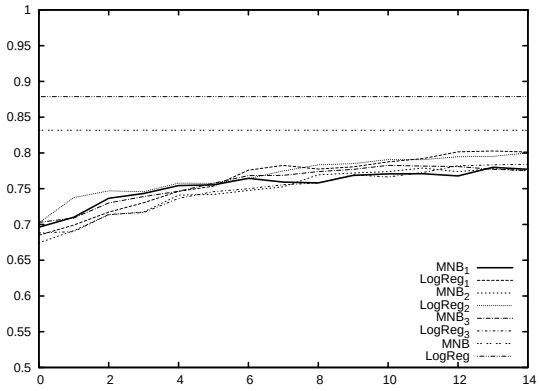
Gráficos AUC ROC vs iterações

A seguir estão os gráficos AUC ROC dos conjunto de dados restantes.

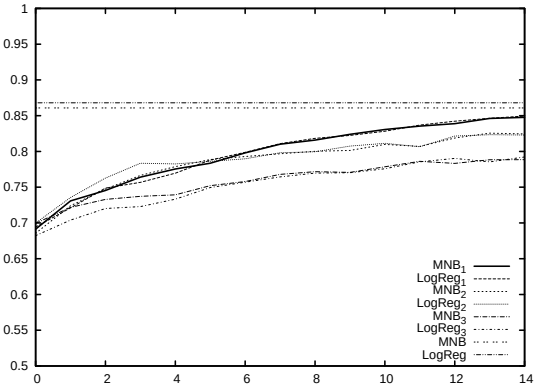
Figura B.1: Valores AUC ROC restantes dos conjuntos de dados *celebrity* e *drugs* ao longo das iterações



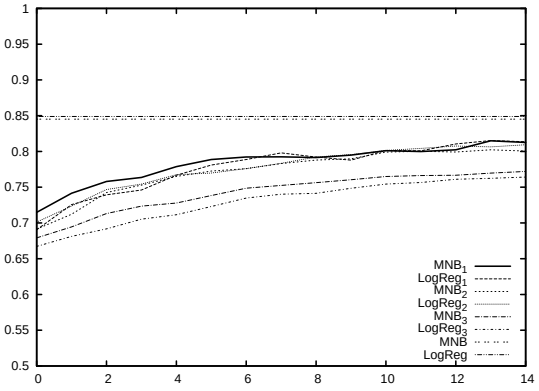
(a) celebrity sem seleção



(b) drugs sem seleção

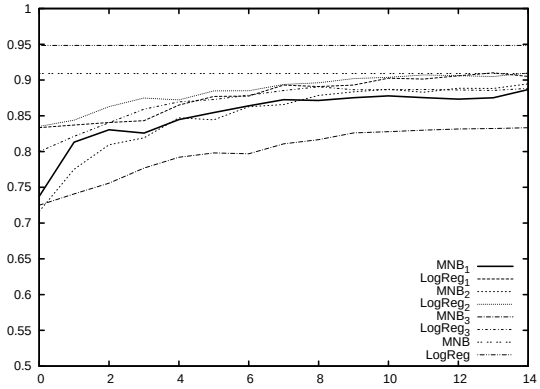


(c) celebrity com BNS

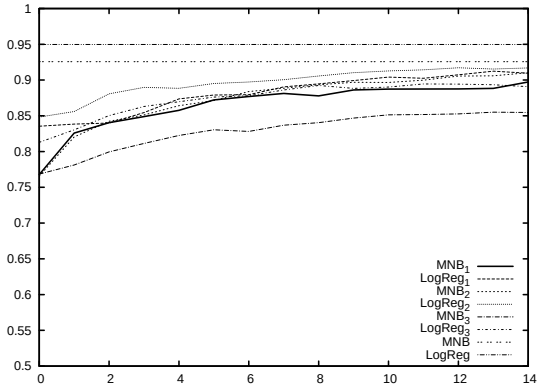


(d) drugs com BNS

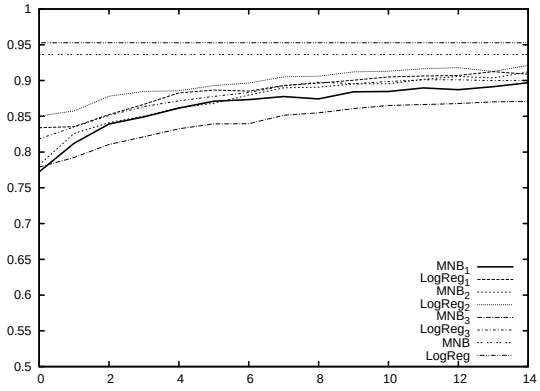
Figura B.2: Valores AUC ROC dos conjuntos de dados *banking*, *games* e *alcohol* ao longo das iterações



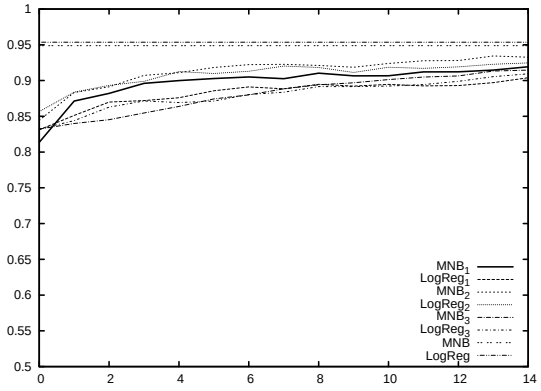
(a) alcohol sem seleção



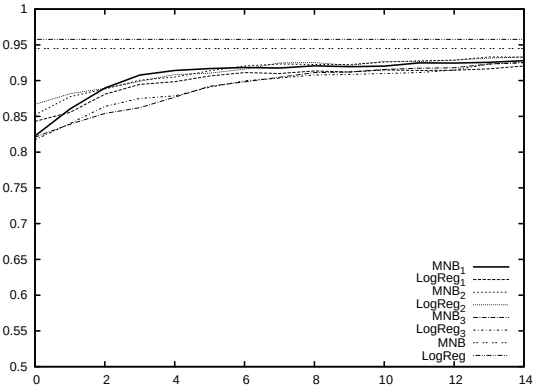
(b) alcohol com chi2



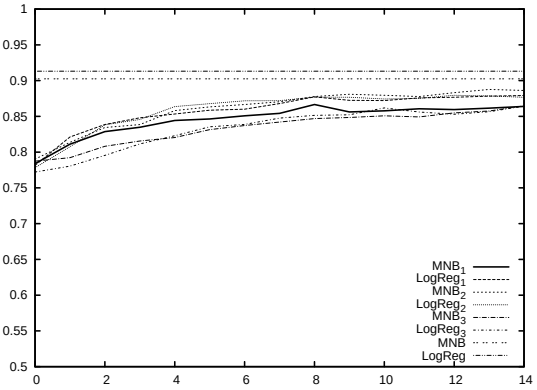
(c) alcohol com BNS



(d) banking com chi2

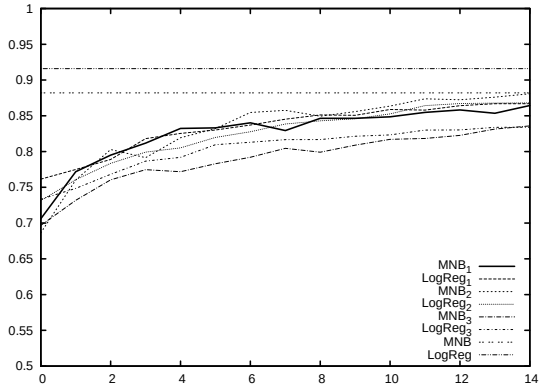


(e) banking com BNS

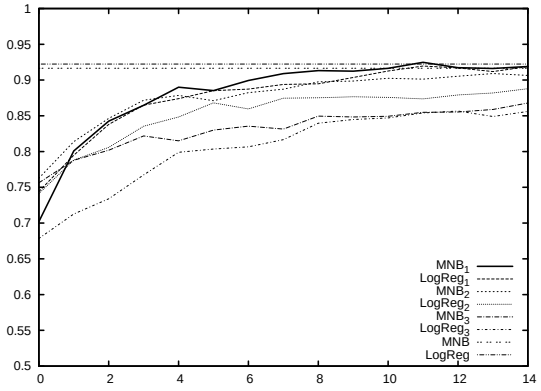


(f) games com BNS

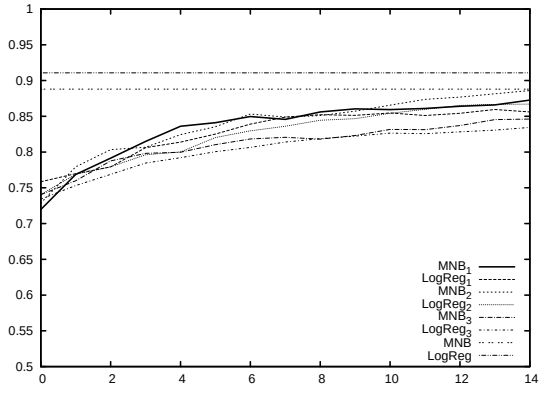
Figura B.3: Valores AUC ROC dos conjuntos de dados *medical* e *news* ao longo das iterações



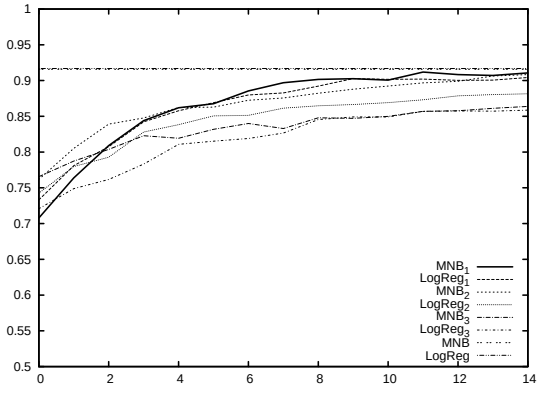
(a) medical sem seleção



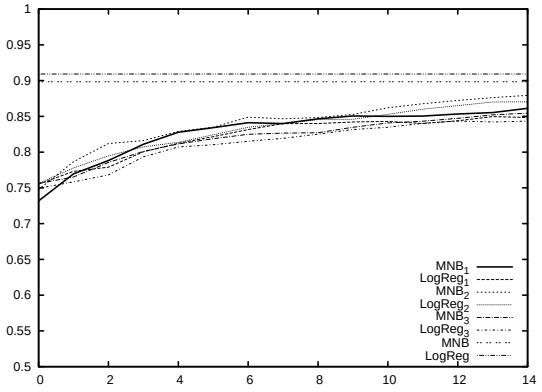
(b) news sem seleção



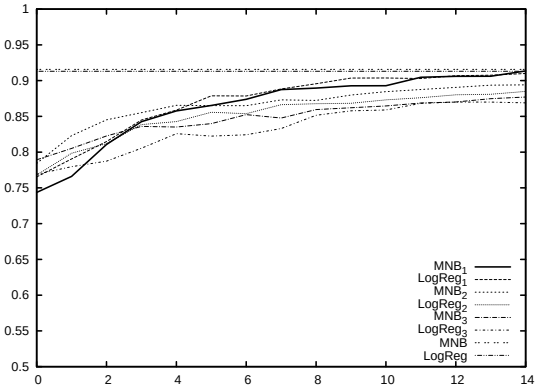
(c) medical com chi2



(d) news com chi2



(e) medical com BNS



(f) news com BNS

Apêndice C

Resultados dos Conjuntos de Dados

No que se segue são apresentados os resultados para todos os conjuntos de dados utilizados na avaliação experimental descrita na Seção 5.3. Os conjuntos de dados estão descritos na Seção 5.1.

Tabela C.1: Valores AUC ROC para classificação do conjunto de dados *porn* sem seleção

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,914	0,917	0,918	0,941	0,942	0,946	0,948	0,949	0,946	0,946	0,956	0,950	0,954	0,953	0,943
$LogReg_1$	0,941	0,953	0,954	0,956	0,957	0,962	0,963	0,967	0,967	0,970	0,970	0,970	0,968	0,970	0,970
MNB_2	0,911	0,917	0,926	0,938	0,949	0,945	0,951	0,953	0,958	0,957	0,958	0,954	0,958	0,958	0,959
$LogReg_2$	0,943	0,943	0,949	0,957	0,965	0,970	0,976	0,980	0,980	0,983	0,983	0,983	0,983	0,984	0,984
MNB_3	0,899	0,908	0,919	0,914	0,916	0,923	0,928	0,932	0,932	0,933	0,932	0,929	0,929	0,930	0,928
$LogReg_3$	0,923	0,931	0,946	0,947	0,946	0,948	0,947	0,954	0,958	0,963	0,965	0,973	0,972	0,973	0,973
MNB	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949
LogReg	0,990	0,990	0,990	0,990	0,990	0,990	0,990	0,990	0,990	0,990	0,990	0,990	0,990	0,990	0,990

Tabela C.2: Valores AUC ROC para classificação do conjunto de dados *porn* usando o seletor chi2

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,905	0,920	0,927	0,942	0,943	0,947	0,949	0,950	0,948	0,948	0,953	0,951	0,952	0,952	0,948
$LogReg_1$	0,945	0,952	0,956	0,963	0,964	0,968	0,970	0,972	0,972	0,974	0,974	0,975	0,974	0,975	0,975
MNB_2	0,899	0,919	0,926	0,935	0,935	0,937	0,941	0,945	0,949	0,947	0,947	0,946	0,948	0,948	0,957
$LogReg_2$	0,933	0,945	0,949	0,956	0,958	0,963	0,968	0,974	0,975	0,975	0,975	0,975	0,975	0,975	0,985
MNB_3	0,892	0,898	0,910	0,910	0,914	0,924	0,933	0,937	0,937	0,939	0,938	0,938	0,937	0,938	0,938
$LogReg_3$	0,920	0,923	0,939	0,942	0,942	0,954	0,958	0,961	0,967	0,971	0,972	0,977	0,977	0,977	0,978
MNB	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953
LogReg	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991

Tabela C.3: Valores AUC ROC para classificação do conjunto de dados *porn* usando o seletor BNS

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,905	0,926	0,931	0,947	0,946	0,950	0,953	0,954	0,952	0,953	0,957	0,955	0,954	0,956	0,953
$LogReg_1$	0,939	0,948	0,953	0,960	0,966	0,969	0,970	0,971	0,972	0,974	0,975	0,975	0,975	0,974	0,975
MNB_2	0,907	0,923	0,932	0,940	0,941	0,943	0,946	0,949	0,951	0,951	0,951	0,950	0,951	0,954	0,960
$LogReg_2$	0,930	0,945	0,953	0,958	0,960	0,965	0,970	0,972	0,975	0,975	0,976	0,976	0,975	0,979	0,985
MNB_3	0,897	0,906	0,918	0,920	0,926	0,932	0,938	0,940	0,942	0,943	0,944	0,945	0,945	0,945	0,944
$LogReg_3$	0,922	0,929	0,943	0,948	0,950	0,959	0,962	0,964	0,968	0,971	0,974	0,977	0,978	0,978	0,977
MNB	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953
LogReg	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991	0,991

Tabela C.4: Valores AUC ROC para classificação do conjunto de dados *drugs* sem seleção

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,696	0,710	0,736	0,743	0,754	0,755	0,764	0,759	0,758	0,768	0,770	0,770	0,767	0,779	0,777
$LogReg_1$	0,684	0,699	0,717	0,730	0,746	0,753	0,775	0,782	0,777	0,780	0,787	0,792	0,801	0,802	0,801
MNB_2	0,674	0,691	0,713	0,717	0,740	0,742	0,747	0,752	0,769	0,771	0,773	0,778	0,773	0,778	0,774
$LogReg_2$	0,702	0,737	0,746	0,745	0,757	0,757	0,764	0,774	0,783	0,785	0,790	0,790	0,794	0,795	0,800
MNB_3	0,702	0,709	0,730	0,738	0,746	0,757	0,768	0,768	0,773	0,776	0,782	0,781	0,780	0,777	0,775
$LogReg_3$	0,688	0,690	0,714	0,716	0,736	0,745	0,749	0,755	0,757	0,769	0,766	0,772	0,781	0,783	0,784
MNB	0,831	0,831	0,831	0,831	0,831	0,831	0,831	0,831	0,831	0,831	0,831	0,831	0,831	0,831	0,831
LogReg	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878

Tabela C.5: Valores AUC ROC para classificação do conjunto de dados *drugs* usando o seletor chi2

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,721	0,744	0,760	0,760	0,784	0,786	0,792	0,792	0,793	0,802	0,806	0,805	0,808	0,813	0,809
$LogReg_1$	0,697	0,719	0,730	0,742	0,762	0,771	0,783	0,794	0,797	0,793	0,800	0,804	0,814	0,812	0,810
MNB_2	0,689	0,713	0,737	0,747	0,758	0,762	0,769	0,773	0,778	0,775	0,786	0,786	0,780	0,786	0,786
$LogReg_2$	0,700	0,724	0,744	0,746	0,759	0,760	0,764	0,774	0,787	0,788	0,791	0,795	0,791	0,787	0,795
MNB_3	0,694	0,698	0,725	0,734	0,741	0,750	0,754	0,759	0,760	0,762	0,765	0,765	0,768	0,767	0,768
$LogReg_3$	0,684	0,684	0,700	0,712	0,715	0,727	0,736	0,742	0,748	0,757	0,758	0,763	0,771	0,769	0,767
MNB	0,805	0,805	0,805	0,805	0,805	0,805	0,805	0,805	0,805	0,805	0,805	0,805	0,805	0,805	0,805
LogReg	0,827	0,827	0,827	0,827	0,827	0,827	0,827	0,827	0,827	0,827	0,827	0,827	0,827	0,827	0,827

Tabela C.6: Valores AUC ROC para classificação do conjunto de dados *drugs* usando o seletor BNS

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,714	0,741	0,758	0,763	0,778	0,788	0,792	0,792	0,791	0,794	0,801	0,799	0,802	0,814	0,812
$LogReg_1$	0,690	0,725	0,739	0,746	0,766	0,781	0,788	0,797	0,792	0,787	0,801	0,800	0,810	0,815	0,813
MNB_2	0,691	0,712	0,742	0,752	0,765	0,772	0,775	0,783	0,787	0,789	0,799	0,800	0,799	0,802	0,800
$LogReg_2$	0,701	0,723	0,746	0,754	0,767	0,769	0,775	0,783	0,792	0,795	0,801	0,804	0,807	0,806	0,809
MNB_3	0,679	0,694	0,713	0,723	0,727	0,738	0,748	0,752	0,756	0,760	0,764	0,766	0,766	0,769	0,772
$LogReg_3$	0,667	0,681	0,691	0,705	0,711	0,722	0,734	0,740	0,741	0,748	0,754	0,756	0,761	0,762	0,764
MNB	0,845	0,845	0,845	0,845	0,845	0,845	0,845	0,845	0,845	0,845	0,845	0,845	0,845	0,845	0,845
LogReg	0,848	0,848	0,848	0,848	0,848	0,848	0,848	0,848	0,848	0,848	0,848	0,848	0,848	0,848	0,848

Tabela C.7: Valores AUC ROC para classificação do conjunto de dados *alcohol* sem seleção

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,737	0,813	0,830	0,825	0,844	0,854	0,863	0,872	0,871	0,875	0,877	0,875	0,873	0,875	0,886
$LogReg_1$	0,833	0,837	0,840	0,843	0,865	0,877	0,877	0,892	0,890	0,892	0,902	0,901	0,905	0,910	0,905
MNB_2	0,716	0,775	0,809	0,819	0,847	0,844	0,862	0,865	0,878	0,883	0,887	0,882	0,888	0,888	0,894
$LogReg_2$	0,835	0,843	0,862	0,874	0,872	0,884	0,885	0,893	0,896	0,902	0,903	0,907	0,906	0,904	0,909
MNB_3	0,724	0,740	0,755	0,776	0,792	0,798	0,796	0,810	0,816	0,826	0,827	0,829	0,831	0,832	0,833
$LogReg_3$	0,799	0,821	0,840	0,859	0,869	0,872	0,878	0,885	0,890	0,886	0,886	0,886	0,886	0,885	0,888
MNB	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909
LogReg	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948

Tabela C.8: Valores AUC ROC para classificação do conjunto de dados *alcohol* usando o seletor chi2

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,767	0,825	0,840	0,848	0,857	0,872	0,876	0,881	0,877	0,886	0,887	0,887	0,887	0,888	0,897
$LogReg_1$	0,835	0,838	0,839	0,854	0,873	0,879	0,879	0,890	0,894	0,899	0,904	0,902	0,907	0,912	0,909
MNB_2	0,765	0,820	0,842	0,851	0,864	0,871	0,883	0,888	0,893	0,896	0,896	0,899	0,905	0,905	0,910
$LogReg_2$	0,848	0,855	0,880	0,889	0,888	0,895	0,897	0,900	0,905	0,910	0,912	0,914	0,917	0,915	0,916
MNB_3	0,768	0,781	0,799	0,811	0,822	0,830	0,828	0,836	0,840	0,846	0,851	0,851	0,852	0,855	0,854
$LogReg_3$	0,812	0,830	0,850	0,863	0,869	0,876	0,878	0,886	0,892	0,888	0,890	0,894	0,894	0,893	0,890
MNB	0,925	0,925	0,925	0,925	0,925	0,925	0,925	0,925	0,925	0,925	0,925	0,925	0,925	0,925	0,925
LogReg	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949	0,949

Tabela C.9: Valores AUC ROC para classificação do conjunto de dados *alcohol* usando o seletor BNS

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,772	0,812	0,839	0,849	0,861	0,871	0,873	0,877	0,874	0,884	0,884	0,889	0,887	0,891	0,896
$LogReg_1$	0,834	0,835	0,852	0,866	0,882	0,886	0,885	0,893	0,896	0,900	0,904	0,906	0,907	0,912	0,908
MNB_2	0,782	0,826	0,841	0,850	0,861	0,868	0,879	0,889	0,890	0,895	0,895	0,901	0,905	0,903	0,912
$LogReg_2$	0,850	0,857	0,878	0,884	0,885	0,892	0,896	0,905	0,906	0,912	0,913	0,916	0,918	0,912	0,921
MNB_3	0,778	0,792	0,810	0,821	0,832	0,839	0,839	0,851	0,854	0,860	0,865	0,866	0,867	0,870	0,870
$LogReg_3$	0,818	0,835	0,851	0,863	0,871	0,877	0,883	0,892	0,897	0,895	0,898	0,901	0,901	0,899	0,900
MNB	0,936	0,936	0,936	0,936	0,936	0,936	0,936	0,936	0,936	0,936	0,936	0,936	0,936	0,936	0,936
LogReg	0,952	0,952	0,952	0,952	0,952	0,952	0,952	0,952	0,952	0,952	0,952	0,952	0,952	0,952	0,952

Tabela C.10: Valores AUC ROC para classificação do conjunto de dados *celebrity* sem seleção

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,605	0,720	0,741	0,765	0,778	0,789	0,800	0,806	0,817	0,830	0,835	0,830	0,835	0,846	0,847
$LogReg_1$	0,649	0,675	0,724	0,736	0,757	0,771	0,782	0,807	0,810	0,824	0,818	0,828	0,834	0,843	0,845
MNB_2	0,656	0,696	0,716	0,749	0,774	0,786	0,789	0,791	0,809	0,803	0,808	0,807	0,809	0,813	0,821
$LogReg_2$	0,680	0,704	0,737	0,755	0,758	0,769	0,765	0,784	0,782	0,794	0,795	0,790	0,801	0,800	0,806
MNB_3	0,684	0,702	0,710	0,722	0,722	0,731	0,735	0,747	0,750	0,755	0,763	0,764	0,767	0,772	0,769
$LogReg_3$	0,667	0,688	0,714	0,718	0,732	0,748	0,747	0,758	0,761	0,765	0,762	0,764	0,773	0,769	0,781
MNB	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879
LogReg	0,892	0,892	0,892	0,892	0,892	0,892	0,892	0,892	0,892	0,892	0,892	0,892	0,892	0,892	0,892

Tabela C.11: Valores AUC ROC para classificação do conjunto de dados *celebrity* usando o seletor chi2

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,650	0,711	0,725	0,752	0,762	0,770	0,788	0,802	0,812	0,824	0,830	0,832	0,835	0,842	0,842
$LogReg_1$	0,660	0,685	0,722	0,734	0,749	0,766	0,782	0,799	0,809	0,816	0,817	0,824	0,830	0,831	0,834
MNB_2	0,670	0,712	0,737	0,754	0,771	0,784	0,789	0,795	0,805	0,802	0,812	0,810	0,820	0,828	0,826
$LogReg_2$	0,681	0,718	0,746	0,765	0,772	0,777	0,774	0,791	0,795	0,802	0,807	0,800	0,813	0,819	0,818
MNB_3	0,690	0,708	0,719	0,726	0,726	0,740	0,744	0,750	0,754	0,756	0,762	0,765	0,768	0,770	0,772
$LogReg_3$	0,656	0,671	0,698	0,704	0,713	0,735	0,739	0,747	0,753	0,749	0,754	0,756	0,768	0,760	0,768
MNB	0,840	0,840	0,840	0,840	0,840	0,840	0,840	0,840	0,840	0,840	0,840	0,840	0,840	0,840	0,840
LogReg	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860

Tabela C.12: Valores AUC ROC para classificação do conjunto de dados *celebrity* usando o seletor BNS

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,691	0,730	0,745	0,764	0,775	0,783	0,797	0,810	0,815	0,823	0,830	0,835	0,838	0,846	0,847
$LogReg_1$	0,694	0,721	0,748	0,756	0,769	0,787	0,798	0,810	0,818	0,822	0,828	0,836	0,842	0,846	0,849
MNB_2	0,685	0,724	0,747	0,766	0,778	0,788	0,792	0,796	0,800	0,801	0,809	0,807	0,818	0,825	0,824
$LogReg_2$	0,699	0,735	0,762	0,783	0,782	0,786	0,789	0,798	0,799	0,807	0,811	0,806	0,821	0,823	0,822
MNB_3	0,699	0,721	0,732	0,737	0,739	0,752	0,757	0,767	0,771	0,770	0,778	0,786	0,783	0,788	0,788
$LogReg_3$	0,682	0,703	0,720	0,722	0,733	0,749	0,756	0,764	0,769	0,770	0,775	0,785	0,790	0,785	0,792
MNB	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860	0,860
LogReg	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868

Tabela C.13: Valores AUC ROC para classificação do conjunto de dados *banking* sem seleção

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,779	0,857	0,871	0,896	0,901	0,904	0,899	0,900	0,910	0,902	0,899	0,903	0,907	0,907	0,915
$LogReg_1$	0,818	0,833	0,856	0,859	0,865	0,872	0,879	0,886	0,895	0,896	0,893	0,893	0,896	0,898	0,909
MNB_2	0,856	0,888	0,904	0,922	0,922	0,926	0,933	0,933	0,929	0,927	0,935	0,938	0,941	0,948	0,944
$LogReg_2$	0,856	0,882	0,896	0,905	0,914	0,919	0,913	0,918	0,918	0,908	0,917	0,920	0,922	0,928	0,931
MNB_3	0,828	0,850	0,855	0,864	0,876	0,886	0,885	0,893	0,896	0,900	0,903	0,904	0,905	0,908	0,908
$LogReg_3$	0,852	0,860	0,876	0,885	0,880	0,885	0,890	0,892	0,895	0,895	0,898	0,895	0,897	0,901	0,903
MNB	0,923	0,923	0,923	0,923	0,923	0,923	0,923	0,923	0,923	0,923	0,923	0,923	0,923	0,923	0,923
LogReg	0,938	0,938	0,938	0,938	0,938	0,938	0,938	0,938	0,938	0,938	0,938	0,938	0,938	0,938	0,938

Tabela C.14: Valores AUC ROC para classificação do conjunto de dados *banking* usando o seletor chi2

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,813	0,871	0,882	0,896	0,899	0,902	0,904	0,902	0,910	0,906	0,906	0,912	0,912	0,914	0,919
$LogReg_1$	0,831	0,851	0,869	0,871	0,876	0,885	0,891	0,888	0,894	0,892	0,894	0,892	0,893	0,897	0,903
MNB_2	0,844	0,883	0,891	0,907	0,911	0,918	0,922	0,922	0,921	0,918	0,923	0,927	0,928	0,934	0,932
$LogReg_2$	0,856	0,883	0,893	0,898	0,912	0,909	0,912	0,920	0,918	0,911	0,918	0,917	0,919	0,922	0,924
MNB_3	0,832	0,839	0,845	0,854	0,863	0,874	0,879	0,888	0,894	0,896	0,901	0,904	0,906	0,913	0,914
$LogReg_3$	0,832	0,843	0,862	0,871	0,869	0,871	0,880	0,883	0,891	0,891	0,892	0,894	0,898	0,905	0,909
MNB	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948	0,948
LogReg	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953	0,953

Tabela C.15: Valores AUC ROC para classificação do conjunto de dados *banking* usando o seletor BNS

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,822	0,860	0,889	0,907	0,914	0,916	0,918	0,917	0,920	0,919	0,920	0,924	0,924	0,925	0,927
$LogReg_1$	0,843	0,855	0,880	0,894	0,898	0,906	0,911	0,910	0,913	0,912	0,914	0,914	0,914	0,916	0,920
MNB_2	0,852	0,877	0,888	0,900	0,905	0,913	0,920	0,923	0,922	0,922	0,925	0,927	0,928	0,933	0,932
$LogReg_2$	0,866	0,881	0,889	0,899	0,908	0,910	0,916	0,924	0,925	0,921	0,926	0,926	0,928	0,931	0,933
MNB_3	0,820	0,838	0,854	0,862	0,876	0,892	0,898	0,904	0,911	0,912	0,915	0,917	0,917	0,923	0,925
$LogReg_3$	0,817	0,839	0,863	0,875	0,878	0,890	0,899	0,903	0,907	0,908	0,909	0,911	0,914	0,923	0,925
MNB	0,944	0,944	0,944	0,944	0,944	0,944	0,944	0,944	0,944	0,944	0,944	0,944	0,944	0,944	0,944
LogReg	0,957	0,957	0,957	0,957	0,957	0,957	0,957	0,957	0,957	0,957	0,957	0,957	0,957	0,957	0,957

Tabela C.16: Valores AUC ROC para classificação do conjunto de dados *games* sem seleção

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,772	0,816	0,818	0,810	0,838	0,849	0,853	0,864	0,869	0,864	0,869	0,867	0,870	0,876	0,881
$LogReg_1$	0,801	0,825	0,840	0,851	0,862	0,871	0,878	0,881	0,882	0,889	0,892	0,898	0,904	0,903	0,902
MNB_2	0,780	0,790	0,820	0,826	0,853	0,860	0,867	0,862	0,871	0,868	0,874	0,874	0,887	0,893	0,892
$LogReg_2$	0,776	0,792	0,809	0,827	0,842	0,847	0,850	0,859	0,860	0,860	0,861	0,865	0,875	0,875	0,876
MNB_3	0,773	0,792	0,791	0,808	0,811	0,818	0,823	0,825	0,835	0,839	0,840	0,842	0,846	0,851	0,857
$LogReg_3$	0,777	0,780	0,793	0,819	0,832	0,837	0,839	0,836	0,847	0,843	0,848	0,848	0,845	0,846	0,847
MNB	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868	0,868
LogReg	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910

Tabela C.17: Valores AUC ROC para classificação do conjunto de dados *games* usando o seletor chi2

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,785	0,811	0,823	0,824	0,839	0,845	0,851	0,856	0,863	0,858	0,862	0,866	0,865	0,868	0,872
$LogReg_1$	0,785	0,812	0,826	0,840	0,849	0,855	0,858	0,866	0,868	0,874	0,875	0,880	0,881	0,881	0,884
MNB_2	0,791	0,808	0,823	0,833	0,856	0,866	0,868	0,869	0,874	0,879	0,886	0,883	0,892	0,895	0,893
$LogReg_2$	0,775	0,796	0,823	0,837	0,854	0,857	0,862	0,868	0,870	0,870	0,868	0,870	0,877	0,877	0,878
MNB_3	0,790	0,799	0,805	0,811	0,813	0,823	0,829	0,832	0,840	0,842	0,843	0,843	0,849	0,853	0,856
$LogReg_3$	0,776	0,782	0,791	0,808	0,817	0,823	0,826	0,833	0,844	0,842	0,851	0,848	0,845	0,850	0,849
MNB	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879	0,879
LogReg	0,897	0,897	0,897	0,897	0,897	0,897	0,897	0,897	0,897	0,897	0,897	0,897	0,897	0,897	0,897

Tabela C.18: Valores AUC ROC para classificação do conjunto de dados *games* usando o seletor BNS

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,783	0,811	0,828	0,834	0,844	0,846	0,850	0,854	0,866	0,856	0,857	0,860	0,859	0,861	0,863
$LogReg_1$	0,781	0,821	0,838	0,848	0,853	0,858	0,860	0,867	0,877	0,872	0,872	0,875	0,876	0,878	0,879
MNB_2	0,790	0,813	0,834	0,838	0,858	0,863	0,866	0,870	0,877	0,880	0,879	0,877	0,883	0,887	0,886
$LogReg_2$	0,778	0,807	0,838	0,845	0,863	0,867	0,871	0,872	0,876	0,876	0,874	0,875	0,879	0,878	0,876
MNB_3	0,787	0,792	0,808	0,815	0,820	0,831	0,837	0,842	0,846	0,848	0,850	0,849	0,854	0,857	0,864
$LogReg_3$	0,772	0,780	0,795	0,811	0,822	0,835	0,838	0,847	0,851	0,852	0,861	0,856	0,852	0,856	0,864
MNB	0,902	0,902	0,902	0,902	0,902	0,902	0,902	0,902	0,902	0,902	0,902	0,902	0,902	0,902	0,902
LogReg	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913

Tabela C.19: Valores AUC ROC para classificação do conjunto de dados *medical* sem seleção

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,706	0,771	0,794	0,811	0,832	0,833	0,840	0,829	0,846	0,846	0,848	0,854	0,858	0,853	0,864
$LogReg_1$	0,761	0,774	0,788	0,818	0,825	0,830	0,837	0,845	0,851	0,850	0,858	0,857	0,864	0,866	0,866
MNB_2	0,686	0,760	0,802	0,791	0,819	0,832	0,854	0,857	0,849	0,855	0,863	0,873	0,872	0,876	0,880
$LogReg_2$	0,731	0,760	0,783	0,799	0,805	0,819	0,827	0,838	0,842	0,845	0,852	0,864	0,867	0,867	0,867
MNB_3	0,696	0,731	0,760	0,774	0,771	0,782	0,792	0,804	0,799	0,808	0,817	0,818	0,822	0,831	0,835
$LogReg_3$	0,733	0,748	0,768	0,786	0,791	0,809	0,813	0,816	0,816	0,821	0,823	0,830	0,830	0,833	0,833
MNB	0,882	0,882	0,882	0,882	0,882	0,882	0,882	0,882	0,882	0,882	0,882	0,882	0,882	0,882	0,882
LogReg	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915

Tabela C.20: Valores AUC ROC para classificação do conjunto de dados *medical* usando o seletor chi2

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,719	0,768	0,791	0,814	0,836	0,841	0,849	0,845	0,856	0,860	0,859	0,860	0,863	0,865	0,872
$LogReg_1$	0,758	0,769	0,779	0,806	0,813	0,825	0,839	0,848	0,852	0,851	0,854	0,851	0,854	0,859	0,856
MNB_2	0,728	0,779	0,803	0,806	0,824	0,835	0,853	0,848	0,850	0,857	0,865	0,873	0,876	0,881	0,885
$LogReg_2$	0,740	0,768	0,779	0,796	0,800	0,820	0,829	0,836	0,844	0,846	0,854	0,859	0,865	0,866	0,867
MNB_3	0,740	0,759	0,787	0,798	0,799	0,810	0,818	0,820	0,818	0,823	0,831	0,831	0,837	0,845	0,846
$LogReg_3$	0,734	0,753	0,768	0,784	0,792	0,800	0,806	0,813	0,818	0,822	0,826	0,825	0,828	0,830	0,834
MNB	0,887	0,887	0,887	0,887	0,887	0,887	0,887	0,887	0,887	0,887	0,887	0,887	0,887	0,887	0,887
LogReg	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910	0,910

Tabela C.21: Valores AUC ROC para classificação do conjunto de dados *medical* usando o seletor BNS

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,731	0,769	0,788	0,810	0,827	0,834	0,841	0,840	0,846	0,850	0,850	0,850	0,853	0,855	0,861
$LogReg_1$	0,755	0,773	0,779	0,800	0,812	0,821	0,831	0,839	0,839	0,842	0,842	0,840	0,843	0,849	0,848
MNB_2	0,747	0,786	0,811	0,816	0,829	0,834	0,848	0,846	0,848	0,852	0,862	0,867	0,872	0,876	0,879
$LogReg_2$	0,756	0,777	0,794	0,807	0,813	0,824	0,834	0,840	0,845	0,845	0,852	0,860	0,865	0,870	0,870
MNB_3	0,756	0,765	0,785	0,800	0,811	0,818	0,824	0,826	0,827	0,834	0,840	0,843	0,847	0,852	0,854
$LogReg_3$	0,749	0,758	0,768	0,793	0,807	0,810	0,815	0,819	0,825	0,831	0,834	0,840	0,843	0,842	0,843
MNB	0,898	0,898	0,898	0,898	0,898	0,898	0,898	0,898	0,898	0,898	0,898	0,898	0,898	0,898	0,898
LogReg	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909	0,909

Tabela C.22: Valores AUC ROC para classificação do conjunto de dados *news* sem seleção

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,701	0,800	0,842	0,864	0,889	0,885	0,899	0,909	0,913	0,912	0,916	0,924	0,917	0,916	0,918
$LogReg_1$	0,744	0,794	0,837	0,864	0,873	0,885	0,887	0,893	0,894	0,903	0,912	0,919	0,916	0,911	0,918
MNB_2	0,762	0,813	0,846	0,871	0,878	0,871	0,882	0,887	0,897	0,898	0,902	0,901	0,905	0,909	0,906
$LogReg_2$	0,741	0,788	0,805	0,835	0,848	0,868	0,859	0,874	0,875	0,876	0,875	0,873	0,879	0,881	0,887
MNB_3	0,756	0,787	0,801	0,821	0,814	0,829	0,835	0,831	0,849	0,848	0,849	0,854	0,855	0,858	0,867
$LogReg_3$	0,678	0,712	0,733	0,767	0,799	0,803	0,806	0,816	0,839	0,844	0,847	0,853	0,856	0,849	0,856
MNB	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916
LogReg	0,922	0,922	0,922	0,922	0,922	0,922	0,922	0,922	0,922	0,922	0,922	0,922	0,922	0,922	0,922

Tabela C.23: Valores AUC ROC para classificação do conjunto de dados *news* usando o seletor chi2

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,707	0,763	0,809	0,843	0,861	0,867	0,885	0,897	0,901	0,902	0,900	0,911	0,908	0,907	0,910
$LogReg_1$	0,733	0,781	0,807	0,842	0,857	0,868	0,879	0,882	0,892	0,902	0,901	0,902	0,900	0,900	0,904
MNB_2	0,761	0,805	0,839	0,847	0,862	0,862	0,872	0,875	0,882	0,887	0,892	0,896	0,898	0,906	0,908
$LogReg_2$	0,742	0,779	0,792	0,828	0,838	0,850	0,851	0,861	0,864	0,866	0,869	0,873	0,878	0,880	0,881
MNB_3	0,765	0,787	0,803	0,822	0,819	0,831	0,839	0,832	0,847	0,847	0,849	0,856	0,857	0,861	0,863
$LogReg_3$	0,720	0,748	0,761	0,783	0,810	0,815	0,818	0,826	0,845	0,849	0,849	0,856	0,857	0,857	0,858
MNB	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915
LogReg	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916	0,916

Tabela C.24: Valores AUC ROC para classificação do conjunto de dados *news* usando o seletor BNS

passos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MNB_1	0,743	0,766	0,810	0,842	0,857	0,865	0,873	0,887	0,889	0,892	0,892	0,904	0,906	0,906	0,913
$LogReg_1$	0,765	0,790	0,814	0,844	0,858	0,878	0,878	0,888	0,895	0,903	0,903	0,902	0,906	0,907	0,909
MNB_2	0,784	0,823	0,845	0,855	0,865	0,865	0,864	0,873	0,872	0,879	0,884	0,887	0,890	0,893	0,894
$LogReg_2$	0,767	0,798	0,811	0,838	0,842	0,855	0,853	0,866	0,867	0,868	0,873	0,876	0,880	0,880	0,884
MNB_3	0,789	0,805	0,822	0,836	0,835	0,839	0,852	0,847	0,859	0,862	0,864	0,868	0,870	0,874	0,877
$LogReg_3$	0,769	0,779	0,787	0,805	0,825	0,822	0,824	0,832	0,851	0,857	0,858	0,868	0,869	0,870	0,868
MNB	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915	0,915
LogReg	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913	0,913

Referências Bibliográficas

Alecrim, E. (2004). Firewall: conceitos e tipos. `http://www.infowester.com/firewall.php`. Acesso em 12 Out 2012.

Alexa (2012). Top sites. `http://www.alexa.com`. Acesso em 15 Jul 2012.

Allfro (2013). Python micro interceptor proxy. `https://github.com/allfro/pymiproxy`. Acesso em 15 Jul 2012.

Blum, A. e Mitchell, T. (1998). Combining labeled and unlabeled data with Co-Training. *COLT'98: Proceedings of the 11th Annual Conference on Computational Learning Theory*, 11:92–100.

Braga, I. A. (2010). Aprendizado semissupervisionado multidescrição em classificação de textos. Tese de Mestrado, ICMC–USP.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., e Hullender, G. (2005). Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, páginas 89–96, New York, NY, USA. ACM.

Cayton, L. (2005). Algorithms for manifold learning. Relatório técnico, UCSD tech report CS2008-0923.

Chapelle, O., Schölkopf, B., e Zien, A. (2006). *Semi-Supervised Learning*. The MIT Press.

Chen, J., Huang, H., Tian, S., e Qu, Y. (2009). Feature selection for text classification with naive bayes. *Expert Systems with Applications*, 36(3, Part 1):5432 – 5435.

Cohen, E., Krishnamurthy, B., e Rexford, J. (1998). Improving end-to-end performance of the web using server volumes and proxy filters. *SIGCOMM Comput. Commun. Rev.*, 28:241–253.

Collins, M. e Singer, Y. (1999). Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, páginas 100–110.

Cortez, P. e Morais, A. (2007). A Data Mining Approach to Predict Forest Fires using Meteorological Data. In et al., J. N., editor, *New Trends in Artificial Intelligence,*

13th EPIA 2007 - Portuguese Conference on Artificial Intelligence, páginas 512–523, Guimarães, Portugal. APPIA.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27:861–874.

Fawcett, T. e Provost, F. (1997). Adaptive fraud detection. *Data Min. Knowl. Discov.*, 1:291–316.

Ferri, C., Flach, P., e Hernández-Orallo, J. (2002). Learning decision trees using the area under the ROC curve. In *Proceedings of the 19th International Conference on Machine Learning*, páginas 139–146. Morgan Kaufmann.

Fisher, L. e Ness, J. W. V. (1971). Admissible clustering procedures. *Biometrika*.

Flach, P. A. e Wu, S. (2005). Repairing concavities in ROC curves. In *Proceedings of the 19th international joint conference on Artificial intelligence*, páginas 702–707, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305.

Globo.com (2010). Internet ganha primeiro censo de sites em 25 anos. `http://g1.globo.com/Noticias/Tecnologia/0,,MUL150660-6174,00.html`. Acesso em 11 Fev 2012.

Guyon, I. e Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182.

Haffari, G. e Sarkar, A. (2007). Analysis of semi-supervised learning with the yarowsky algorithm. Technical Report TR 2007-07, Simon Fraser University, 8888 University Drive, Burnaby, B.C. Canada V5A 1S6.

ITU (2013). The world in 2013 - ict facts and figures. `http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013.pdf`. Acesso em 2 Jun 2013.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, páginas 200–209, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kamal Nigam, A. M. (1998). A comparison of event models for naive bayes text classification. *Association for the Advancement of Artificial Intelligence*, páginas Relatório Técnico WS–98–05.

Lallich, S., Muhlenbach, F., e Zighed, D. (2002). Improving classification by removing or relabeling mislabeled instances. In *Foundations of Intelligent Systems*, volume 2366, páginas 5–15. Springer Berlin / Heidelberg.

Lewis, D. D. e Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In Cohen, W. W. e Hirsh, H., editors, *Proceedings of ICML-94, 11th International Conference on Machine Learning*, páginas 148–156, New Brunswick, US. Morgan Kaufmann Publishers, San Francisco, US.

Li, Y., Guan, C., Li, H., e Chin, Z. (2008). A self-training semi-supervised SVM algorithm and its application in an eeg-based brain computer interface speller system. In *Pattern Recognition Letters 29*, páginas 1285–1294.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Maeireizo, B., Litman, D., e Hwa, R. (2004). Co-training for predicting emotions with spoken dialogue data. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, Morristown, NJ, USA. Association for Computational Linguistics.

Martins, C. A. (2003). Uma abordagem para pré-processamento de dados textuais em algoritmos de aprendizado. Tese de Doutorado, USP.

Matsubara, E. T. (2004). O algoritmo de aprendizado semi-supervisionado co-training e sua aplicação na rotulação de documentos. Tese de Mestrado, ICMC–USP.

Matsubara, E. T. (2008). Relações entre ranking, análise roc e calibração em aprendizado de máquina. Tese de Doutorado, ICMC–USP.

Matsubara, E. T. e Monard, M. C. (2003). Pretext: Uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words.

Merz, C., St. Clair, D., e Bond, W. (1992). Semi-supervised adaptive resonance theory (smart2). In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume III, pagina 851 a 856.

Michaelis (2012). *Moderno Dicionário da Língua Portuguesa*. Companhia Melhoramentos, São Paulo.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Netcraft (2013). February 2013 web server survey. `http://news.netcraft.com/archives/2013/02/01/february-2013-web-server-survey.html`. Acesso em 1 Jun 2013.

NIC.br (2012). Painel IBOPE/NetRatings. `http://www.cetic.br/usuarios/ibope/`. Acesso em 2 Jun 2013.

Oxford (2008). *Oxford English Dictionary A*. Oxford UK.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., e Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Provost, F. e Domingos, P. (2003). Tree induction for probability-based ranking. *Mach. Learn.*, 52:199–215.

Provost, F. e Fawcett, T. (1998). Robust classification systems for imprecise environments. In *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, AAAI '98/IAAI '98, páginas 706–713, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Rosenberg, C., Hebert, M., e Schneiderman, H. (2005). Semi-supervised self-training of object detection models. In *Seventh IEEE Workshop on Applications of Computer Vision*, páginas 29–36.

Russell, S. J. e Norvig, P. (2002). *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall.

Salton, G. e Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, páginas 513–523.

Sebastiani, F. e Ricerche, C. N. D. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47.

Settles, B. (2009). Active learning literature survey. Relatório Técnico 1648.

Shi, L., Ma, X., Xi, L., Duan, Q., e Zhao, J. (2011). Rough set and ensemble learning based semi-supervised algorithm for text classification. *Expert Systems with Applications*, 38(5):6300 – 6306.

Spackman, K. A. (1989). Signal detection theory: valuable tools for evaluating inductive learning. In *Proceedings of the sixth international workshop on Machine learning*, páginas 160–163, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Stats, I. W. (2013). Top 20 internet countries. `http://www.internetworldstats.com/top20.htm`. Acesso em 2 Jun 2013.

Sun, Z., Ye, Y., Zhang, X., Huang, Z., Chen, S., e Liu, Z. (2012). Batch-mode active learning with semi-supervised cluster tree for text classification. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12, páginas 388–395, Washington, DC, USA. IEEE Computer Society.

Tolman, E. C. e Honzik, C. (1930). Introduction and removal of reward, and maze performance in rats. *University of California, Publications in Psychology*, 4:257–275.

van Rossum, G. e de Boer, J. (1991). Linking a stub generator (AIL) to a prototyping language (Python). páginas 229–247.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Wajeed, M. e Adilakshmi, T. (2011). Semi-supervised text classification using enhanced KNN algorithm. In *Information and Communication Technologies (WICT), 2011 World Congress on*, páginas 138–142.

Weston, J. (2008). Large scale semi-supervised learning. In *Proceedings of NATO Advanced Study Institute on Mining Massive Data Sets for Security*, páginas 62–75. IOS Press.

Witten, I. H. e Frank, E. (2011). *Data mining: practical machine learning tools and techniques*, volume 31. Elsevier, New York, NY, USA.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. *ACL' 95: Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 1:189–196.

Zeng, H. e Cheung, Y.-M. (2009). A new feature selection method for gaussian mixture clustering. *Pattern Recogn.*, 42(2):243–250.

Zhu, X. e Goldberg, A. B. (2009). *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Zipf, G. (1949). Human behaviour and the principle of least-effort. Addison-Wesley, Cambridge, MA.

Glossário

bag-of-words é um modelo de representação simplificada usada em processamento de linguagem natural e visão computacional para processamento e obtenção de informações. Nesse modelo, um texto (como uma sentença ou documento) é representado como uma coleção de palavras não ordenadas. 38

proxy é um servidor intermediário que atende a requisições repassando os dados do cliente. Um servidor proxy pode, opcionalmente, alterar a requisição do cliente ou a resposta do servidor e, algumas vezes, pode disponibilizar esse recurso mesmo sem se conectar ao servidor especificado. 26, 37

stoplist é uma lista de *stopword*s. 31, 37

stopword é uma palavra que pode ser considerada irrelevante para o conjunto de dados. Exemplos: as, e, os, de, para, com, sem. 31, 91

AUC (*Area under a ROC curve*, ou área abaixo da curva ROC) é o cálculo da área abaixo da curva ROC. Como o AUC é uma porção de área de unidade quadrada, seu valor estará sempre entre 0 e 1,0. 22

DNS significa *Domain Name System* (Sistema de Nomes de Domínios), é um sistema de gerenciamento de nomes hierárquico e distribuído. 27

FTP significa *File Transfer Protocol* (Protocolo de Transferência de Arquivos), é uma forma bastante rápida e versátil de transferir arquivos, sendo uma das mais usadas na Internet. 27

HTML significa *HyperText Markup Language* (Linguagem de Marcação de Hipertexto), é uma linguagem de marcação utilizada para produzir páginas na Web. Documentos HTML podem ser interpretados por navegadores. 36

HTTP significa *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto), é um protocolo de comunicação utilizado para sistemas de informação de hipermídia distribuídos e colaborativos. Seu uso para a obtenção de recursos interligados levou ao estabelecimento da *World Wide Web* (WWW). 27, 92

ROC (*Receiver Operating Characteristics*) ou curva ROC, é uma representação gráfica que ilustra a performance de um sistema classificador binário e como o seu limiar de discriminação é variado. 20, 22, 91

sítio também chamado de site web ou sítio web, vem do inglês *website* e é um conjunto de páginas web, isto é, de hipertextos acessíveis geralmente pelo protocolo HTTP na Internet. O conjunto de todos os sítios públicos existentes compõe a *World Wide Web* (WWW). As páginas em um sítio são organizadas a partir de um endereço URL básico, onde fica a página principal. 1, 35

URL significa *Uniform Resource Locator*, (Localizador-Padrão de Recursos), é o endereço de um recurso (um arquivo, uma impressora etc.) disponível em uma rede; seja a Internet, ou uma rede corporativa (uma intranet). Uma URL tem a seguinte estrutura: protocolo://máquina/caminho/recurso. 27, 38, 92

WWW significa *World Wide Web* (Rede de alcance mundial, também conhecida como Web), é um sistema de documentos em hipermídia que são interligados e executados na Internet. 91, 92