MARCELO HENRIQUE CAVALCANTI JUCÁ

UMA ABORDAGEM SUPERVISIONADA SOBRE A CLASSIFICAÇÃO DA "SACCHAROMYCES CEREVISIAE"

Universidade Federal de Pernambuco

Recife, Setembro/2004.

MARCELO HENRIQUE CAVALCANTI JUCÁ

UMA ABORDAGEM SUPERVISIONADA SOBRE A CLASSIFICAÇÃO DA "SACCHAROMYCES CEREVISIAE"

Monografia apresentada à Universidade Federal de Pernambuco como exigência para a obtenção do grau de bacharel em Ciências da Computação, sob orientação do professor Francisco de Assis Tenório de Carvalho e co-orientação do professor Valdir de Queiroz Balbino.

Universidade Federal de Pernambuco

Recife, Setembro/2004.

Marcelo Henrique Cavalcanti Jucá

UMA ABORDAGEM SUPERVISIONADA SOBRE A CLASSIFICAÇÃO DA "SACCHAROMYCES CEREVISIAE"

Aprovado: ___/___/___

Nota: ___(_____)

Banca Examinadora:

_____
Prof.(a)

_____
Prof.(a)

Universidade Federal de Pernambuco

Recife, Setembro/2004.

Jucá, Marcelo Henrique Cavalcanti.

Uma Abordagem Supervisionada sobre a Classificação da "Saccharomyces cerevisiae"/

Marcelo Henrique Cavalcanti Jucá.- Recife: Edição do autor, 2004.

1.Informática. I Título.

CDD...

AGRADECIMENTO

Agradeço a Deus, que possibilitou a concretização de um sonho, à minha família e a que me deu apoio durante a elaboração deste trabalho, à minha namorada que teve muita paciência comigo, ao meu orientador e co-orientador que me revelaram muitos dos seus conhecimentos e às demais pessoas com que tive discutido o meu trabalho, sejam elas colegas de faculdade ou outros professores.

“Comece fazendo o necessário,
depois o possível e, de repente,
você estará fazendo o impossível”.

São Francisco de Assis

RESUMO

Na Biotecnologia, seqüências de genes são largamente produzidas e fazem com que biólogos tentem achar novas tecnologias para extrair informação delas.

A Informática pode ajudar a resolver este problema oferecendo serviços através da Aprendizagem de Máquina que estuda formas de encontrar padrões em bases de dados.

Este trabalho tenta mostrar como isto pode acontecer. Alguns dados da ***Saccharomyces cerevisiae,*** estudadas por Filho (2003), são submetidas a quatro modelos de Aprendizagem de Máquina supervisionados.

Os resultados mostram que diferentes modelos podem ser comparados em termos de taxas de erro utilizando alguns testes estatísticos.

Palavras-chaves: ***Saccharomyces cerevisiae***; Aprendizagem de Máquina; Informática; BioTecnologia.

ABSTRACT

In BioTechnology, sequences of genes are large produced and makes Biologists try to find new technologies to extract information from them.

Informatics can help to solve this problem offering services through Machine Learning that studies ways to find patterns in data bases.

This work try to show how it can happen. Some data from ***Saccharomyces cerevisiae*** studied by Filho (2003) are submmited to four supervised Machine Learning schemes.

The results show that different schemes can be compared in terms of error taxes using some statistical tests.

Key words: ***Saccharomyces cerevisiae***; Machine Learning; Informatics; BioTechnology

SUMÁRIO

INTRODUÇÃO

No campo computacional, uma grande quantidade de dados nem sempre está disposta de forma organizada para fornecer informações relevantes. A enorme velocidade de geração de dados compete com a rapidez de assimilação dos mesmos. Portanto, em alguns momentos, para que um determinado conjunto de dados torne-se informativo, é necessário submetê-lo a uma filtragem para torná-lo útil e compreensível em futuras análises (HAND *et al.*, 2001).

Há situações em que os dados não são fidedignos, estão incompletos, ou até mesmo não existem (KALAPANIDAS *et al.*, 2003). Por exemplo, suponha o preenchimento de um formulário para uma pesquisa interna a uma empresa que deseja traçar o perfil de seus funcionários. Se perguntas deste formulário do tipo "Qual é a quantidade de horas realmente trabalhadas por dia?" tiverem que ser respondidas diariamente, pode ocorrer de algumas respostas serem mascaradas, evitando-se a ocorrência de uma possível demissão. Porém, outras respostas podem se mostrar incompletas para perguntas do tipo *"Qual é o custo do material escolar da sua prole?"*. Outras ainda, por exemplo, *"Quantas vezes você viaja a trabalho para o exterior?"*, podem nem sequer existir para determinado perfil de funcionário.

Neste contexto, Hand *et al.* (2001) afirmam que nem todos os dados são relevantes para uma análise. Deve-se, portanto, saber escolhê-los e para isto é necessária a participação de especialistas para avaliar quais são os atributos significativos em um conjunto de dados (WITTEN & FRANK, 2000).

Ainda sobre o exemplo citado anteriormente, caso os formulários da empresa fictícia sejam recolhidos após um ano de pesquisa para descobrir o perfil dos funcionários no contexto de quantidade de horas trabalhadas e custos extras para a empresa, pode ser que sejam encontradas tanto informações inúteis ("Viajam ao exterior funcionários que trabalham no mínimo sete horas por dia."), quanto informações significativas ("Funcionários que recebem menos de que dez salários mínimos aceitam fazer hora-extra.").

Verifica-se, então, que em situações como esta, faz-se necessário uma avaliação mais detalhada para evitar dados que possam mascarar, poluir ou mesmo descaracterizar o resultado de uma análise (KALAPANIDAS *et al.*, 2003).

Este cuidado com os dados para se extrair informações valiosas também é visto no campo da Biologia onde o rápido crescimento do volume de dados proveniente de seqüenciamentos genéticos e trabalhos derivados tem levado os biólogos a procurar novas metodologias para descobrir informações escondidas nestas fontes.

Alguns estudos, como por exemplo o de Filho (2003), demonstram que é possível unir a Informática e a Biologia para se encontrar padrões em dados genéticos.

Portanto, percebe-se que técnicas computacionais podem ser utilizadas de forma a contribuir cada vez mais na busca de classificação de dados. E, uma das áreas da Informática mais indicada para se trabalhar com este tipo de procedimento, é a Aprendizagem de Máquina que é o foco deste trabalho.

Considerando-se o exposto acima, questiona-se: É possível comparar modelos de Aprendizagem de Máquina gerados sobre alguns dados da ***Saccharomyces cerevisiae*** utilizando uma abordagem supervisionada?

Para tanto, o estudo realizado em Filho (2003) além de ser utilizado como incentivo, serviu como referencial e fonte de dados para uma abordagem supervisionada sobre os mesmos dados utilizados pelo autor.

Sem querer ter a pretensão de esgotar o assunto, este estudo avaliará a identificação de padrões através de quatro modelos de Aprendizagem de Máquina (Árvore de Decisão, Regras de Decisão, Classificador Bayesiano Ingênuo e K-Vizinhos) sobre dados da levedura ***S. cerevisiae***. A escolha destes modelos deveu-se à limitação e quantificação dos mesmos, uma vez que existem outros modelos que poderiam contribuir também para o objetivo deste trabalho.

Para o processamento dos dados, Witten & Frank (2000) fornece o software escolhido para auxiliar este trabalho. O software Weka, como é conhecido, possui esquemas de aprendizagem de Aprendizagem de Máquina já implementados que, a partir de uma entrada, geram modelos como saída, e permite analisar informações através de dados numéricos ou gráficos.

A estrutura deste trabalho possui a seguinte seqüência: o primeiro capítulo descreve alguns modelos que foram utilizados para uma abordagem supervisionada, onde se conhece a classificação para cada uma das instâncias da base de dados; o segundo capítulo aborda a sistemática de avaliação de modelos de aprendizagem que permite ler a descrição de algumas

possíveis situações com relação à quantidade de dados; o terceiro capítulo apresenta a descrição dos dados utilizados onde se destaca o estudo feito por Filho (2003) sobre o organismo ***S. cerevisia***; o quarto capítulo mostra a descrição dos experimentos baseados na coleta de dados empregados para a qual se adota uma abordagem supervisionada sobre os experimentos; o quinto capítulo descreve os resultados obtidos; e o último capítulo apresenta a conclusão e recomendação de trabalhos futuros.

CAPÍTULO I

1 MODELOS DE APRENDIZAGEM DE MÁQUINA

A “Aprendizagem de Máquina” estuda a previsão de dados futuros baseada na identificação de padrões de um certo conjunto de dados. Este capítulo descreverá alguns modelos que foram utilizados neste trabalho para uma abordagem supervisionada, ou seja, onde já se conhece a classificação para cada uma das instâncias da base de dados.

Na seção “Descrição dos Modelos de Aprendizagem”, há uma breve introdução sobre o manejo dos dados para poder submetê-los aos algoritmos de aprendizagem. E nas seções seguintes estão as descrições dos modelos utilizados neste trabalho.

1.1 DESCRIÇÃO DOS MODELOS DE APRENDIZAGEM

Seja uma base de dados representada por um conjunto de instâncias (linhas), cada uma com atributos (colunas) valorados numérica ou nominalmente, conforme o exemplo visto na Tabela 1.

Na abordagem supervisionada, dentre os atributos, há um que representa o classificador das instâncias. É sobre ele que se deseja inferir quando chegarem novos dados. Após a base de dados inicial ser submetida a um classificador, o resultado obtido é um modelo que pode ser utilizado na classificação de dados novos, ou seja, dados que não foram utilizados na construção do modelo. O principal objetivo dos modelos é encontrar um conjunto de padrões sobre as instâncias analisadas, isto é, fazer com que o computador “aprenda” um comportamento padrão dos dados.

Tabela 1 – Base de dados onde "Jogar Tênis" é o atributo classificatório.

Dia	Visual	Temperatura	Umidade	Vento	Jogar Tênis
D1	ensolarado	quente	alta	fraco	não
D2	ensolarado	quente	alta	forte	não
D3	nublado	quente	alta	fraco	sim
D4	chuvoso	média	alta	fraco	sim
D5	chuvoso	boa	normal	fraco	sim
D6	chuvoso	boa	normal	forte	não
D7	nublado	boa	normal	forte	sim
D8	ensolarado	média	alta	fraco	não
D9	ensolarado	boa	normal	fraco	sim
D10	chuvoso	média	normal	fraco	sim
D11	ensolarado	média	normal	forte	sim
D12	nublado	média	alta	forte	sim
D13	nublado	quente	normal	fraco	sim
D14	chuvoso	média	alta	forte	não

Fonte: Adaptado de Witten & Frank, 2000, p. 9.

O conceito de aprendizado é, portanto, a capacidade de prever, com o auxílio de suposições, o valor de um atributo escolhido como classificatório a partir dos valores dos demais atributos. Esta previsão pode ser expressa na forma de modelos do tipo árvore de decisão, regras de associação, dentre outros.

No processo de aprendizagem, como os modelos são gerados a partir dos dados, é importante ter amostras de dados que retratem bem a população de qual se originam. Uma metodologia aplicada para se garantir a construção de modelos através de dados representativos é a divisão das instâncias em dois conjuntos, o de treinamento e o de teste. O primeiro é responsável pela criação do modelo. O segundo, pela medição da taxa de erro proveniente do modelo criado. Vale ressaltar que as instâncias do primeiro conjunto não são utilizadas para a medição do erro, e nem as do segundo são utilizadas na criação do modelo (MITCHELL, 1997).

Para esclarecer um pouco mais, a taxa de erro é o número de instâncias classificadas incorretamente dividido pelo número total de instâncias. Assim, no caso da Tabela 1, o atributo classificatório "Jogar Tênis" possui duas valorações apenas, sim e não. E um modelo gerado em cima destes dados classificará corretamente as instâncias com sim (*true positives* - TP) e com não (*true negatives* - TN) além de incorretamente com sim (*false positive* - FP) e

com não (*false negative* - FN) também. Portanto, a taxa de erro deste modelo será igual a seguinte fórmula:

$$TaxaErro = \frac{FP + FN}{(FP + FN) + (TP + TN)} \quad (1.1)$$

A seguir, há uma breve descrição dos modelos utilizados neste trabalho.

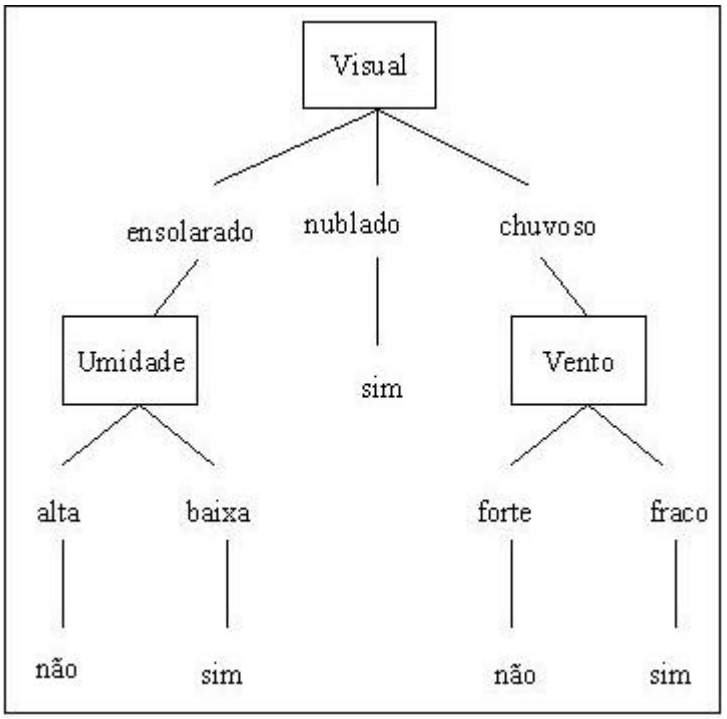
1.2 APRENDIZAGEM POR ÁRVORE DE DECISÃO

O primeiro modelo a ser descrito possui o nome de "Árvore de Decisão". Além de ele ser amplamente utilizado em métodos práticos de inferência indutiva e conseguir fazer aproximações de funções com valores discretos, possui a vantagem de ser robusto para manipular dados com ruídos e ser capaz de aprender expressões disjuntivas. Uma desvantagem deste modelo é que, em alguns casos, a àrvore gerada pode estar sobre-ajustada necessitando ser podada (WITTEN & FRANK, 2000).

Conforme Mitchell (1997), este modelo classifica as instâncias percorrendo uma árvore a partir do nó raiz até alcançar uma folha. Cada um dos nós testa o valor de um único atributo e, para cada uma de suas valorações, oferece arestas diferentes a serem percorridas na árvore a partir deste nó. Sua vantagem é a estratégia adotada conhecida por "dividir-para-conquistar" que divide um problema maior em outros menores. Assim, sua capacidade de discriminação dos dados provém da divisão do espaço definido pelos atributos em sub-espaços.

A Figura 1, que se baseia na Tabela 1, ilustra bem um exemplo de Árvore de Decisão.

Figura 1 – Exemplo de uma árvore de decisão.



Fonte: Adaptado de Mitchell, 1997, p. 53.

Para Witten & Frank (2000), uma característica das árvores de decisão é que cada um dos caminhos desde a raiz até as folhas representa uma conjunção de testes sobre os atributos. Assim, este modelo representa disjunções de conjunções de testes sobre os atributos, ou seja, a Figura 1 pode ser vista da seguinte maneira: (Visual = ensolarado $\wedge$ Umidade = alta $\Rightarrow$ JogarTênis = não) $\vee$ (Visual = nublado $\Rightarrow$ JogarTênis = sim) $\vee$ (Visual = chuvoso $\wedge$ Vento = fraco $\Rightarrow$ JogarTênis = sim).

1.2.1 ID3

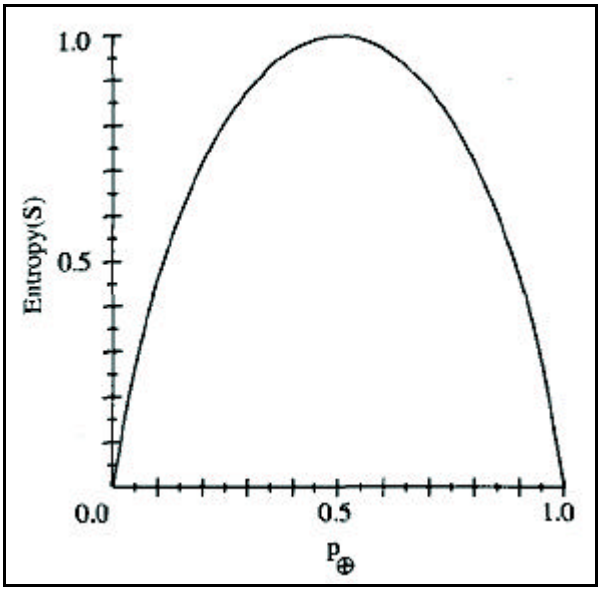
O “ID3” é um dos algoritmos que implementa Árvores de Decisão. De acordo com Witten & Frank (2000), ele é um algoritmo recursivo de busca gulosa que procura, sobre um conjunto de atributos, aqueles que “melhor” se encaixam nas raízes das sub-árvores a serem construídas. Inicialmente, todos os atributos, menos o classificatório, são reunidos em um conjunto. Em seguida, o “melhor” atributo é escolhido e passa a ser a raiz da sub-árvore em construção. Para cada possível valoração deste atributo, é criada uma aresta até as futuras sub-

árvores obtidas com a recursividade deste algoritmo. Os dois únicos critérios de parada são quando não há mais instâncias ou atributos a serem analisados.

Dentre um conjunto de atributos, o "melhor", para ser um nó raiz de uma sub-árvore, é aquele que gera a menor sub-árvore cujas folhas são as mais puras possíveis, ou seja, tendem a possuir instâncias de uma única classe. A função utilizada para esta medição é o "Ganho de Informação". Todavia, é necessário explicar uma outra função conhecida por "Entropia" que a auxilia.

A Entropia possui valores de máximo e de mínimo iguais a um e a zero respectivamente. Ela atinge o seu valor mínimo quando a proporção de ocorrência de uma determinada valoração para um determinado atributo for igual a zero. E atinge o seu valor máximo quando estas proporções forem iguais. Na Figura 2, é possível ver um esboço desta função em relação à proporção de exemplos positivos ($p_{\oplus}$) para a classificação booleana da Tabela 1.

Figura 2 – Função Entropia relativa a uma classificação booleana.
A proporção de exemplos positivos p_A varia entre 0 e 1.



Fonte: Mitchell, 1997, p. 57.

A fórmula da Entropia conforme Mitchell (1997) é:

$$\text{Entropia}(S) = \sum_{i=1}^{c} (-p_i * (\log_2 p_i)) \quad (1.2)$$

Fonte: Adaptado de Mitchell, 1997, p. 57.

onde "S" é o conjunto de instâncias, "c" é a quantidade total de valorações para o atributo classificatório e "p_i" é a proporção de ocorrência da i-ésima valoração do atributo classificatório. A unidade de medida da Entropia é o *bit*.

Assim, o Ganho de Informação de um atributo "A" sobre um conjunto de exemplos "S" é definido pela fórmula:

$$\text{Ganho}(S,A) = \text{Entropia}(S) - \sum_{v \in \text{Valores}(A)} \left(\frac{|S_v|}{|S|} * \text{Entropia}(S_v) \right) \quad (1.3)$$

Fonte: Adaptado de Mitchell, 1997, p. 58.

Através dela é possível medir a redução esperada na Entropia causada pela partição de instâncias de acordo com seus atributos. Esta função prefere atributos com um grande número possível de valores. Desta forma, atributos identificadores teriam a preferência caso fossem utilizados, gerando sub-árvores com tantas partições quanto fosse a quantidade de instâncias analisadas e não informando nada de novo sobre os dados (MITCHELL, 1997).

Para eliminar a escolha de um atributo que tenda a particionar os dados analogamente às partições obtidas com atributos identificadores, utiliza-se uma função conhecida por "Taxa de Ganho" que penaliza, o Ganho de Informação para atributos que dividem os dados uniformemente. A sua fórmula é:

$$\text{TaxaGanho}(S,A) = \text{Ganho}(S,A) \,/\, \text{DivisaoInformação}(S,A) \quad (1.4)$$

Fonte: Adaptado de Mitchell, 1997, p. 74.

onde a "Divisão de Informação" é outra função cuja fórmula é dada por:

$$\text{DivisaoInformação}(S,A) = -\sum_{i=1}^{v} \left(\frac{|S_i|}{|S|} * \log_2 \left(\frac{|S_i|}{|S|} \right) \right) \quad (1.5)$$

Fonte: Adaptado de Mitchell, 1997, p. 73.

onde "|S|" é a cardinalidade do conjunto de instâncias e "$|S_i|$" representa a cardinalidade do subconjunto de instâncias com a i-ésima valoração dentre as "v" possíveis do atributo "A".

Com tantas fórmulas, Witten e Frank (2000) afirmam que a heurística geralmente utilizada na construção de árvores de decisão calcula, inicialmente, o Ganho para cada atributo e, só depois, a Taxa de Ganho para aqueles atributos que obtenham Ganho acima da média dos Ganhos de todos os atributos. No final, o maior resultado indica o "melhor" atributo.

1.2.2 C4.5

Para Quinlan (*apud* Witten & Frank, 2000), o C4.5 é uma melhora do ID3, ou seja, além de possuir as mesmas características, ele possui a vantagem de poder lidar com a poda (*prunning*) da árvore para evitar o sobre-ajustamento, com a ausência de valores, com a valoração numérica de atributos e com a presença de ruídos nos dados.

Ao contrário do algoritmo que o originou, que manipula apenas dados nominais, o C4.5 pode manipular também dados numéricos. Contudo, lidar com este tipo de dado não é tão simples, até porque atributos nominais são testados uma única vez em qualquer caminho da raiz até as folhas, enquanto que atributos numéricos podem ser testados mais de uma vez no mesmo percurso caracterizando uma possível desvantagem do C4.5 pois, em alguns casos, pode tornar a árvore difícil de se entender. Uma forma de se evitar esta dispersão é utilizar um teste que resulta em apenas duas respostas gerando uma árvore de decisão binária (WITTEN & FRANK, 2000). Por exemplo, comparar se a valoração de um atributo é maior que um determinado valor resultará em uma resposta positiva ou negativa.

Além da presença de dados numéricos em conjuntos de dados, outra característica levantada por Witten & Frank (2000) é a ausência de valoração para alguns atributos. Uma das formas que o C4.5 consegue lidar com esta dificuldade, para continuar a construção da árvore, é atribuir um valor fixo ao atributo da instância em questão caso o mesmo seja

significante para a análise de alguma maneira. Caso contrário, ele ignora a instância por completo.

Por último, para evitar o desenvolvimento de árvores com sobre-ajustamento nos dados responsáveis por sua construção, duas técnicas de poda são utilizadas pelo C4.5.

A primeira é conhecida por *preprunning* e consiste em tomar a decisão de parar o desenvolvimento de sub-árvores no decorrer da construção da árvore principal. Assim, evita-se o trabalho de construir uma árvore por completo e, em seguida, eliminar partes dela.

Esta última descrição é basicamente o que a segunda técnica, *postprunning*, faz. Ela é capaz de criar árvores com uma boa performance de classificação. Para isto, utiliza combinações de atributos que somente é possível obter após se gerar uma árvore completa e não podá-la. Tais atributos podem até nem contribuir muito se tomados individualmente, porém, o *postprunning* consegue identificar algumas importantes combinações através de duas operações. Uma é o deslocamento de uma sub-árvore inteira para nós acima de sua posição original. Esta ação é geralmente restringida ao ramo mais popular da árvore pelo fato de ser custosa devido à necessidade de classificar novamente todos os nós da sub-árvore deslocada; a outra operação, que é a substituição de uma sub-árvore por uma simples folha, necessita de estatísticas sobre as taxas de erro esperadas em todos os nós da árvore para comparar o erro de um nó com o de seus filhos.

Assim, para um determinado nível de confiança cujo valor padrão é 25%, calcula-se a estatística do erro combinado dos filhos de um dado nó. Caso o resultado seja maior que o erro do nó pai, os filhos são podados. A diminuição do nível de confiança implica em obter mais podas (WITTEN & FRANK, 2000).

1.3 APRENDIZAGEM POR REGRAS DE DECISÃO

Um conjunto de Regras de Decisão é um dos modelos mais compreensíveis e legíveis para o ser humano segundo Mitchell (1997). O autor também afirma que o aprendizado por este modelo pode seguir uma estratégia de gerar uma regra por vez e eliminar as instâncias cobertas por esta regra do conjunto de treinamento. Assim, a geração de outras regras continua até que não haja mais instâncias no conjunto. Os algoritmos que seguem esta estratégia recebem o nome de algoritmos de cobertura seqüencial.

Para Witten & Frank (2000), o 1R é um algoritmo básico que segue esta estratégia. A sua descrição pode ser observada na Figura 3. Ele suporta tanto atributos com valores

numéricos quanto atributos sem valoração. Neste último caso, o algoritmo trata a ausência de valor simplesmente como um outro valor qualquer.

Figura 3 – Pseudo-código para o algoritmo de cobertura seqüencial 1R.

```
Para cada atributo:
  Para cada valor do atributo, gerar um regra da seguinte forma:
    Contar a quantidade de vezes que cada classe aparece;
    Encontrar a classe mais freqüente;
    Fazer a regra determinar a classe para este valor do atributo.
  Calcular a taxa de erro para todas as regras.
Escolher as regras com a menor taxa de erro.
```

Fonte: Adaptado de Witten & Frank, 2000, p. 79.

Uma forma de se obter um conjunto de regras de decisão é a partir de árvores de decisão. Conforme já foi citado, uma árvore é uma disjunção de conjunções de testes sobre os atributos. Desta forma, é possível criar uma regra para cada disjunção da árvore. O conjunto resultante será composto por regras não ambíguas, porém também será composto por regras muito ajustadas aos dados de treinamento.

Com o intuito de se eliminar este sobre-ajustamento podam-se as regras através de uma abordagem gulosa descrita a seguir. Para cada uma das regras originais, a taxa de erro é calculada sobre o conjunto de treinamento. Cada uma das condições desta regra é eliminada individualmente e novas taxas de erro são calculadas para as novas regras. Caso surja alguma cuja taxa de erro seja menor que a da regra original, esta última é substituída. Repete-se o processo até que não se encontre mais melhoras. No final, é necessário eliminar a ambigüidade possivelmente existente entre as regras resultantes.

Uma desvantagem da abordagem gulosa é o custo computacional significativo que ela demanda, pois a eliminação de cada uma das condições de uma regra deve ser avaliada sobre todo o conjunto de treinamento (WITTEN & FRANK, 2000).

1.3.1 LISTAS DE DECISÃO

Conforme Witten & Frank (2000), as listas de decisão são regras de decisão que devem ser executadas em ordem, pois à medida que uma regra é disparada, algumas instâncias não precisam mais ser avaliadas, ou seja, à medida que a lista é percorrida, o conjunto de dados é lido.

1.4 APRENDIZAGEM BAYESIANA

Em Aprendizagem de Máquina, há um interesse grande em determinar a melhor hipótese sobre um conjunto de instâncias, a partir de alguns dados observados.

Sob este contexto, Mitchell (1997) afirma que as quantidades de cada classe de interesse são governadas por uma distribuição probabilística e as decisões para se classificar otimamente podem ser tomadas levando-se em consideração estas probabilidades juntamente com os dados.

Verifica-se, então, que o conhecimento *a priori* é necessário para o desenvolvimento deste método e, para cada uma das possíveis hipóteses, pode ser associada uma probabilidade *a priori*, possibilitando, assim, o suporte a mais de uma hipótese através de pesos.

O Teorema de Bayes permite calcular a melhor hipótese baseado em probabilidades *a priori*. Sejam a hipótese "h" e o conjunto de treinamento "D". Entende-se "P(h)" pela probabilidade inicial de uma hipótese "h" acontecer antes de se observar qualquer conjunto de treinamento, também conhecida por probabilidade *a priori*. Caso não haja esta informação, admite-se que cada uma das possíveis hipóteses possui a mesma probabilidade. De forma análoga, "P(D)" é a probabilidade *a priori* do conjunto de treinamento antes de se admitir alguma hipótese para este conjunto. Já "P(D|h)" significa a probabilidade de se observar o conjunto de treinamento admitindo-se a hipótese "h", ou seja, é a probabilidade de "D" dado "h".

Nos problemas de Aprendizagem de Máquina, o foco é na "P(h|D)", ou seja, na probabilidade posterior de "h" dado o conjunto de treinamento "D". Ela mede a influência do conjunto de treinamento em contraste com a probabilidade *a priori* (MITCHELL, 1997). Abaixo, pode-se observar a fórmula do Teorema de Bayes.

$$P(h|D) = \frac{P(D|h) * P(h)}{P(D)} \quad (1.6)$$

Fonte: Adaptado de Mitchell, 1997, p. 156.

1.4.1 CLASSIFICADOR BAYESIANO INGÊNUO

Dentre os métodos de aprendizado Bayesiano, existe um conhecido por Classificador Ingênuo de Bayes que, em alguns casos, pode apresentar bons resultados de performance, se comparado a de outras metodologias, principalmente quando combinado com alguns métodos de seleção de atributos para a eliminação de redundância de informação. A vantagem deste classificador advém da simplicidade no seu cálculo, pois admite, ingenuamente, independência entre atributos resultando na busca pela classificação que maximiza o produtório da sua fórmula. Todavia, pode ser que haja casos em que esta suposição não retrate a realidade, prejudicando a análise final (MITCHELL, 1997).

A fórmula do Teorema de Bayes para classificadores ingênuos é

$$h_{NB} = \underset{h_j \in H}{\operatorname{argmax}} P(h_j) * \prod_i P(a_i|h_j) \quad (1.7)$$

Fonte: Adaptado de Mitchell, 1997, p. 177.

onde a hipótese de *Naive Bayes* ("h_{NB}") é a que maximiza o valor de um produto entre a probabilidade de ocorrência de uma hipótese "h_j" (uma entre as possíveis no conjunto de hipóteses H) e um produtório de probabilidades das valorações dos i-ésimos atributos dada a hipótese "h_j".

Um problema que pode ser alcançado por este classificador é quando a combinação de uma possível valoração de um atributo com uma certa classe não ocorre, ou seja, o contador é igual a zero. Isto faz com que esta probabilidade seja zero e, conseqüentemente, o h_{NB} também o seja. Uma saída para este problema é utilizar o estimador Laplaciano, o qual

consiste em iniciar o contador de cada valoração possível de um atributo com o número 1. Um caso especial do problema citado é quando existem instâncias cuja valoração do atributo é inexistente. Neste caso, o contador de freqüência não é incrementado e a probabilidade se baseia no número de instâncias valoradas ao invés de se basear no número total de instâncias (MITCHELL, 1997).

1.5 APRENDIZAGEM BASEADA EM INSTÂNCIA

Finalizando o capítulo, demonstra-se uma descrição sobre o método do Aprendizado Baseado em Instância.

Segundo Mitchell (1997), em contraste com os demais métodos de aprendizagem, que criam modelos sobre o conjunto de treinamento para classificar as novas instâncias, o Aprendizado Baseado em Instância atribui uma classificação a cada elemento do conjunto de treinamento e os armazena para poder classificar as novas instâncias. A generalização oferecida por outros modelos é feita sob demanda neste método, ou seja, à medida que chegam novas instâncias. Eis o porquê de ser denominado de método preguiçoso de aprendizagem e da demora no processamento fazendo com que a nova instância deva ser comparada a todas as instâncias já classificadas.

1.5.1 K-VIZINHOS

O método mais básico de Aprendizado Baseado em Instância é o algoritmo k-Vizinhos mais Próximos, ou apenas k-Vizinhos, onde as instâncias são agrupadas conforme a maior proximidade entre elas. A função que mede a distância entre as instâncias é de suma importância para este algoritmo. Um exemplo de função que é muito utilizada, quando o atributo é do tipo numérico, é a distância Euclidiana (MITCHELL, 1997). A sua fórmula é

$$d_E(r, s) = \sqrt{\sum_{i=1}^{n} (a_{i,r} - a_{i,s})^2} \quad (1.8)$$

Fonte: Adaptado de Witten & Frank, 2000, p. 115.

onde, para duas instâncias "r" e "s", "a_i" representa o valor do i-ésimo atributo, num total de "n" atributos.

Porém, Witten & Frank (2000) alertam para as situações em que os atributos são medidos em diferentes escalas onde, para evitar uma análise incorreta, os seus valores devem

ser normalizados entre 0 e 1. Assim, como as valorações dos atributos da instância em questão são comparadas às valorações de todas as demais instâncias, a classificação dos k vizinhos mais próximos definem a classificação da instância em questão. Uma forma adotada para esta classificação final pode ser o voto majoritário, por exemplo.

A fórmula utilizada na normalização é representada por

$$a_{i,x} = \frac{v_i - \min(v_i)}{\max(v_i) - \min(v_i)} \quad (1.9)$$

Fonte: Adaptado de Witten & Frank, 2000, p. 115.

onde "$a_{i,x}$" representa o ~~i~~ésimo atributo da instância x, "v_i" representa o valor atual do ~~i~~ ésimo atributo e "min(v_i)" e "max(v_i)" representam, respectivamente, os valores mínimo e máximo deste atributo dentre todas as instâncias da base de dados.

Uma desvantagem observada neste classificador é a possibilidade de se classificar incorretamente uma instância onde os k-Vizinhos não corresponda corretamente à maioria dos vizinhos da instância em questão.

CAPÍTULO II

2 AVALIAÇÃO DOS MODELOS

Após criar os modelos de aprendizagem sobre o conjunto de treino e verificar sua taxa de erro em um conjunto de teste, necessita-se avaliar quão diferentes os métodos são. À primeira vista, avaliar modelos pode parecer simples, mas este capítulo mostra que não é bem assim.

Na seção "Avaliação", é possível ler a descrição de algumas das possíveis situações com relação à quantidade de dados a ser avaliada. Em "Divisão dos Conjuntos de dados", é explicado o porquê de se usar o conjunto de treinamento e de teste. Como fazer esta divisão em um conjunto limitado de dados é visto em "Validação Cruzada em K Pastas". E, na última seção do capítulo, descreve-se como Witten & Frank (2000) sugerem uma comparação entre classificadores.

2.1 AVALIAÇÃO

Supondo que exista uma quantidade enorme de dados e que uma análise seja requisitada sobre ela, a saída indicada por Witten & Frank (2000) é dividir a base de dados em dois grandes conjuntos: o de treinamento e o de teste. O modelo é gerado a partir do primeiro e a avaliação da taxa de erro é feita sobre o segundo. Quanto maior for o conjunto de treinamento, melhor será o classificador. E quanto maior for o conjunto de teste, mais precisa será a taxa de erro.

Assim, para uma situação mais delicada que a anterior, ou seja, para uma base de dados limitada, deve-se ter um maior cuidado com os dois conjuntos para se evitar uma análise mascarada. Nestes casos, um dos processos indicados por Witten & Frank (2000) é obter amostras representativas com relação à bases de dados através de um processo conhecido por "Validação Cruzada em K pastas".

2.2 DIVISÃO DOS CONJUNTOS DE DADOS

Já que os dados são os responsáveis pela criação do modelo e por sua validação então é necessário saber manipulá-los para se obter uma performance satisfatória do classificador, ou modelo, como também é conhecido.

Uma forma bem intuitiva de se medir o nível da performance de um classificador é geralmente através da taxa de erro. O erro é obtido no caso do classificador inferir incorretamente a classe de uma instância, enquanto que o sucesso é obtido em caso contrário. Caso o modelo seja avaliado sobre os dados que o construíram, pode-se obter um resultado mascarado. Assim, prefere-se dividir os dados em dois conjuntos independentes antes de qualquer construção, o de treinamento e o de teste. Na prática, é comum deixar dois terços para o conjunto de treinamento e um terço para o de teste (WITTEN & FRANK, 2000). Outra saída é fazer uso de uma técnica que será explicada na próxima seção.

2.3 VALIDAÇÃO CRUZADA EM K PASTAS

Witten & Frank (2000) afirmam que, durante a divisão dos dados, deve-se garantir a mesma proporção de classes existentes na base de dados original para ambos os conjuntos. Assim, é possível obter uma avaliação mais próxima do real. Um processo que tende a diminuir as diferenças obtidas entre amostras particulares da população em questão é repetir a obtenção dos conjuntos de treinamento e de teste.

A Validação Cruzada em K pastas é uma técnica estatística que funciona conforme explicado a seguir. K é igual tanto ao número de pastas em que a base de dados é particionada quanto ao número de iterações que ocorre sobre a base. As pastas possuem uma quantidade de instâncias o mais igual possível. Cada iteração escolhe uma das pastas, dentre as que ainda não foram escolhidas, elegendo-a como o conjunto de teste e elegendo as demais como um único conjunto de treinamento. Em seguida, o modelo é criado e avaliado. No final das iterações, a taxa de erro do aprendizado é a média das taxas de erro dos K modelos obtidos.

Vale ressaltar que a técnica, aqui avaliada, não elimina a variação da taxa de erro proveniente de amostras distintas, mas consegue reduzi-la. Uma estimativa mais precisa pode ser obtida repetindo-se este procedimento estatístico dez vezes e obtendo-se a média destas repetições (WITTEN & FRANK, 2000).

2.4 COMPARAÇÃO ENTRE CLASSIFICADORES

Após a criação de modelos, deseja-se compará-los para saber qual é o mais satisfatório. Na seção anterior, uma aproximação da verdadeira taxa de erro é obtida. Porém, necessita-se de resultados mais confiantes.

Abaixo, encontra-se a descrição de como se fazer uma comparação entre diferentes classificadores através de teste de hipóteses e de intervalos de confiança conforme afirmam Witten & Frank (2000) e Bussab & Morettin (2003).

Inicialmente, a partir de uma única base de dados, dois conjuntos de amostras independentes e aleatórias a_1, ... a_N e b_1, ... b_N são obtidos. Fazendo-se N = K, cada uma das amostras é então submetida a diferentes métodos de aprendizagem com Validação Cruzada em K pastas. Para se obter melhores resultados, deve-se garantir que as i-ésimas partições para amostras pareadas (a_1 e b_1, ... a_N e b_N) sejam iguais, ou seja, a partição 1 das amostras a_1 e b_1 sejam iguais, ..., a partição K das amostras a_N e b_N sejam iguais. Esta última observação ressalta uma característica de um "teste emparelhado".

Através do procedimento de Validação Cruzada em K pastas, obtém-se uma média da taxa de erro para cada amostra (x_1 é a média de a_1, ..., x_k é a média de a_k e y_1 é a média de b_1, ..., y_k é a média de b_k) e as médias destes resultados (x_m e y_m).

Em seguida, assume-se que os dados amostrais sobre os quais se deseja fazer a análise são provenientes de uma população que possui uma distribuição de probabilidade Normal, ou seja, N(média = μ, variância = σ^2). Com esta suposição, pode-se afirmar que a variável aleatória $d_i = x_i - y_i$ possui uma distribuição Normal $N(\mu_d, \sigma^2_d)$ e que a variável aleatória $d_m = x_m - y_m$ possui uma distribuição Normal $N(\mu_d, \sigma^2_d/k)$. Como a variância da população estudada é desconhecida, recorre-se a uma aproximação cuja fórmula é

$$S_d^2 = \frac{1}{n-1} * \sum_{i=1}^{k} (d_i - d_m)^2 \qquad (2.1)$$

Fonte: Adaptado de Bussab &Morettin, 2003, p. 375.

Desta forma, a estatística T, cuja aproximação da variância é S_d^2, tem distribuição *t* de *Student*, com (k-1) graus de liberdade e sua fórmula é

$$T = \frac{d_m}{S_d / \sqrt{n}} \qquad (2.2)$$

Fonte: Adaptado de Bussab &Morettin, 2003, p. 376.

Dando continuidade a linha de raciocínio, a comparação entre os modelos pode adotar um teste de hipóteses unilateral onde a hipótese nula afirma que a média das diferenças da população é igual a zero. E, em contrapartida, a hipótese alternativa afirma que a média das diferenças da população é maior que zero conforme as fórmulas abaixo, onde H_0 é a hipótese nula e H_1 é a hipótese alternativa.

$$\begin{cases} H_0 : \mu_d = 0 \\ H_1 : \mu_d > 0 \end{cases} \quad (2.3)$$

Fonte: Adaptado de Bussab &Morettin, 2003, p. 376.

A característica principal do teste de hipóteses é informar a probabilidade de se cometer um erro do tipo I, ou seja, de rejeitar a hipótese nula quando ela for verdadeira, para um dado nível α de significância. Portanto, é através deste teste e da estatística T que se sabe se dois modelos diferem a um certo nível α de significância.

Na seguinte fórmula

$$P(\text{erro do tipo I}) = P(T > t_c) = \alpha \quad (2.4)$$

Fonte: Adaptado de Bussab &Morettin, 2003, p. 331.

é possível observar a fórmula da probabilidade de se ocorrer um erro do tipo I, ou seja, da estatística T ser maior que um valor t_c tabelado pela distribuição *t* de *Student* a um nível α de significância.

E para finalizar a comparação, é possível construir um intervalo de confiança IC para um classificador a partir do estimador T com um nível de confiança igual a $(1 - \alpha)$ seguindo a fórmula

$$IC = [d_m - t_{\alpha/2} * \sqrt{(S_d^2/n)},\ d_m + t_{\alpha/2} * \sqrt{(S_d^2/n)}\ ] \quad (2.5)$$

Fonte: Adaptado de Bussab &Morettin, 2003, p. 306.

CAPÍTULO III

3 DADOS

Neste capítulo, há uma descrição sobre o conjunto de dados utilizado neste trabalho. Experimentos como os de Filho (2003) e de Cho *et al.* (1998) que auxiliaram este estudo também são brevemente explanados.

3.1 ORIGEM DOS DADOS

Os dados estudados, neste trabalho, é um subconjunto de dados utilizados por Filho (2003), mais especificamente o *Series CDC 25* do *Mitotic Cell Cycle*, sobre os quais o autor elaborou um estudo comparativo de cinco métodos de agrupamento (agrupamento hierárquico aglomerativo, CLICK, agrupamento dinâmico, K-médias e mapas auto-organizáveis) e três índices de proximidade (distância Euclidiana, separação angular e correlação de Pearson) para a análise de séries temporais de expressão gênica do organismo ***S. cerevisiae***.

Além disto, com o intuito de avaliar os métodos, o autor empregou um procedimento de Validação Cruzada adaptado para métodos não-supervisionados. A precisão dos resultados foi medida através da comparação das partições obtidas nestes experimentos com dados de anotação de genes, como função de proteínas e classificação de séries temporais.

Tais partições seguem os níveis de um esquema de classificação de funções protéicas da levedura disponibilizado pelo *Munich Information Center for Protein Sequences Yeast Genome Database* (MYGD). Tal esquema é composto por uma árvore com 249 classes divididas em cinco níveis, de acordo com informações bioquímicas e estudos gênicos (FILHO, 2003). O autor percebeu a existência de instâncias que se encaixavam em mais de um dos cinco níveis do esquema do MYGD e criou um sexto nível.

Na Tabela 2 a seguir, encontra-se o esquema do MYGD seguido por Filho (2003).

Tabela 2 – Esquema de níveis do MYGD seguido por Filho (2003).

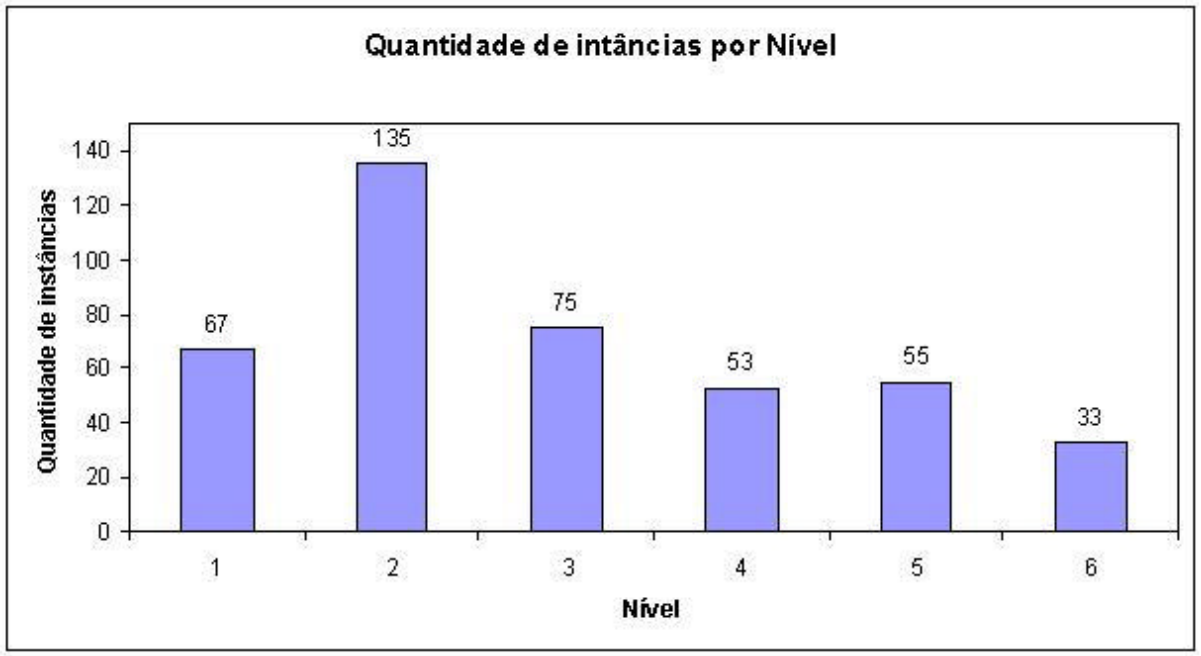
Nível	Quantidade de classes
1	16
2	107
3	85
4	39
5	2

Fonte: Adaptado de Filho, 2003, p. 59.

A base de dados *Mitotic Cell Cycle* resultado do experimento de Cho *et al.* (1998) utilizado por Filho (2003) mede, através de arrays de oligonucleotídeos, níveis de transcrição de RNAm (Ácido Ribonucléico mensageiro) de células da ***S. cerevisiae*** sincronizadas em 17 intervalos regulares, com duração de dez minutos cada, durante o ciclo celular. Estas medições, ao invés de terem sido feitas nas quatro sub-fases da divisão celular da levedura (G1, S G2 e M), foram feitas em cinco sub-fases denominadas por Cho *et al.* (1998) de G1 anterior, G1 posterior, G2 , S G2 e M que compõem a classificação visual sobre a qual está baseada os dados do *Series CDC 25*, acima citado.

A união dos resultados obtidos por Cho *et al.* (1998) e por Filho (2003) formam a base de dados deste trabalho, a qual é constituída por 418 instâncias com 17 atributos numéricos resultantes do estudo de Cho *et al.* (1998) e um atributo nominal e classificatório baseado em seis níveis, onde cinco níveis são do esquema do MYGD e um nível é para as instâncias que se enquadram em mais de um destes cinco. Na Figura 4, é possível visualizar a distribuição das instâncias pelo nível de classificação adotado por Filho (2003).

Figura 4 – Histograma dos níveis utilizados por Filho (2003)



3.2 MODELAGEM DOS DADOS

Os dados utilizados neste trabalho foram modelados em um conjunto de instâncias (linhas) valoradas em seus atributos (colunas) numéricos e nominais, onde um dos atributos é o que classifica a instância. A modelagem dos dados foi feita desta forma, pois, através de métodos de Aprendizagem de Máquina, é possível encontrar padrões entre os atributos das instâncias, de forma a construir modelos capazes de classificar dados novos.

3.3 ORGANISMO ESTUDADO

O organismo em foco é a levedura ***S. cerevisiae***. De acordo com o CYGD (2004), um instituto alemão responsável por estudar e armazenar as seqüências de proteínas da levedura, o código genético do organismo estudado é conhecido por completo e possui um genoma de tamanho considerado pequeno, por volta de 13.478.000 bases nucleotídicas. Além do mais, dentre os seus 16 cromossomos, um total de 6335 ORFs (*Open Read Frames*) são conhecidos e apenas 3307 proteínas foram caracterizadas. Vale ressaltar que os dados acima citados são mais recentes que os dados que serviram de base para este trabalho.

CAPÍTULO IV

4 EXPERIMENTOS

Os experimentos deste trabalho resumiram-se em coletar os resultados obtidos por Filho (2003) e por Cho *et al.* (1998), pré-processá-los, submetê-los a cinco diferentes classificadores e comparar as suas performances dois-a-dois através de um teste emparelhado.

Pelo fato de Filho (2003) ter feito um experimento não-supervisionado e de os dados utilizados neste trabalho possuírem valoração para os atributos classificadores, adotou-se uma abordagem supervisionada sobre os experimentos.

4.1 PRÉ-PROCESSAMENTO

A criação da base de dados necessitou de um processamento dos dados provenientes de Filho (2003) e de Cho *et al.* (1998), os quais foram tratados por alguns scripts escritos na linguagem Java para transformá-los no formato legível pelo Weka, ou seja, o ARFF (vide "APÊNDICE A – Formato ARFF" e "APÊNDICE B – Weka").

Em seguida, para se criar conjuntos de treinamentos e de testes, seguiu-se uma sugestão de Witten & Frank (2000). Esta sugestão consiste em uma repetição de 10 vezes o processo de Validação Cruzada em 10 pastas para aumentar as possibilidades de se manter a mesma proporção de classes tanto na base de dados original quanto nos conjuntos de treinamento e de teste gerados.

Porém no Weka, a Validação Cruzada cria agrupamentos de instâncias em quantidades a partir da primeira instância da base de dados. Observa-se, então, a importância na ordenação das instâncias de uma amostra.

Daí, com o módulo *Preprocess* do *Weka Explorer*, sobre a base de dados, seguiram-se dez aplicações de um filtro, cujo nome é *Resample*, que ordena aleatoriamente as instâncias na base de dados Cada aplicação gerou uma amostra "v_i", onde i varia de 0 a 9. Assim, 10 amostras contendo todas as instâncias da base foram geradas.

4.2 PROCESSAMENTO

Sobre cada uma das 10 amostras obtidas, foram aplicados 5 diferentes classificadores com Validação Cruzada em 10 pastas através do módulo *Classify* do *Weka Explorer* seguindo os parâmetros padrões do Weka.

Vale ressaltar que, para uma amostra qualquer, basta habilitar a opção de Validação Cruzada do módulo *Classify* do *Weka Explorer* para garantir que as partições obtidas por este procedimento sejam as mesmas em diferentes classificadores. Como estes procedimentos foram executados neste trabalho e conforme citado na seção "2.4", testes emparelhados puderam ser realizados.

O primeiro classificador utilizado foi o "J48" que é uma versão do C4.5 no Weka. Seus parâmetros padrões são: nível_de_confiança=0,25, podar_árvore=falso e número_mínimo_de_instâncias_por_folha=2 (vide seção "1.2.2").

O segundo foi o "PART", que é uma versão de lista de decisão no Weka onde se constrói uma árvore parcial C4.5 seguindo uma estratégia de "dividir-para-conquistar" para auxiliar os resultados. Seus parâmetros padrões são: nível_de_confiança=0,25, podar_árvore=falso e número_mínimo_de_instâncias_por_folha=2 (vide seção "1.3.1").

O "Naive Bayes", que é um dos classificadores ingênuo de Bayes no Weka, foi o terceiro classificador utilizado. Valores de estimativas numéricas necessárias para seus cálculos são escolhidos dependendo da análise do conjunto de treinamento (vide seção "1.4").

Os dois últimos classificadores seguiram uma versão do k-Vizinhos para o Weka cujo nome é "LBk". Um deles foi valorado com k=3 e o outro com k=1. O LBk utiliza a distância Euclidiana normalizada para encontrar as "k" instâncias (que já foram classificadas no treinamento) mais próximas da instância de teste. Após encontrar, ele faz a predição com a classe que predomina nas "k" instâncias já treinadas. Se múltiplas instâncias tiverem a mesma (menor) distância, a primeira encontrada é utilizada (vide seção "1.5.1").

No final, cada classificador possuirá 10 médias de taxas de erro por amostra. Isto se explica, pois, fixados um classificador e uma amostra, o procedimento de Validação Cruzada em 10 pastas possui um total de 10 iterações obtendo como resultado uma média da taxa de erro.

4.3 AVALIAÇÃO DE MODELOS DE APRENDIZAGEM

Os modelos de Aprendizagem de Máquina utilizados neste estudo foram avaliados com o auxílio da ferramenta "Minitab® Release 14 – Statistical Software" da seguinte forma. Primeiramente, as taxas de erro foram estimadas pontualmente para cada algoritmo. Logo em seguida, os seus intervalos de confiança foram obtidos com níveis de significância $\alpha_1 = 0,01$ e $\alpha_2 = 0,05$.

Depois, para se comparar os classificadores utilizou-se de uma variável aleatória que mediu as diferenças das médias das taxas de erro entre eles. E, para se chegar a um teste de hipóteses definitivo, utilizou-se outro teste, o de Kolmogorov-Smirnov, para saber se as distribuições destas diferenças seguiam ou não uma distribuição Normal. Descobrindo-se quais combinações seguiam tal distribuição, testes emparelhados *t* de *Student* com 9 graus de liberdade foram executados. E, sobre as demais combinações, foram executados testes não-paramétricos conhecidos por teste de Mann-Withney.

No próximo capítulo, podem-se observar os resultados alcançados por esta avaliação.

CAPÍTULO V

5 RESULTADOS

Após o processamento dos dados pelo Weka, utilizou-se uma licença temporariamente gratuita do software "Minitab® Release 14 – Statistical Software" para se fazer a análise estatística dos mesmos.

O primeiro passo foi gerar dez médias de taxas de erro para cada um dos 5 classificadores (J48, PART, Naive Bayes, LB3 e LB1) sobre as amostras (v0, v1, v2, v3, v4, v5, v6, v7, v8 e v9). Cada uma destas médias foi obtida diretamente do modelo gerado pelo classificador correspondente.

Em seguida, as estimações pontuais (e.p.) das médias e os desvios padrões (d.p.) foram obtidos para o J48 (e.p.=47,6794% e d.p.=2,7110), para o Naive Bayes (e.p.= 67,5598% e d.p.= 0,5770), para o PART (e.p.=45,1435% e d.p.=1,5990), para o LB3 (e.p.=39,9761% e d.p.=1,2580) e, por último, para o LB1 (e.p.=42,3206% e d.p.=0,9670) conforme pode-se visualizar na Tabela 3.

Tabela 3 – Estimação pontual para as taxas de erro dos quatro classificadores.

Amostras	J48 (%)	Naive Bayes (%)	PART (%)	LB3 (%)	LB1 (%)
v0	44,2584	67,2249	44,4976	41,1483	42,5837
v1	46,4115	67,7033	44,2584	41,1483	43,0622
v2	43,0622	67,9426	44,7368	40,6699	43,5407
v3	47,1292	67,9426	44,7368	38,0383	43,5407
v4	48,8038	67,4641	44,0191	39,7129	41,6268
v5	48,0861	67,2249	42,3445	39,9522	40,9091
v6	49,5215	66,7464	46,1722	40,6699	41,8660
v7	49,0431	66,7464	48,0861	38,9952	43,0622
v8	52,6316	68,4211	46,4115	41,3876	41,1483
v9	47,8469	68,1818	46,1722	38,0383	41,8660
Estimação Pontual	**47,6794**	**67,5598**	**45,1435**	**39,9761**	**42,3206**
Desvio Padrão	**2,7110**	**0,5770**	**1,5990**	**1,2580**	**0,9670**

Outra forma de analisar as taxas de erro para cada classificador é através dos histogramas visualizados nas figuras 5, 6, 7, 8 e 9. Para melhor entendê-los basta-se explicar um deles.

Portanto, na Figura 5, é possível verificar o histograma das taxas de erro para o classificador J48, onde é possível observar que há duas amostras com taxas de erro no

intervalo entre 43% e 45%, uma amostra entre 45% e 47%, quatro amostras entre 47% e 49%, duas amostras entre 49% e 51% e, finalmente, uma amostra entre 51 e 53%.

Figura 5 – Histograma da taxas de erro do classificador J48.

Figura 6 – Histograma da taxas de erro do classificador Naive Bayes.

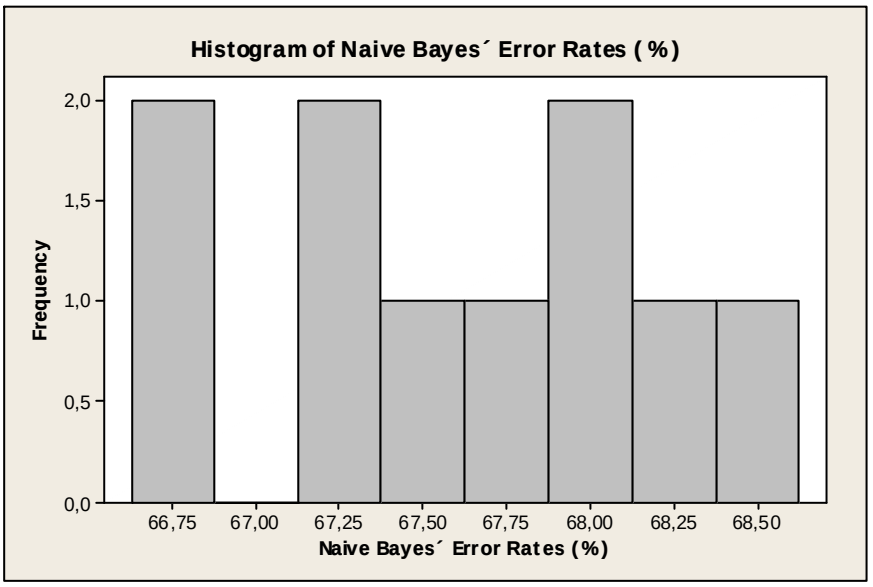


Figura 7 – Histograma da taxas de erro do classificador PART.

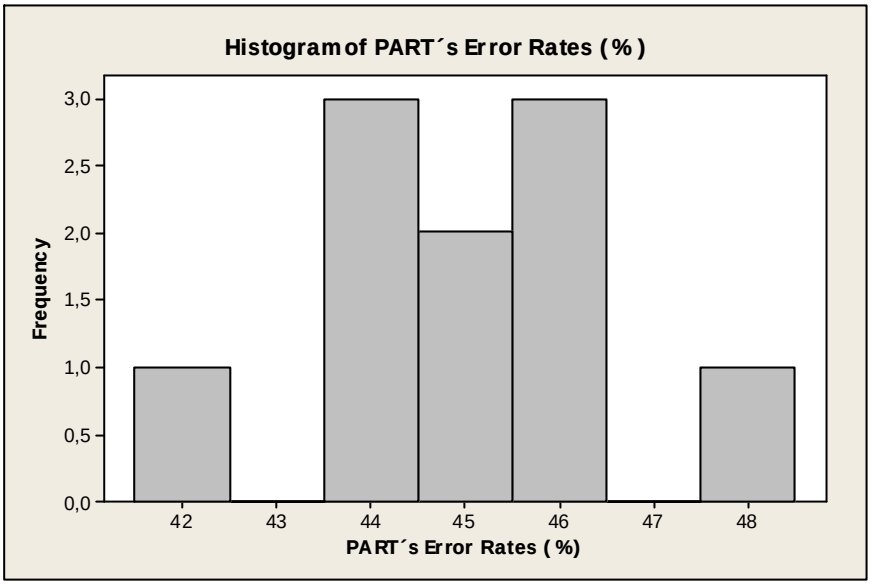


Figura 8 – Histograma da taxas de erro do classificador LB3.

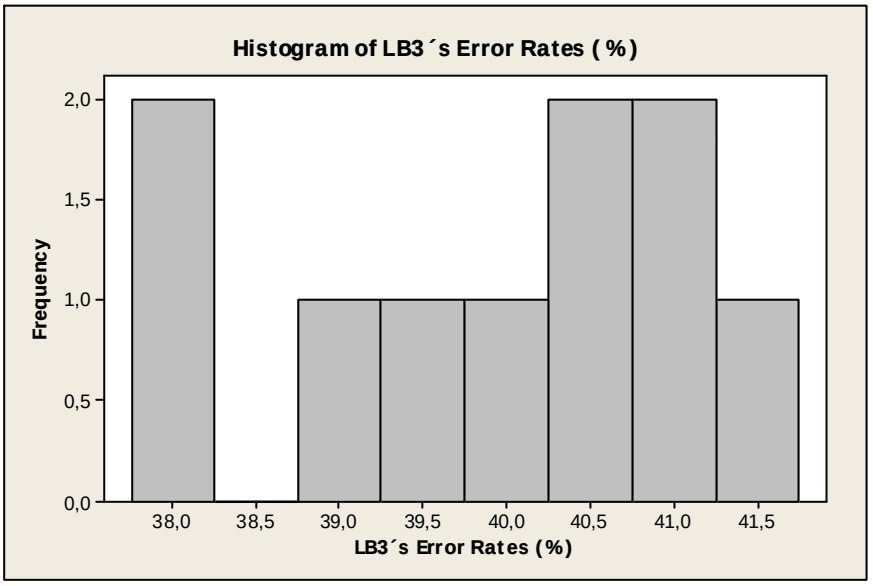
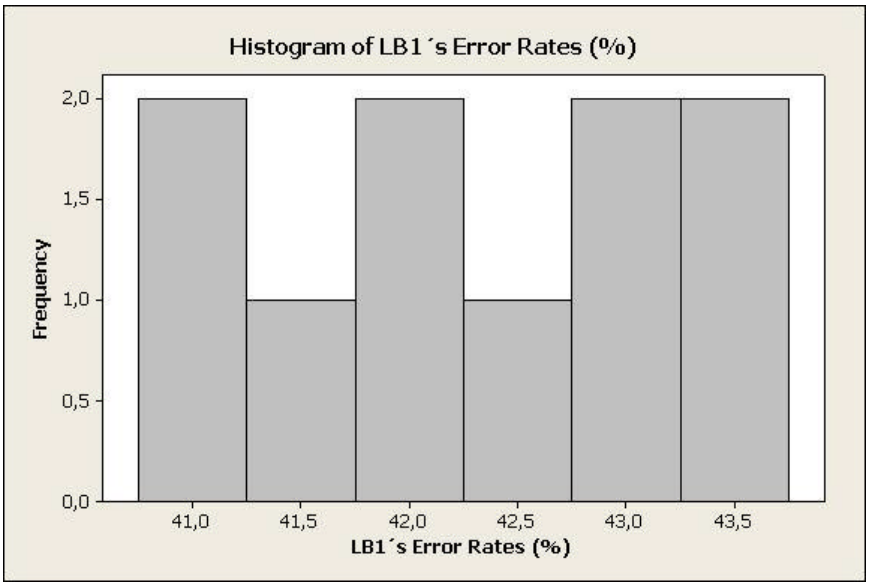


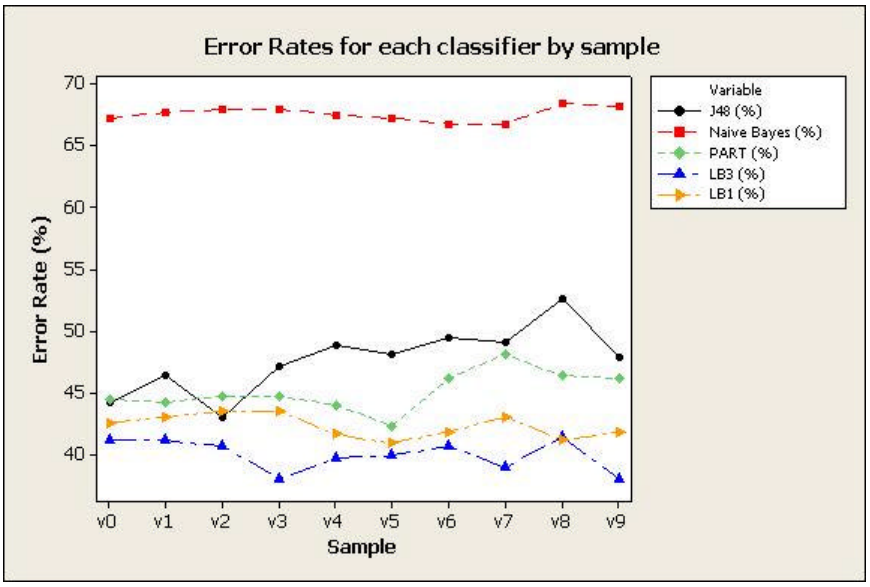
Figura 9 – Histograma da taxas de erro do classificador LB1.



Além dos histogramas, uma visão mais comparativa entre os classificadores pode ser feita na Figura 10 onde estão impressas, em um único gráfico, as taxas de erro obtidas dos cinco classificadores à medida em que foram submetidos à geração de modelos para cada uma das amostras. Nesta figura é possível verificar que o Naive Bayes possui as mais altas taxas de erro seguidas pelas do J48, PART e LB1 e, por último, pelas do LB3. Pode-se verificar também, de certa forma, que o J48, o PART, o LB1 e o LB3 formam um agrupamento à parte se comparados ao Naive Bayes.

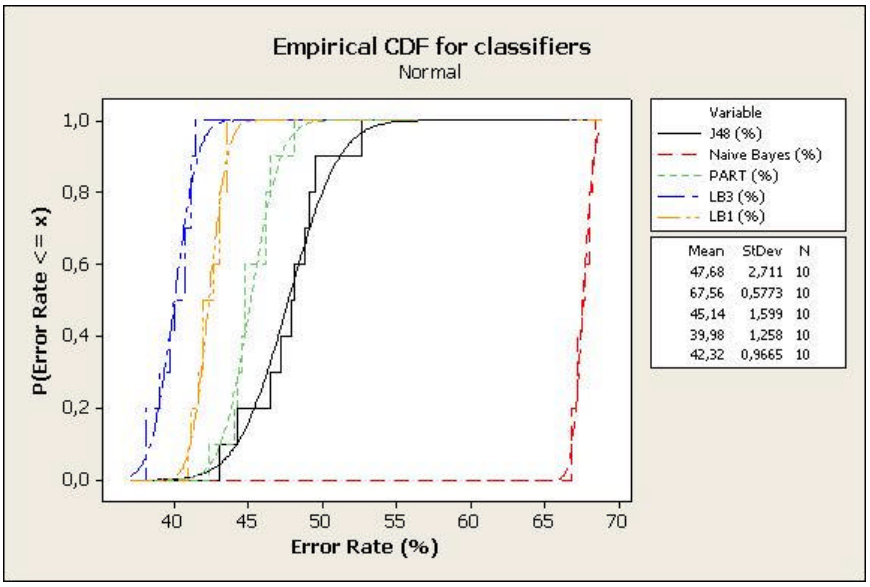
Outra observação é que as médias das taxas de erro obtidas foram altas para todos os classificadores, principalmente para o Naive Bayes, pois o experimento utilizou os classificadores do Weka apenas com seus parâmetros padrões. Sobre estas altas médias obtidas, uma comparação importante de ser relatada é entre o LB1 e o LB3 onde, pode-se constatar que o modelo de Aprendizagem de Máquina k-Vizinhos com k=3 obteve melhores resultados com as médias sendo menores que LB1 em sua maioria.

Figura 10 – Taxas de erro para cada um dos classificadores



Os gráficos das Funções de Distribuição Empírica e uma aproximação das Funções de Distribuição Cumulativa para as médias de taxas de erro dos classificadores estão na Figura 11. Nela pode-se observar, da esquerda para a direita, os gráficos do LB3, do LB1, do PART, do J48 e do Naive Bayes. Verifica-se, no eixo das ordenadas, a probabilidade de a média de taxas de erro ser menor ou igual a um valor x do eixo das abscissas. Assim, a probabilidade de que ocorra uma média de taxa de erro menor que 41,03% sobre o LB3 é de 0,8, ou seja, $P(\text{Taxa de Erro}_{LB3} <= 41{,}03\%) = 0{,}8$.

Figura 11 – Função de Distribuição Cumulativa



O intervalo de confiança para a média populacional das médias das taxas de erro também foi obtido para cada classificador (μ_{J48}, $\mu_{NaiveBayes}$, μ_{PART}, μ_{LB1} e μ_{LB3}) com níveis de confiança de 99,5% (α_1=0,01) e de 95 % (α_2=0,05). Assim, os valores numéricos dos limites superior e inferior e da média destes limites para cada classificador podem ser vistos na Tabela 4 e na Tabela 5, enquanto que, na Figura 12 e na Figura 13, estão os gráficos dos intervalos de confiança sobre os quais percebe-se a interseção entre dois deles, o do J48 e o do PART. Isto indica que μ_{J48} e μ_{PART} podem vir a serem iguais.

Tabela 4 – Dados referentes aos intervalos de confiança a 99,5% dos classificadores J48, Naive Bayes, PART e LB3.

$alfa_1$ = 0,01	J48	Naive Bayes	PART	LB1	LB3
Limite Superior	50,4659	68,1530	46,7869	43,3139	41,2690
Limite Inferior	44,8929	66,9666	43,5002	41,3273	38,6832
Média	47,6794	67,5598	45,1435	42,3206	39,9761

Tabela 5 – Dados referentes aos intervalos de confiança a 95% dos classificadores J48, Naive Bayes, PART e LB3.

$alfa_2$ = 0,05	J48	Naive Bayes	PART	LB1	LB3
Limite Superior	49,6191	67,9728	46,2875	43,0120	40,8761
Limite Inferior	45,7397	67,1469	43,9996	41,6292	39,0761
Média	47,6794	67,5598	45,1435	42,3206	39,9761

Figura 12 - Gráfico dos intervalos de confiança a 99,5% dos classificadores J48, Naive Bayes, PART e LB3.

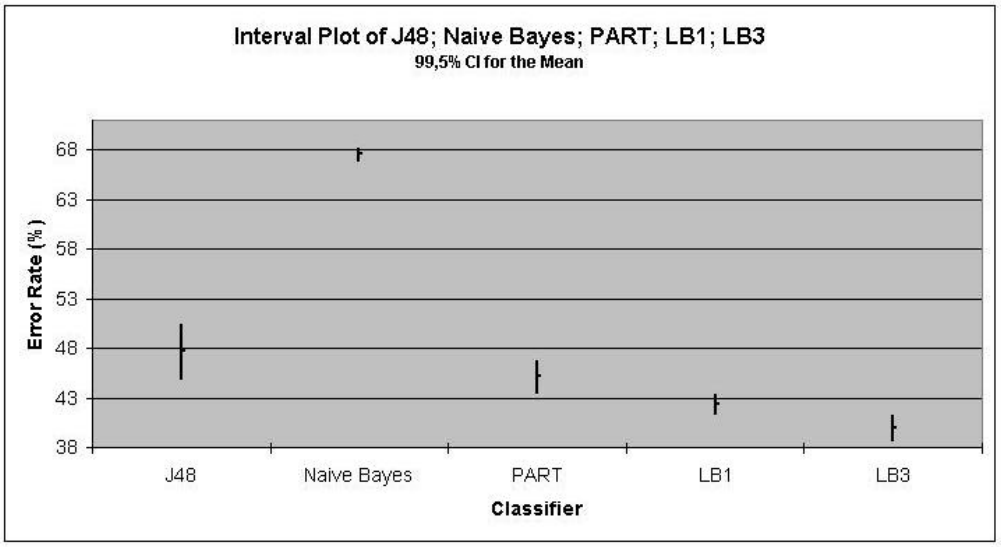
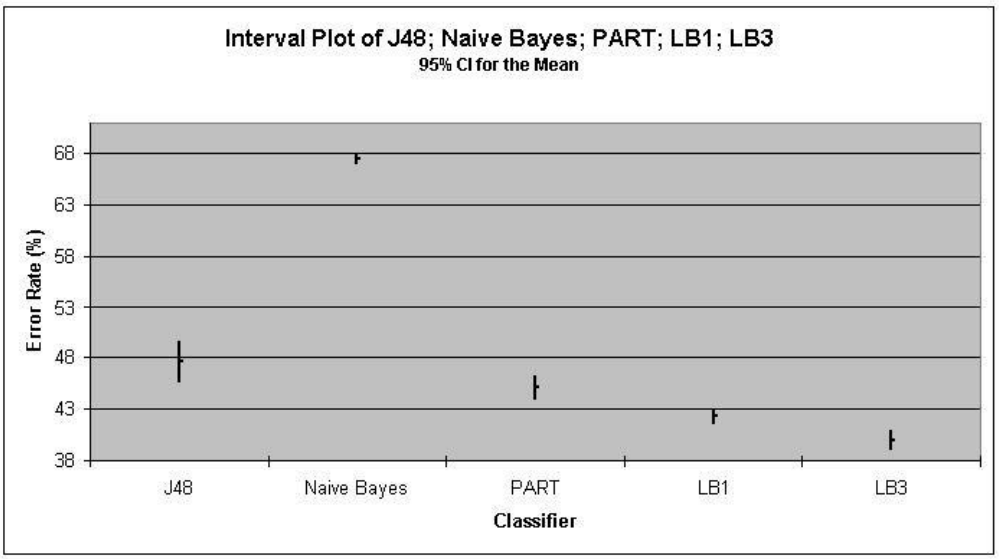


Figura 13 – Gráfico dos intervalos de confiança a 95% dos classificadores J48, Naive Bayes, PART e LB3.



Logo em seguida os classificadores foram combinados dois-a-dois: (J48,NB), (J48,PART), (J48,LB3), (NB,PART), (NB,LB3), (PART, LB3), (J48, LB1), (NB, LB1), (PART, LB1) e (LB3, LB1), onde NB é uma redução para o nome Naive Bayes. E para se

comparar as performances de cada combinação através de testes de hipóteses, foi criada uma variável aleatória que mediu a diferença entre as médias das taxas de erro entre os dois classificadores da combinação. Assim, para uma combinação entre dois classificadores "x" e "y", as diferenças das combinações "d(x,y)" podem ser vistas na Tabela 6 e na Tabela 7.

Tabela 6 – Dados referentes às diferenças das médias das taxas de erro entre os classificadores J48, Naive Bayes, PART e LB3.

Amostras	d(J48, NB) (%)	d(J48,PART) (%)	d(J48,LB3) (%)	d(NB,PART) (%)	d(NB,LB3) (%)	d(PART,LB3) (%)
v0	-22,9665	-0,2392	3,1100	22,7273	26,0766	3,3493
v1	-21,2919	2,1531	5,2632	23,4450	26,5550	3,1100
v2	-24,8804	-1,6746	2,3923	23,2057	27,2727	4,0670
v3	-20,8134	2,3923	9,0909	23,2057	29,9043	6,6986
v4	-18,6603	4,7847	9,0909	23,4450	27,7512	4,3062
v5	-19,1388	5,7416	8,1340	24,8804	27,2727	2,3923
v6	-17,2249	3,3493	8,8517	20,5742	26,0766	5,5024
v7	-17,7033	0,9569	10,0478	18,6603	27,7512	9,0909
v8	-15,7895	6,2201	11,2440	22,0096	27,0335	5,0239
v9	-20,3349	1,6746	9,8086	22,0096	30,1435	8,1340

Tabela 7 – Dados referentes às diferenças das médias das taxas de erro entre os classificadores entre os classificadores J48, Naive Bayes, PART, LB3 e LB1.

Amostras	d(LB1, J48) (%)	d(LB1, PART) (%)	d(LB1, LB3) (%)	d(LB1, NB) (%)
v0	-1,6746	-1,9139	1,4354	-24,6411
v1	-3,3493	-1,1962	1,9139	-24,6411
v2	0,4785	-1,1962	2,8708	-24,4019
v3	-3,5885	-1,1962	5,5024	-24,4019
v4	-7,1770	-2,3923	1,9139	-25,8373
v5	-7,1770	-1,4354	0,9569	-26,3158
v6	-7,6555	-4,3062	1,1962	-24,8804
v7	-5,9809	-5,0239	4,0670	-23,6842
v8	-11,4833	-5,2632	-0,2392	-27,2727
v9	-5,9809	-4,3062	3,8278	-26,3158

Foi necessário, também, identificar em que tipo de teste cada combinação se encaixaria: se em um teste paramétrico (*t* de *Student*) ou não-paramétrico (Mann-Whitney). Mas para isto, foram feitos nove testes de Kolmogorov-Smirnov com o objetivo de se testar a normalidade da distribuição da diferença das médias das taxas de erro entre os dois classificadores de uma dada combinação. Os resultados obtidos (vide Tabela 8 e Tabela 9) indicam que as diferenças $d_{J48,NB}$, $d_{J48,PART}$ e $d_{LB1,LB3}$ não seguem uma distribuição Normal pois os p-valores empíricos encontrados ($KS_{J48,NB}$=0,106, $KS_{J48,PART}$=0,122 e $KS_{LB1,LB3}$=0.149) foram abaixo do p-valor teórico (KS=0,150), enquanto que as diferenças $d_{NB,PART}$, $d_{NB,LB3}$, $d_{PART,LB3}$, $d_{LB1,J48}$, $d_{LB1,PART}$ e $d_{LB1,NB}$ seguem distribuições Normais com suas devidas médias e desvios padrões informados.

Tabela 8 – Resultados dos teste de Kolmogorov-Smirnov sobre a diferença das médias das taxas erro entre os classificadores J48, Naive Bayes, PART e LB3.

Amostras	d(J48, NB) (%)	d(J48,PART) (%)	d(J48,LB3) (%)	d(NB,PART) (%)	d(NB,LB3) (%)	d(PART,LB3) (%)
Média	-19,8800	2,5360	7,7030	22,4200	27,5800	5,1670
Desvio Padrão	2,7520	2,5470	3,0380	1,7440	1,4180	2,2080
KS	0,1060	0,1220	0,2560	0,1710	0,2030	0,1520

Tabela 9 – Resultados dos teste de Kolmogorov-Smirnov sobre a diferença das médias das taxas erro entre os classificadores J48, Naive Bayes, PART, LB3 e LB1.

Amostras	d(LB1, J48) (%)	d(LB1, PART) (%)	d(LB1, LB3) (%)	d(LB1, NB) (%)
Média	-5,359	-2,823	2,345	-25,24
Desvio Padrão	3,415	1,702	1,717	1,129
KS	0,151	0,203	0,149	0,225

A partir destas informações, a avaliação da performance dos classificadores foi obtida com o auxílio de mais dois outros testes: o de Mann-Whitney para as diferenças d(J48,NB), d(J48,PART) e d(LB1,LB3) e o teste *t* de *Student* para as demais diferenças. Desta forma, os testes serviram para verificar a existência de diferenças significativas a 1% (α_1=0,01) e a 5% (α_2=0,05) entre os classificadores de uma combinação.

As hipóteses destes testes foram obtidas com a ajuda dos gráficos de dispersão (figuras 14 a 23), listados abaixo. Neles, observam-se pares ordenados (x,y) que representam as taxas de erro de dois classificadores em uma combinação.

Nas figuras 14, 15 e 16, a maioria dos pares ordenados (x_{LB1},$y_{NaiveBayes}$), (x_{J48},$y_{NaiveBayes}$) e (x_{J48},y_{PART}) estão posicionados acima da reta y=x. Por isto, o teste emparelhado para cada combinação de classificadores possui as seguintes hipóteses nula (H_0) e alternativa (H_1):

$H_0: \mu_x = \mu_y$, ou $H_0: \mu_d = 0$,

$H_1: \mu_x > \mu_y$, ou $H_0: \mu_d < 0$,

onde μ_x é a média das taxas do classificador no eixo das abscissas, μ_y é a média das taxas do classificador no eixo das ordenadas e μ_d é a média das diferenças entre as taxas dos dois classificadores.

Figura 14 – Gráfico de dispersão entre as taxas de erro do Naive Bayes e do LB1.

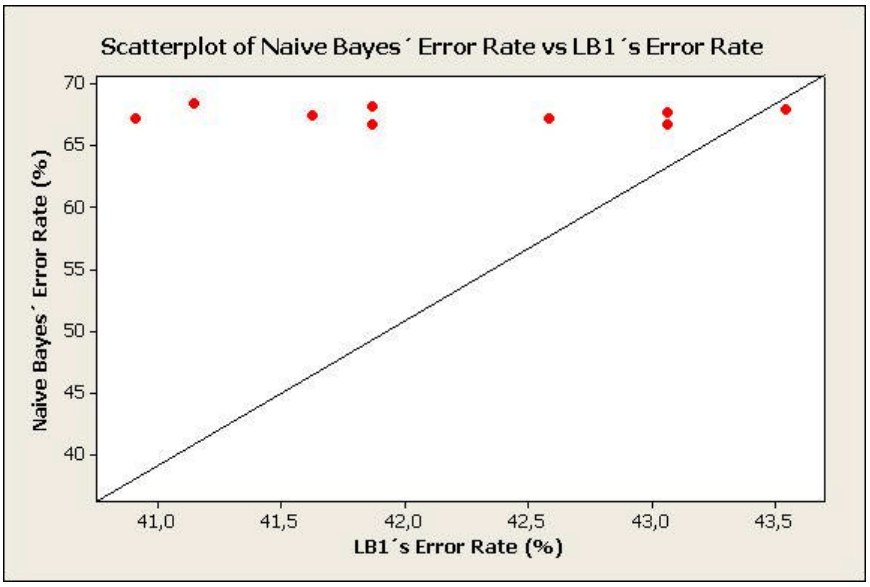


Figura 15 – Gráfico de dispersão entre as taxas de erro do Naive Bayes e do J48.

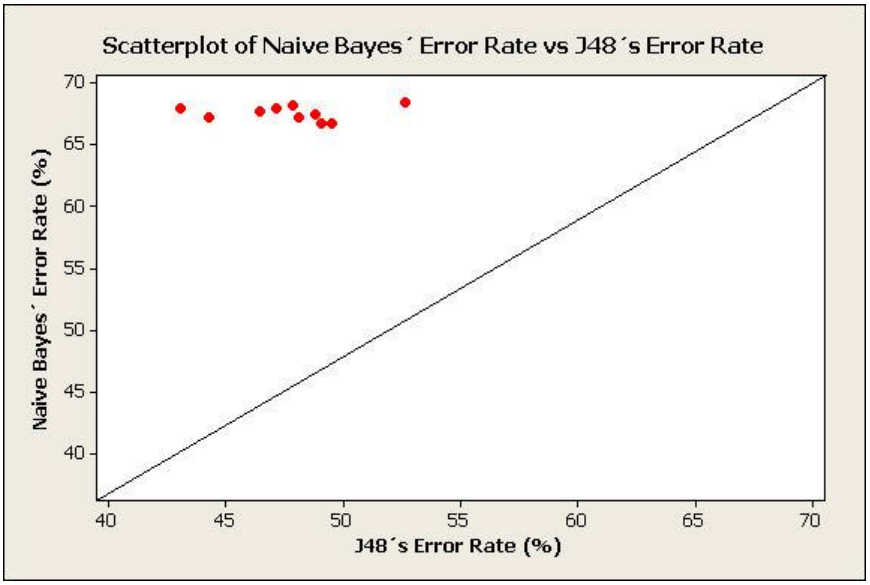
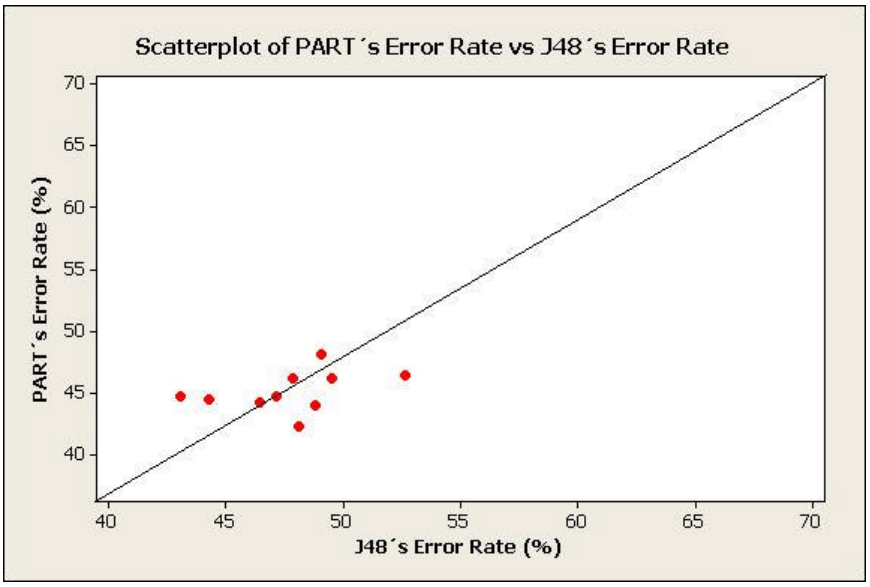


Figura 16 – Gráfico de dispersão entre as taxas de erro do PART e do J48.



Para os demais testes emparelhados, onde as médias das taxas de erro dos classificadores são representados pelos pares ordenados (x_{J48},y_{LB3}), $(x_{NaiveBayes},y_{PART},)$, $(x_{NaiveBayes},y_{LB3},)$, (x_{PART},y_{LB3}), (x_{LB1},y_{J48}), (x_{LB1},y_{PART}) e (x_{LB1},y_{LB3}) verifica-se que a maioria dos pares estão posicionados abaixo da reta y=x conforme as figuras 17 a 23. Portanto as hipóteses de cada um destes testes são:

$H_0: \mu_x = \mu_y$, ou $H_0: \mu_d = 0$,

$H_1: \mu_x < \mu_y$, ou $H_0: \mu_d > 0$,

Figura 17 – Gráfico de dispersão entre as taxas de erro do PART e do Naive Bayes.

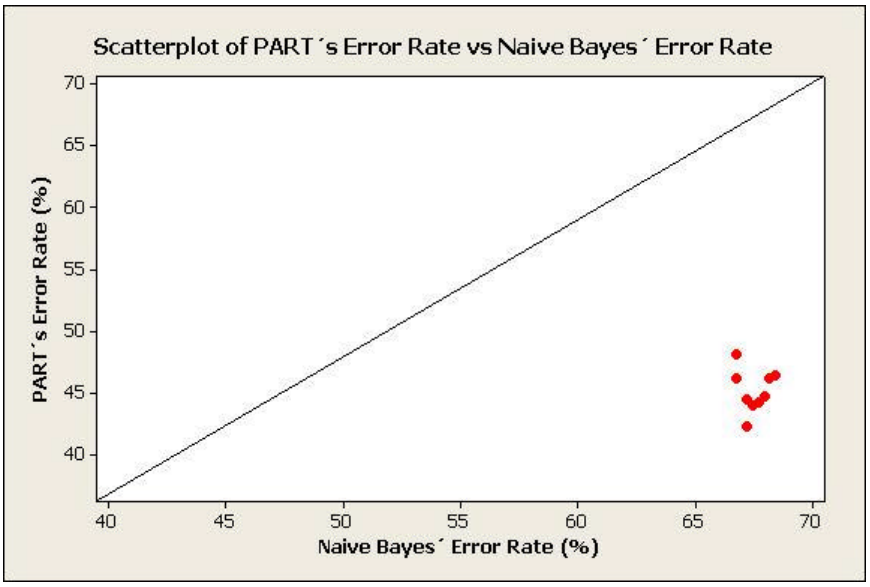


Figura 18 – Gráfico de dispersão entre as taxas de erro doLB3 e do Naive Bayes.

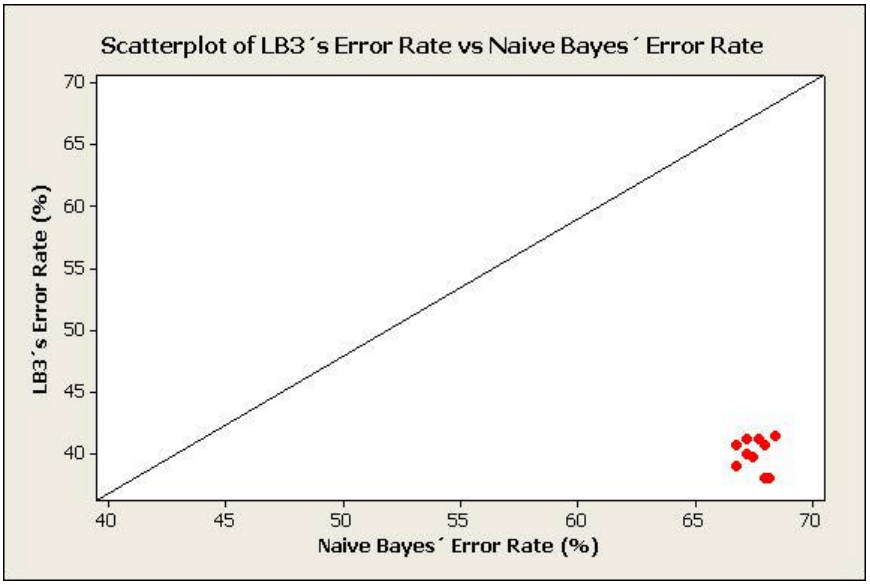


Figura 19 – Gráfico de dispersão entre as taxas de erro do LB3 e do PART.

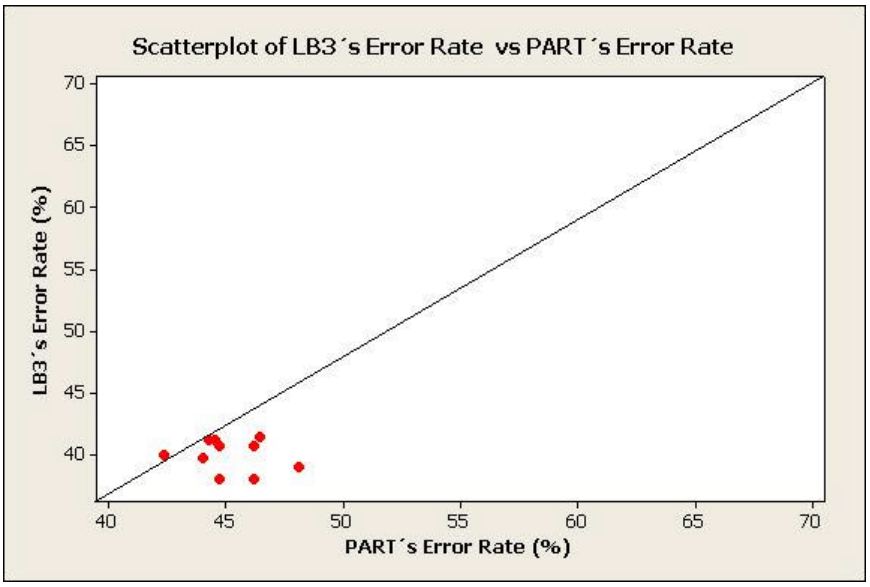


Figura 20 – Gráfico de dispersão entre as taxas de erro do J48 e do LB1.

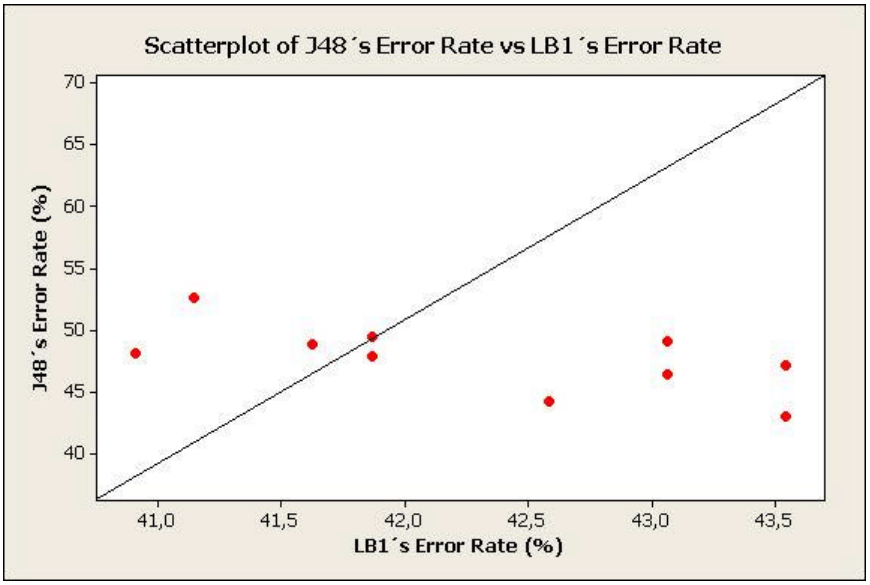


Figura 21 – Gráfico de dispersão entre as taxas de erro do PART e do LB1.

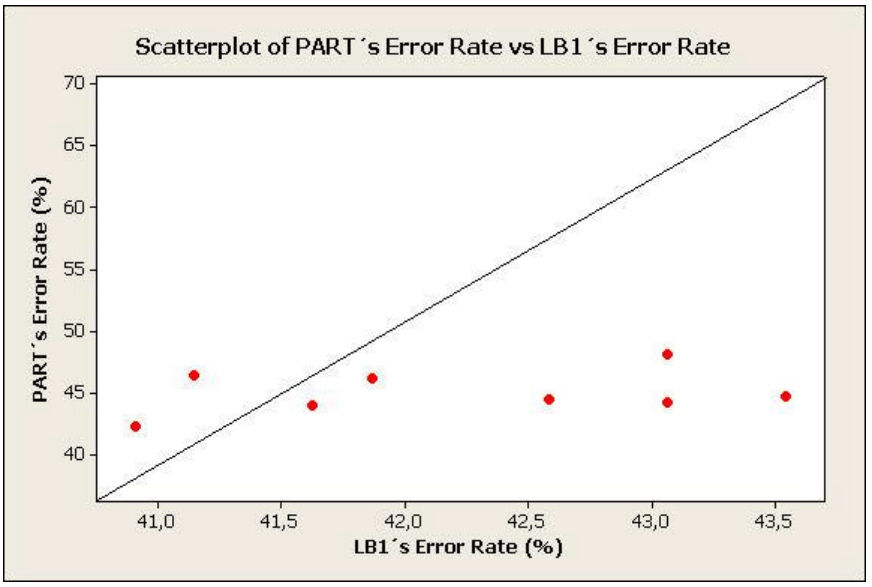
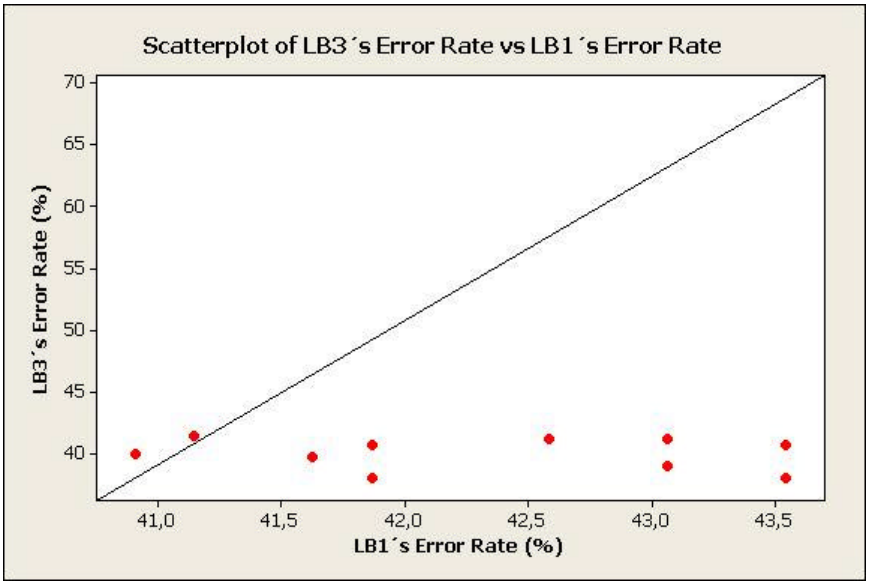


Figura 22 – Gráfico de dispersão entre as taxas de erro do LB3 e do LB1.



Abaixo estão os resultados para os testes emparelhados de Mann-Whitney.

Para as diferenças $d_{J48,NB}$ tendo α_1 como nível de significância (n.s.), a estatística foi $W_{J48,NB}$=55,0, o ponto estimado foi –19,617 e o intervalo de confiança foi (-23,685;-17,703). Por isto, H_0 pôde ser rejeitada. Verificou-se também que o teste de "$d_{J48,NB}$=0 versus $d_{J48,NB}$<0" é significante a um n.s. de 0,0001.

O mesmo teste para n.s. α_2 obteve os mesmos resultados com a exceção do intervalo de confiança que foi (-21,531;-18,420).

Para as diferenças $d_{J48,PART}$ tendo α_1 como n.s., a estatística foi $W_{J48,PART}$=134,5, o ponto estimado foi 2,6232 e o intervalo de confiança foi (-1,197;5,742). Como $W_{J48,PART}$ >105 não se pôde rejeitar H_0 para o teste "$d_{J48,PART}$ =0 versus $d_{J48,PART}$ <0".

O mesmo teste para n.s. α_2 obteve os mesmos resultados com a exceção do intervalo de confiança que foi (0,239;4,546).

Para as diferenças $d_{LB1,LB3}$ tendo α_1 como n.s., a estatística foi $W_{LB1,LB3}$=151,0, o ponto estimado foi 2,273 e o intervalo de confiança foi (0,478;4,068). Por isto, H_0 pôde ser rejeitada. Verificou-se também que o teste de "$d_{LB1,LB3}$=0 versus $d_{LB1,LB3}$>0" é significante a um n.s. de 0,0003.

O mesmo teste para n.s. α_2 obteve os mesmos resultados com a exceção do intervalo de confiança que foi (1,196;3,588).

Os resultados para os testes *t* de *Student* estão listados logo em seguida.

Para as diferenças $d_{NB,PART}$ os testes emparelhados rejeitaram H_0 em ambos níveis de significância pois:

$RC\alpha_1$ = (20,6244; +∞), com estimativa $T_{NaiveBayes,PART}$ = 40,65;

$RC\alpha_2$ = (21,4055; +∞) com estimativa $T_{NaiveBayes,PART}$ = 40,65.

Para as diferenças $d_{NB,LB3}$ os testes emparelhados rejeitaram H_0 em ambos níveis de significância pois:

$RC\alpha_1$ = (26,1267;+∞), com estimativa $T_{NB,LB3}$= 61,52;

$RC\alpha_2$ = (26,7619;+∞) com estimativa $T_{NB,LB3}$= 61,52.

Para as diferenças $d_{PART,LB3}$ os testes emparelhados rejeitaram H_0 em ambos níveis de significância pois:

$RC\alpha_1 = (2{,}8986; +\infty)$, com estimativa $T_{PART,LB3} = 7{,}40$;

$RC\alpha_2 = (3{,}8877; +\infty)$ com estimativa $T_{PART,LB3} = 7{,}40$.

Para as diferenças $d_{LB1,J48}$ os testes emparelhados rejeitaram H_0 em ambos níveis de significância pois:

$RC\alpha_1 = (-8{,}8688; +\infty)$, com estimativa $T_{LB1,J48} = -4{,}96$;

$RC\alpha_2 = (-7{,}3387; +\infty)$ com estimativa $T_{LB1,J48} = -4{,}96$.

Para as diferenças $d_{LB1,NB}$ os testes emparelhados rejeitaram H_0 em ambos níveis de significância pois:

$RC\alpha_1 = (-\infty; -24{,}0788)$, com estimativa $T_{LB1,NB} = -70{,}68$;

$RC\alpha_2 = (-\infty; -24{,}5847)$ com estimativa $T_{LB1,NB} = -70{,}68$.

Para as diferenças $d_{LB1,PART}$ os testes emparelhados não rejeitaram H_0 em ambos níveis de significância pois:

$RC\alpha_1 = (-4{,}5721; +\infty)$, com estimativa $T_{LB1,PART} = -5{,}24$;

$RC\alpha_2 = (-3{,}8096; +\infty)$ com estimativa $T_{LB1,PART} = -5{,}24$.

CAPÍTULO VI

6 CONCLUSÃO E TRABALHOS FUTUROS

O objetivo deste trabalho foi comparar a performance de quatro modelos de Aprendizagem de Máquina (Árvore de Decisão, Regras de Decisão, Classificador Bayesiano Ingênuo e K-Vizinhos) através de implementações já realizadas no software Weka (J48, PART, Naive Bayes e LBk, respectivamente). As análises do LBk foram feitas com as seguintes valorações: k=3 e k=1.

Para tanto, utilizou-se dados de medições de expressão gênica ***S. cerevisiae*** obtidos por Cho *et al.* (1998) e dados classificatórios sobre o mesmo organismo obtidos por Filho (2003) o que permitiu uma abordagem supervisionada.

Os dados de Filho (2003) foram pré-processados para identificar o subconjunto a ser utilizado. Logo em seguida, os dados foram submetidos ao Weka que gerou dez amostras da base de dados. Seguindo a sugestão de Witten & Frank (2000), para garantir uma maior representatividade na construção dos modelos, cada amostra foi submetida a dez repetições de Validação Cruzada em dez pastas. Cada repetição obteve uma média da taxa de erro sobre os modelos gerados. Por último, uma comparação entre diferentes modelos de aprendizagem foi feita através de análises estatísticas através de testes pareados sobre as diferenças, entre os classificadores, das médias de taxas de erro obtidas dos classificadores após a criação dos seus modelos.

Verificou-se que estas diferenças podem ser consideradas nulas a níveis de significância 0,01 e 0,05 apenas para os seguintes pares de classificadores: (J48, PART) e (LB1, PART). Assim, pode-se substituir o modelo gerado pelo J48 pelo do PART e vice-versa além de se poder substituir também o do LB1 pelo do PART e vice-versa obtendo-se modelos que geram uma média de taxas de erro semelhantes.

Porém, não se pode dizer o mesmo para os modelos provenientes do J48 e do LB1, pois o teste *t* de *Student* rejeitou a hipótese nula para ambos os níveis de significância. Analogamente, as demais combinações possíveis entre os classificadores restantes rejeitaram a hipótese nula para os dois níveis de significância.

A partir dos resultados obtidos no escopo deste trabalho, percebe-se que este estudo pode ser estendido buscando-se aplicar diferentes valorações para os parâmetros utilizados no Weka de forma que os classificadores apresentem melhores resultados. Outra forma possível seria utilizar-se de uma nova base de dados.

REFERÊNCIAS BIBLIOGRÁFICAS

BUSSAB, W. O.; MORETTIN, P. A. **Estatística Básica**. Recife, Pernambuco, Brasil: Editora Saraiva, 2003.

CHO, R.; CAMPBELL, M.; WINZELER, E.; STEINMETZ, L.; CONWAY, A.; WODICKA, L.; WOLFSBERG, T.; GABRIELIAN, A.; LANDSMAN, D.; LOCKHART, J.; DAVIS, W. **A genomewide transcriptional analysis of the mitotic cell cycle**, Molecular Cell, v.2, jul. 1998. p.65-73.

CHU, S.; DELRISI, J.; EISEN, M.; MULHOLLAND, J.; BOTSTEIN, D.; BROWN, P.O.; HERSKOWITZ I. **The Transcriptional Program of Sporulation in Budding Yeast**. Science, v.282, out. 1998. p.699-705.

CYGD. Comprehensive Yeast Genome Database. **Munich Information Center for Protein Sequences**. Disponível em: <http://mips.gsf.de/genre/proj/yeast/index.jsp> Consultado em: 20 julho 2004.

DERISI, J. L.; IYER V. R.; BROWN P. O. **Exploring the metabolic and genetic control of gene expression on a genomic scale**. Science, v. 278, out. 1997. p.680-686.

EISEN, M. B.; SPELLMAN, P. T.; BROWN, P. O.; BOTSTEIN, D. **Cluster analysis and display of genome-wide expression patterns**. United States of America: Proc. of National Academy of Sciences, v. 95, dez. 1998. p.14863-14868.

FARAH, S. B. **DNA: Segredos & Mistérios**. São Paulo, Brasil: Sarvier, 1997. p.255-272.

FILHO, I. G. Costa. **Comparative Analysis of Clustering Methods for Gene Expression Data**. Recife: UFPE, 2003. Dissertação de Mestrado.

HAND, David J.; MANNILA, Heikki; SMYTH, Padhraic. **Principles of Data Mining** - Adaptive Computation and Machine Learning. New Jersey, United States of America: Bradford Books, 2001. p.1-9.

HEYER, L. J.; KRUGLYAK, S.; YOOSEPH, S. **Exploring expression data:** identi?cation and analysis of coexpressed genes. Genome Research, v.9, 1999. p.1106-1115.

KALAPANIDAS, Elias; AVOURIS, Nikolaos; CRACIUN, Marian; NEAGU, Daniel. **Machine Learning algorithms:** a study on noise sensitivity. Thessaloniki, Greece: 1st Balcan Conference in Informatics, 2003.

KELLER, Frank. Introduction to Machine Learning. - Connectionist and Statistical Language Processing. **School of Informatics**. Disponível em:

<http://homepages.inf.ed.ac.uk/keller/teaching/connectionism/lecture8_4up.pdf> Consultado em: 20 julho 2004.

MITCHELL, Tom M. **Machine Learning**. New York, United States of America: McGraw-Hill, 1997.

RUSSEL, Stuart J.; NORVIG, Peter. **Artificial Inteligence** – A Modern Approach. New Jersey, United States of America: Prentice-Hall, 1995. p.523-647.

SPELLMAN, P. T.; SHERLOCK, G.; ZHANG, M. Q.; IYER, V. R.; ANDERS, K.; EISEN, M. B.; BROWN, P. O.; BOTSTEIN, D.; FUTCHER, B. **Comprehensive Identi?cation of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hibridization**. Molecular Biology of the Cell, v.9, dez. 1998. p. 3273-3297.

WITTEN, Ian H.; FRANK, Eibe. **Data Mining** – Practical Machine Learning Tools and Techniques with Java Implementations. San Francisco, CA, United States of America: Morgan Kaufmann Publishers, 2000.

APÊNDICE A – FORMATO ARFF

ARFF é o acrônimo para *Attribute-Relation File Format*.que representa a formatação de arquivo aceita pelo Weka (mais informações no "APÊNDICE A – Formato ARFF").

Qualquer linha do arquivo que iniciar por %, representa uma linha comentada, ou seja, nenhum processamento ocorrerá sobre os dados desta linha. A primeira linha sem comentário deve possuir no seu início `@relation` seguido do nome da relação a ser analisada. Logo depois, deve haver um bloco de linhas iniciadas por `@attribute`, seguida do nome do atributo e do nome `numeric` ou `real` se for um atributo numérico, ou então de um conjunto de valores nominais separados por vírgulas e delimitados por parênteses. Após este bloco de linhas, deve vir uma iniciada por `@data` indicando o início das instâncias. E em cada uma das linhas seguintes, deve haver valorações para cada um dos atributos anteriormente declarados.

Um exemplo de um arquivo neste formato é possível ser visualizado na Figura 23.

```
%Arquivo no formato ARFF
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
```

Figura 23 – Exemplo de um arquivo no formato ARFF.

APÊNDICE B – WEKA

"Weka Machine Learning Project" é uma coleção de algoritmos para tarefas de mineração de dados desenvolvido na linguagem Java pelo Departamento de Ciências da Computação da Universidade de Waikato, na Nova Zelândia. Estes algoritmos podem ser aplicados diretamente sobre os dados analisados ou ser chamados a partir de um código Java qualquer bastando que os dados estejam no formato ARFF (ver APÊNDICE A – Formato ARFF).

O Weka, *Waikato Environment for Knowledge Analysis*, segue a Licença Pública Geral do GNU para o código aberto e é composto pelas seguintes ferramentas: *Weka Simple CLI*, *Weka Explorer*, *Weka Experimenter* e *Weka Knowledge Flow*. Com elas é possível fazer o pré-processamento, a classificação, a regressão, a clusterização, a visualização de gráficos e a geração de regras de associação dos dados, além do desenvolvimento de novos modelos de aprendizagem.

No site http://www.cs.waikato.ac.nz/~ml/, pode-se fazer o *download* dos arquivos necessários para a sua instalação tanto no Microsoft Windows quanto no Linux. É possível também, acessar um F.A.Q. (*Frequent Asked Question*) e um *newsgroup* onde se encontram respostas para perguntas sobre a instalação, qual o melhor modelo de aprendizagem de máquina utilizar para um conjuntos de dados em particular, entre outras.

Outra boa referência sobre o Weka é o Witten & Frank (2000), pois além de possuir explicações de conceitos de mineração de dados, esta fonte ensina os passos básicos para se manipulá-lo.