UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

Uma comparação entre classificadores para predição da classe de cor a partir de dados estruturais em proteínas fluorescentes

Roger Sá da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Prof. Dr. Adriano V. Werhli
Coorientadora: Prof[a]. Dr[a]. Karina S. Machado

Rio Grande, 2016

Ficha catalográfica

S586c Silva, Roger Sá da.
Uma comparação entre classificadores para predição da classe de cor a partir de dados estruturais em proteínas fluorescentes / Roger Sá da Silva. – 2016.
120 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-graduação em Engenharia de Computação, Rio Grande/RS, 2016.
Orientador: Dr. Adriano Velasque Werhli.
Coorientadora: Drª. Karina dos Santos Machado.

1. Bioinformática 2. Mineração de dados 3. Proteínas fluorescentes I. Werhli, Adriano Velasque II. Machado, Karina dos Santos II. Título.

CDU 004.6

Catalogação na Fonte: Bibliotecário Me. João Paulo Borges da Silveira CRB 10/2130

UNIVERSIDADE FEDERAL DO RIO GRANDE

Centro de Ciências Computacionais

Programa da Pós-Graduação em Computação

Curso de Mestrado em Engenharia de Computação

DISSERTAÇÃO DE MESTRADO

Uma comparação entre classificadores para predição da classe de cor a partir de dados estruturais em proteínas fluorescentes

Roger Sá da Silva

Banca examinadora:

Prof. Dr. Márcio Dorn

Prof. Dr. Eduardo Nunes Borges

Prof. Dr. Adriano Velasque Werhli
Orientador

AGRADECIMENTOS

Agradeço aos meus pais, à Laura, aos meus familiares e amigos, pelo incentivo e pelo apoio e suporte constantes.

Aos professores Adriano e Karina pela orientação, compreensão, dedicação e importantes sugestões na elaboração deste trabalho. Professores sempre disponíveis, participativos e comprometidos com os alunos e, em especial, com o grupo de pesquisa de Biologia Computacional.

Ao professor e pesquisador Luis Fernando Marins, à coordenadora do projeto Peixes Transgênicos Fluorescentes Daniela Volcan Almeida e à bióloga Natalia Ossa Hernández, pela disponibilidade e pelo compartilhamento de dados e ensinamentos sobre biologia molecular e proteínas fluorescentes.

Aos colegas do grupo de Biologia Computacional, em especial ao Alex Camargo e ao Vinícius Seus, pela convivência, troca de conhecimentos e dicas na elaboração deste trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela concessão do apoio financeiro.

A todos que de alguma forma contribuíram para a conclusão deste trabalho.

RESUMO

SILVA, Roger Sá da. **Uma comparação entre classificadores para predição da classe de cor a partir de dados estruturais em proteínas fluorescentes**. 2016. 119 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

Proteínas Fluorescentes são importantes ferramentas em pesquisas de Biologia Molecular e possuem grande valor comercial na produção de peixes transgênicos fluorescentes. De modo geral, a criação de variantes de cor destas proteínas ocorre por alterações estruturais na macromolécula, ocasionadas por mutações na sequência de aminoácidos. Porém, relacionar de forma exata dados estruturais e sequenciais com a definição de cor de emissão de proteínas fluorescentes ainda necessita de mais estudos. Neste contexto, a aplicação do processo de descoberta de conhecimento em bases de dados se apresenta como uma possibilidade de obtenção de conhecimento sobre essa relação da sequência/estrutura e a cor de emissão. Dessa forma, é realizado neste trabalho a comparação entre três classificadores (baseados em Árvore de Decisão, Redes Neurais Artificiais e Máquinas de Vetores de Suporte) com o intuito de investigar a performance deles na predição da classe de cor de proteínas fluorescentes a partir de seus dados estruturais no âmbito do projeto Peixes Transgênicos Fluorescentes. Para tanto, uma ferramenta web é desenvolvida para o armazenamento, organização e preparação dos dados estruturais utilizados no treinamento dos classificadores. Ao final, um processo de comparação quantitativa e qualitativa é realizado sobre métricas de desempenho e capacidades de cada classificador, culminando na escolha do classificador baseado em Árvore de Decisão como o mais adequado na tarefa de predição da classe de cor de proteínas fluorescentes.

Palavras-chave: Bioinformática, Mineração de Dados, Proteínas fluorescentes.

ABSTRACT

SILVA, Roger Sá da. **A comparison of classifiers for predicting color class from structural data on fluorescent proteins**. 2016. 119 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

Fluorescent proteins are important tools in molecular biology research and have great commercial value in production of fluorescent transgenic fishes. In general, the creation of color variants of these proteins occurs by structural changes in the macromolecule caused by mutations in amino acid sequence. However, to relate accurately structural and sequence data of fluorescent proteins with its emission color still needs further study. In this context, the application of knowledge discovery in databases process presents a possibility of obtaining knowledge on this relationship of the sequence / structure and emission color. Thus, in this work it is carried out a comparison between classifiers (based on Decision Tree, Artificial Neural Networks and Support Vector Machines) in order to investigate their performance in predicting the class color of fluorescent proteins from their structural data, in the context of Fluorescent Transgenic Fishes project. Therefore, an web tool is designed for the storage, organization and preparation of structural data used in the classifiers training. At the end, a quantitative and qualitative comparison process is carried out on performance metrics and capabilities of each classifier, culminating in the selection of the classifier based on Decision Tree as the most appropriate for the task of predicting the fluorescent proteins color class.

Keywords: Bioinformatics, Data Mining, Fluorescent Proteins.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AD	Árvore de Decisão
DsRed	*Discosoma Red fluorescent protein*
FBR	Função de Base Radial
FK	*Foreign Key*
GFP	*Green Fluorescent Protein*
GH	*Growth Hormone*
IA	Inteligência Artificial
KDD	*Knowledge Discovery in Databases*
LBM	Laboratório de Biologia Molecular
MD	Mineração de Dados
MDL	*Minimum Description Length*
MLP	*Multi Layer Perceptron*
MVC	Modelo-Visão-Controle
NFL	*No Free Lunch*
PK	*Primary Key*
RNA	Rede Neural Artificial
SGBD	Sistema Gerenciador de Banco de Dados
SQL	*Structured Query Language*
SVM	*Support Vector Machine*
TDIDT	*Top-Down Induction of Decision Tree*
WEKA	*Waikato Environment for Knowledge Analysis*

SUMÁRIO

1 INTRODUÇÃO

Devido ao potencial brasileiro na produção de peixes ornamentais e da grande demanda internacional por estas espécies e do alto valor comercial deste mercado, a piscicultura ornamental é uma área em pleno desenvolvimento, segundo (FIGUEIREDO et al., 2007) e (FIGUEIREDO et al., 2012) . Porém, problemas como o extrativismo e o surgimento de poucas novas espécies são enfrentados neste mercado, como mostram (MAGALHÃES, 2007) e (CASIMIRO et al., 2010).

Neste cenário, projetos de pesquisa têm surgido com a intenção de utilizar novas técnicas na produção de peixes ornamentais, apresentando assim atrativos diferenciados ao mercado, como a emissão de fluorescência (GONG et al., 2003). É o caso do projeto "Peixes Transgênicos Fluorescentes: um novo campo para a piscicultura ornamental no Brasil", em desenvolvimento no Laboratório de Biologia Molecular da Universidade Federal do Rio Grande - FURG.

O projeto do Laboratório de Biologia Molecular (LBM) pretende produzir novas variantes de proteínas fluorescentes para que tais proteínas apresentem melhores características necessárias à piscicultura ornamental, tais como espectro de emissão de luz mais próximo do visível e diferentes cores (ALMEIDA, 2014). É importante ressaltar que a produção das proteínas fluorescentes pelo LBM é realizada através da mutagênese randômica, ou seja, geração de mutações aleatórias, o que se traduz em desconhecimento sobre o resultado das mutações, incluindo-se o comprimento de onda a ser emitido pelas novas mutantes (LABROU, 2010).

Estudos já demonstraram que a estrutura tridimensional da proteína fluorescente, em especial devido à interação criada com o grupo cromóforo (grupo de átomos responsável pela cor de um composto), tem relação na definição do comprimento de onda que a proteína fluorescente emite (CHUDAKOV et al., 2010). Mais recentemente, sabe-se também que a presença de determinados aminoácidos espacialmente próximos ao grupo cromóforo influenciam na definição da cor emitida por proteínas fluorescentes (LABROU, 2010).

1.1 Objetivo Geral

Dessa forma, o presente trabalho, como parte integrante do projeto "Peixes Transgênicos Fluorescentes: um novo campo para a piscicultura ornamental no Brasil", tem por objetivo aplicar análises e técnicas de bioinformática com o intuito de contribuir no entendimento de como as estruturas moleculares tridimensionais das proteínas fluorescentes se relacionam com o comprimento de onda emitido por elas.

Para alcançar o objetivo, propõe-se organizar os dados relativos às proteínas fluorescentes em um sistema de informação e, posteriormente, aplicar e comparar técnicas de descoberta de conhecimento nestes dados, através de algoritmos já implementados no software WEKA (BOUCKAERT et al., 2015).

1.2 Objetivos Específicos

De forma mais específica, o objetivo do trabalho é comparar métodos de classificação a fim de se investigar a performance de classificadores no problema da predição da classe de cor de proteínas fluorescentes a partir de sua estrutura molecular tridimensional. É importante destacar que a comparação entre os classificadores é baseada nos resultados oriundos de uma mesma base de dados, que por usa vez é parte do sistema de informação desenvolvido exclusivamente para este trabalho.

Além do mais, como objetivo específico também espera-se que seja possível a utilização de um classificador como ferramenta de apoio à pesquisa e como fonte de novos conhecimentos sobre os dados, sendo capaz de contribuir no desenvolvimento de novas variantes de proteínas fluorescentes no âmbito do projeto Peixes Transgênicos Fluorescentes. A intenção é realizar a predição *in silico* da classe de cor a partir de dados estruturais das proteínas fluorescentes, para em seguida sugerir mutações dirigidas no desenvolvimento das novas variantes.

1.3 Trabalhos Relacionados

Nesta seção são abordados trabalhos e estudos que relacionam métodos de descoberta de conhecimento com a predição de características relacionadas à fluorescência em proteínas. É importante destacar que todos os trabalhos e publicações apresentam um forte cunho bioquímico, portanto, os resultados computacionais não foram explorados nem demonstrados de forma adequada nestes trabalhos. Isso explica a maneira como são apresentados ao longo desta seção.

O primeiro trabalho abordado trata da predição computacional da absorção máxima para os cromóforos da proteína verde fluorescente (TIMERGHAZIN et al., 2008). A produção de novas variantes de proteínas verdes fluorescentes é realizada de forma empírica, através de modificações genéticas na estrutura do cromóforo e/ou de seu ambi-

ente, com a intenção de encontrar variantes com novas propriedades fotofísicas. Ainda de acordo com os autores, o processo de identificação de variantes melhoradas pode ser imensamente facilitado se guiado por métodos de descoberta de conhecimento. Nesta direção, o trabalho analisa métodos de regressão para predição da absorção máxima em cromóforos de 10 proteínas verde fluorescentes experimentais análogas, tendo como base, por exemplo, valores de energia de excitação e forças de oscilação. De acordo com os autores, foi possível concluir, após a comparação entre os resultados experimentais e computacionais, que existe boa correlação linear entre o valor calculado e o experimental em relação a energia de excitação, conforme o método de regressão utilizado.

Dois outros trabalhos pesquisados tratam da correlação entre parâmetros da fluorescência do tipo de aminoácido Triptofano com parâmetros e propriedades estruturais da proteína e do ambiente onde este tipo de aminoácido encontra-se na estrutura tridimensional (RESHETNYAK; KOSHEVNIK; BURSTEIN, 2001) (HIXON; RESHETNYAK, 2009). Conforme (RESHETNYAK; KOSHEVNIK; BURSTEIN, 2001), comprovou-se em trabalhos anteriores a existência de cinco classes discretas para emissão de fluorescência do Triptofano em proteínas. A diferença nas propriedades fluorescentes dos Triptofanos destas cinco classes reflete as diferenças em interações do aminoácido Triptofano com o ambiente no qual está inserido na estrutura da proteína. No trabalho em questão, a aplicação de métodos de clusterização para um conjunto de parâmetros do ambiente de 137 aminoácidos Triptofano de 48 proteínas permitiu, dentre outras questões, encontrar a existência de correlação entre fatores espectrais do aminoácido com fatores físico-estruturais do ambiente, evidenciando diferenças entre tais parâmetros para as diferentes classes do aminoácido Triptofano. No trabalho (HIXON; RESHETNYAK, 2009), algoritmos de aprendizado supervisionado e não-supervisionado foram aplicados a dados relacionados a fluorescência do aminoácido Triptofano e a propriedades estruturais das proteínas, evidenciando, como no trabalho anterior, a correlação entre ambos os parâmetros.

Em mais um trabalho pesquisado, os autores abordaram a aplicação de métodos de descoberta de conhecimento para examinar a estabilidade de proteínas na ocorrência de dupla mutação (HUANG et al., 2014). A predição correta de mudanças na estabilidade da proteína na ocorrência de mutações é muito útil para aumentar a eficácia em experimentos de estudos biológicos. Os autores ainda esclarecem que o objetivo do estudo foi introduzir efetivamente técnicas de mineração de dados para investigar as mudanças na estabilidade de proteínas quando duplas mutações são realizadas. Através de um conjunto de dados não redundantes aplicou-se algumas técnicas de mineração de dados como classificação (com um algoritmo baseado em árvores de decisão) e regras de associação, para a aquisição de conhecimento sobre a mudança na estabilidade de proteínas. Comparando-se com resultados experimentais, concluem os autores, os métodos utilizados no trabalho podem servir como uma ferramenta eficiente para o entendimento sobre a predição em

mudanças na estabilidade em proteínas que apresentam duplas mutações.

Além dos citados anteriormente, existem mais dois trabalhos relacionados entre si, que abordam a predição de propriedades espectrais das proteínas verdes fluorescentes (NANTASENAMAT et al., 2007) (NANTASENAMAT et al., 2013). No trabalho (NANTASENAMAT et al., 2007), a predição da excitação e emissão máxima de cromóforos da proteína verde fluorescente foi realizada por um estudo da relação quantitativa das propriedades estruturais destas proteínas. Para realizar a predição, uma rede neural artificial implementando o algoritmo de retropropagação foi utilizada sobre um conjunto de dezenove cromóforos de variantes da proteína verde fluorescente e vinte e nove cromóforos sintéticos. A abordagem computacional proposta conseguiu predizer a excitação e emissão máxima com um coeficiente de correlação superior a 0,90, o que demonstra um bom resultado. Os autores ainda destacam que a utilização de redes neurais artificiais é mais indicada na predição de propriedades espectrais da proteína verde fluorescente do que, métodos de regressão linear múltipla, tradicionalmente utilizado, devido aos melhores resultados apresentados ao se comparar ambas técnicas. Em (NANTASENAMAT et al., 2013), por sua vez, a nova abordagem para predição de propriedades espectrais de proteínas fluorescentes utilizou como método de predição a máquina de vetores de suporte. Com este novo método, os resultados obtidos foram ainda melhores, pois atingiu-se coeficientes de correlação superiores a 0,95. Ambos os trabalhos concluem que os dois métodos de predição analisados apresentam potencial para acelerar os processos de produção de novas variantes das proteínas fluorescentes, devido aos bons resultados preditos na comparação com os valores experimentais de absorção e emissão máxima para os cromóforos analisados. Por fim, cabe destacar que devido à características intrínsecas aos algoritmos utilizados, estes trabalhos não apresentam quais variáveis ou parâmetros em análise são determinantes na relação com a emissão de cor das proteínas fluorescentes.

1.4 Motivação e Justificativa

Após abordar os trabalhos relacionados à temática da predição de propriedades de proteínas fluorescentes e o vínculo deste trabalho como parte integrante do projeto Peixes Transgênicos Fluorescentes, é possível explicar a motivação para algumas escolhas tratadas nos objetivos.

Primeiramente, é necessário destacar a motivação por comparar a performance de métodos de classificação. A opção por realizar um processo de comparação entre métodos de classificação está diretamente vinculada ao fato deste trabalho ser parte do projeto Peixes Transgênicos Fluorescentes. No projeto, pretende-se utilizar os classificadores como ferramenta de apoio à pesquisa, através da extração de conhecimento sobre os dados existentes e permitindo a predição da classe de cor a partir de dados estruturais das proteínas fluorescentes estudadas (ALMEIDA, 2014). Sob este ponto de vista, a comparação da

performance de classificadores é justificada, pois permite evidenciar o classificador mais adequado para utilização no contexto do projeto de pesquisa.

Outro aspecto que precisa ser abordado está relacionado à motivação pela escolha de classificadores baseados em Árvores de Decisão, em Redes Neurais Artificiais e em Máquinas de Vetores de Suporte para realizar o processo de comparação aqui proposto. Esta escolha justifica-se, inicialmente, pelo extenso uso e bons resultados de classificadores baseados nestas técnicas em trabalhos de Bioinformática (FRANK et al., 2004) (VERLI, 2014).

Porém, o fato que justifica mais fortemente a escolha destes três classificadores para comparação de sua performance foi a utilização deles em trabalhos relacionados à predição de propriedades de proteínas fluorescentes. Os autores dos trabalhos destacam que os classificadores baseados nas três técnicas apresentaram resultados satisfatórios, porém estes métodos de classificação foram analisados separadamente em estudos distintos e com bases de dados também distintas, mesmo que relacionadas à proteínas fluorescentes.

Dessa forma, um dos diferenciais do presente trabalho reside no fato de que, para um mesmo conjunto de dados estruturais de proteínas fluorescentes, as três técnicas de classificação são aplicadas e as performances dos classificadores construídos são avaliadas comparativamente, possibilitando a escolha do classificador mais adequado para uso como ferramenta de apoio à pesquisa no âmbito do projeto Peixes Transgênicos Fluorescentes.

1.5 Organização do Texto

Este trabalho está organizado da seguinte forma:

O Capítulo 2 aborda conceitos biológicos correlatos ao presente trabalho. O Capítulo 3 trata do processo de Descoberta de Conhecimento em Bases de Dados (KDD), através das cinco etapas que o compõe e das tarefas que este processo permite executar. A etapa de Mineração de Dados do processo de KDD é melhor detalhada no Capítulo 4. No Capítulo 5, os métodos de avaliação e as métricas de desempenho são tratados. Além disso, aborda-se a comparação de classificadores a partir destas métricas. O Capítulo 6 aborda as quatro etapas da metodologia do trabalho: o desenvolvimento de um sistema web para organização e preparação dos dados estruturais das proteínas fluorescentes, a execução do processo de descoberta de conhecimento sobre estes dados, o uso dos classificadores para a classificação de uma nova proteína fluorescente e a posterior comparação de performance dos classificadores construídos. As ferramentas e tecnologias utilizadas para a realização das etapas listadas na metodologia deste trabalho são abordadas no Capítulo 7. No Capítulo 8 estão listados todos os resultados obtidos na construção dos classificadores, nos métodos de avaliação, na comparação das métricas de desempenho e

no uso dos classificadores para a classificação de novos exemplos. A análise argumentativa e discussão sobre os resultados obtidos, retomando conceitos e definições tratados ao longo do texto, a fim de cumprir o objetivo do trabalho, estão abordados no Capítulo 9. Ao fim, o Capítulo 10 traz conclusões sobre os resultados já discutidos e aborda possibilidades de trabalhos futuros.

2 ASPECTOS BIOLÓGICOS

As proteínas fluorescentes e a relação entre sua estrutura tridimensional e a definição de cor através do cromóforo são importantes conceitos para o correto entendimento da proposta do trabalho. Além disso, a piscicultura, mais especificamente a piscicultura ornamental, seu atual cenário e os problemas enfrentados são tratados como contexto para o projeto Peixes Transgênicos Fluorescentes, no qual este trabalho está envolvido.

2.1 Piscicultura

Aquicultura é o processo de produção em cativeiro (condições controladas) de organismos que vivem em ambiente predominantemente aquático. É uma atividade praticada pelo ser humano há milhares de anos. Existem registros de que os chineses já tinham conhecimentos sobre estas técnicas há muitos séculos e de que os egípcios criavam espécies há cerca de quatro mil anos. Assim, a aquicultura envolve a produção de peixes, camarões, rãs, ostras e outras espécies (FURTADO, 1995).

Quando se fala especificamente em produção de peixes, essa atividade caracteriza-se como um subtipo da aquicultura denominado de piscicultura. Definindo-se, piscicultura é a criação de peixes e se enquadra como uma especialidade da aquicultura (RASGUIDO; ALBANEZ, 2000).

A piscicultura ornamental é a produção de peixes em cativeiro, desde a reprodução até a engorda, com a finalidade comercial, por exemplo, da ornamentação em aquários (RIBEIRO; LIMA; KOCHENBORGER, 2010). De acordo com (RIBEIRO, 2010), devido ao crescimento na prática do aquarismo, a piscicultura ornamental tornou-se uma área de grande desenvolvimento. Estima-se que o mercado de importação mundial de espécies aquáticas ornamentais movimente 300 milhões de dólares por ano.

A alta demanda por espécies ornamentais nos mercados dos maiores países consumidores obriga a importação de algumas espécies, pois sua produção não é autossuficiente. Cerca de 45% da demanda de mercado do Japão, por exemplo, é oriunda da importação de espécies aquáticas ornamentais. Países europeus, como Alemanha e Reino Unido, necessitam importar em torno de 20% de sua demanda de mercado (RIBEIRO, 2010).

Neste mercado, o Brasil encontra-se como exportador, principalmente devido aos peixes ornamentais coletados da Bacia Amazônica, nos estados do Amazonas e Pará. Somando-se a pesca de organismos aquáticos ornamentais destes dois estados, elas representam 88% do valor exportado anualmente pelos fornecedores brasileiros (RIBEIRO; LIMA; KOCHENBORGER, 2010).

Embora o mercado de peixes ornamentais coletados na natureza tenha importância econômica, ele pode apresentar impactos ambientais consideráveis, como a introdução de espécies exóticas em bacias fluviais devido à fuga das criações ornamentais causada pela pesca excessivas das espécies ornamentais (CASIMIRO et al., 2010) (DUGGAN; RIXON; MACISAAC, 2006) (MAGALHÃES, 2007) e o extrativismo, atividade de extração das espécies ornamentais diretamente de seu habitat (TLUSTY, 2004) (CHUQUIPIONDO, 2007).

2.2 Peixes Fluorescentes

Na tentativa de amenizar os problemas relatados na seção anterior, alternativas foram buscadas no desenvolvimento de novas tecnologias de modificação genética para melhorar as taxas de produção de espécies e introduzir novos atrativos ornamentais, como a emissão de fluorescência e a criação de espécies em diferentes cores. Devido ao aumento da oferta destas novas espécies de peixes ornamentais, pretende-se diminuir o extrativismo e a pesca excessiva de espécies já existentes, tanto em cativeiro quanto em seu habitat natural, gerando um equilíbrio neste mercado (GONG et al., 2003).

Sabe-se, que peixes fluorescentes obtidos através destas modificações genéticas são produzidos e comercializados nos EUA. Grupos de pesquisa de Singapura e Taiwan também produziram espécies de peixes ornamentais geneticamente modificados que apresentam fluorescência nas cores amarela, vermelho, azul, verde e púrpura, por exemplo, quando iluminados com luz ultravioleta. Além disso, verificou-se que essa fluorescência oriunda das modificações genéticas incrementa entre 300% e 400% o valor de mercado das espécies, em comparação com as espécies selvagens dos mesmos peixes, sem a fluorescência (GONG et al., 2003).

No Brasil, segundo (FIGUEIREDO et al., 2007), em 2004 foi produzida a primeira linhagem de peixes transgênicos fluorescentes. Com a intenção de estudar os efeitos do excesso de hormônio de crescimento em uma espécie de peixes, criou-se uma linhagem transgênica, inserindo-se juntamente com as mutações referentes ao hormônio de crescimento, o gene da proteína verde fluorescente para que fosse possível diferenciar as espécies transgênicas das demais. Ainda de acordo com (FIGUEIREDO et al., 2012), em 2010, outro estudo foi realizado criando-se nova espécie transgênica, porém neste caso, foi inserido o gene da proteína vermelha fluorescente para a identificação da linhagem transgênica. Ambas as espécies são mostradas na Figura 1.

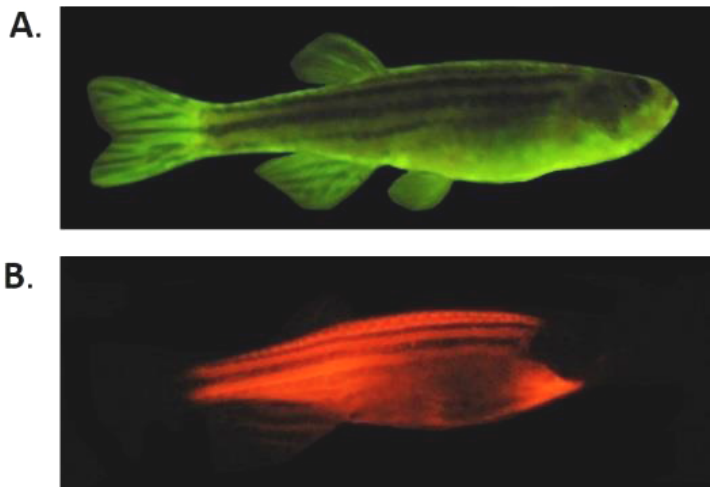


Figura 1: Peixes transgênicos produzidos no laboratório de Biologia Molecular da Universidade Federal do Rio Grande - FURG. A. Zebrafish transgênico para o gene do hormônio do crescimento (GH) e para o gene da proteína verde fluorescente (GFP). B. Zebrafish transgênico para o gene do receptor do hormônio do crescimento (GHR) e para a proteína vermelha fluorescente - figura reproduzida de (ALMEIDA, 2014)

É importante ressaltar que, apesar de ambas as linhagens apresentarem fluorescência verde ou vermelha, elas somente são visíveis quando as espécies de peixes são expostas à radiação ultravioleta. Outro ponto desfavorável foi a constatação da baixa estabilidade (diminuição da emissão de cor ao longo do tempo) da proteína fluorescente na espécie que contém a proteína vermelha. Assim, são destacadas limitações na produção transgênica de espécies ornamentais, evidenciando a necessidade de inovações e pesquisas contínuas nesta área (FIGUEIREDO et al., 2007) (FIGUEIREDO et al., 2012).

Dessa forma, o projeto "Peixes Transgênicos Fluorescentes: um novo campo para a piscicultura ornamental no Brasil" surge com o objetivo de desenvolver linhagens de peixes *zebrafish* (*Danio rerio*) ornamentais fluorescentes, a partir de novas variantes de proteínas fluorescentes que apresentem melhores características à piscicultura ornamental, como maior estabilidade, espectro de emissão de luz mais próximo ao visível e diferentes cores (ALMEIDA, 2014).

O desenvolvimento de novas variantes de proteínas fluorescentes, a partir da mutagênese, e a aquisição de conhecimento sobre as estruturas moleculares relacionadas a novas características impostas pelas mutações, através de análises e processos computacionais, consistem em importantes etapas da produção de peixes transgênicos ornamentais fluorescentes (ALMEIDA, 2014).

Dentre as etapas de análises e processos computacionais referentes ao projeto de pesquisa estão as relacionadas ao presente trabalho, como o desenvolvimento de um banco de dados de proteínas fluorescentes e a aplicação de técnicas de mineração de dados e análise de modelos preditivos (ALMEIDA, 2014).

2.3 Proteínas Fluorescentes

As proteínas fluorescentes coloridas representam um importante grupo de proteínas capazes de emitir fluorescência quando excitadas pelo correto comprimento de onda. Algumas características destas macromoléculas, suas principais aplicações e considerações sobre sua estrutura são abordadas nesta seção.

2.3.1 Características e Aplicações

A partir de estudos sobre a bioluminescência das águas-vivas *Aequorea Victoria* na década de 60, pesquisadores descobriram e isolaram uma importante proteína fluorescente, denominada *Green Fluorescent Protein* (GFP), ou Proteína Fluorescente Verde. Apesar de ela não emitir luminescência por si só, quando irradiada com luz visível na faixa do azul ela emite fluorescência verde intensa, daí a definição de seu nome (FARIAS, 2009).

A importância da GFP está relacionada ao fato do seu cromóforo fazer parte diretamente da cadeia peptídica da proteína, o que dispensa a necessidade de qualquer aditivo para ele fluorescer (YANG; MOSS; PHILLIPS, 1996). Essa característica, dentre outras, permitiu uma larga aplicação da GFP em pesquisas biológicas, sendo utilizada como marcador biológico (CHUDAKOV et al., 2010).

Além disso, pesquisas e experimentos foram realizados com a própria GFP e, através de mutações geradas no gene da proteína, produziu-se variantes melhoradas em estabilidade e brilho e comprimento de onda emitidos (FARIAS, 2009). Tais mutações geraram, inclusive, novas variantes de cor, criando proteínas fluorescentes da classe de cor azul (*Blue Fluorescent Proteins*), da classe de cor ciano (*Cyan Fluorescent Proteins*) e da classe de cor amarela (*Yellow Fluorescent Proteins*) (CHUDAKOV et al., 2010).

Outra proteína que merece destaque, segundo (GROSS et al., 2000), é a *Discosoma Red fluorescent protein* (DsRed), obtida do coral *Discosoma* e que emite fluorescência de cor vermelha. Da mesma forma como ocorreu com a GFP, variantes melhoradas da DsRed também estão sendo desenvolvidas, permitindo a criação de proteínas com diferentes tonalidades de vermelho, emissão de brilho e estabilidade (FARIAS, 2009). Mutações no gene desta proteína também geraram variantes de uma nova classe de cor, as proteínas fluorescentes laranjas (*Orange Fluorescent Proteins*) (CHUDAKOV et al., 2010).

Resumindo sua importância, devido às características já citadas e às suas propriedades fluorescentes, as proteínas fluorescentes se tornaram importantes marcadores de expressão gênica em células e tecidos, possibilitando que seja possível investigar o comportamento de células e também marcar tecidos, embriões e células-tronco, por exemplo. Isso tornou a GFP e suas variantes o principal gene repórter utilizado em pesquisas biológicas, biomédicas e biotecnológicas (FARIAS, 2009).

Além disso, o uso de proteínas fluorescentes no desenvolvimento de peixes

transgênicos fluorescentes, aumentando a diversidade de cores e adicionando fluorescência em espécies até então sem estas propriedades, revela também uma importância comercial desta proteína frente ao mercado da piscicultura ornamental (DUGGAN; RIXON; MACISAAC, 2006).

2.3.2 Estrutura da Proteína e o Cromóforo

As biomacromoléculas, como as proteínas, são classificadas estruturalmente em quatro diferentes níveis de complexidade, sendo que no presente trabalho somente os três primeiros níveis são abordados. A classificação em níveis é importante pois explica o motivo destas macromoléculas adotarem determinadas formas em meio biológico e, assim, desempenharem funções específicas (VERLI, 2014).

O primeiro nível, chamado de estrutura primária, compreende uma sequência de letras que representa a composição de um biopolímero. A representação desta informação é unidimensional, uma vez que esta sequência de letras descreve somente a ordem na qual os monômeros aparecem. Já a estrutura secundária, segundo nível de classificação, compreende as interações entre monômeros vizinhos e com as moléculas de solventes que os circundam. Estas interações, então, originam padrões repetitivos na organização espacial formando alguns tipos de elementos específicos, como por exemplo, no caso das proteínas, as hélices, alças e folhas-beta (VERLI, 2014).

Quando combinados, os elementos da estrutura secundária formam a estrutura tridimensional das biomoléculas, ou seja, o terceiro nível de classificação, chamado de estrutura terciária, que em outras palavras representa a montagem destes elementos da estrutura secundária no espaço tridimensional. A estrutura terciária é a que exerce a função biológica destas biomoléculas (VERLI, 2014).

Assim, segundo (CHUDAKOV et al., 2010), as estruturas terciárias das proteínas fluorescentes da família GFP são constituídas por cerca de 220-240 aminoácidos que assumem uma conformação no formato de um barril constituído por 11 folhas-beta que acomodam uma hélice distorcida em seu interior, como mostrado na Figura 2.

O grupo cromóforo - grupo de átomos responsável pela cor de um composto - é formado por uma única modificação pós-translacional de três aminoácidos da hélice nas posições 65, 66 e 67 (numeradas de acordo com a sequência de aminoácidos do gene selvagem da GFP, que possui o código de acesso 1GFL no *Protein Data Bank*). Do ponto de vista espacial, o cromóforo resultante está localizado no centro do barril e, por isso, está bem protegido do contato com o solvente que rodeia a proteína (CHUDAKOV et al., 2010).

A cadeia lateral dos aminoácidos localizados internamente no barril da proteína fluorescente exerce papel essencial na formação do cromóforo e no ajuste fino das propriedades de emissão de cor. Os aminoácidos mais importantes estão localizados no centro das folhas-beta, próximo do grupo cromóforo. Portanto, cada fita exerce uma espécie de

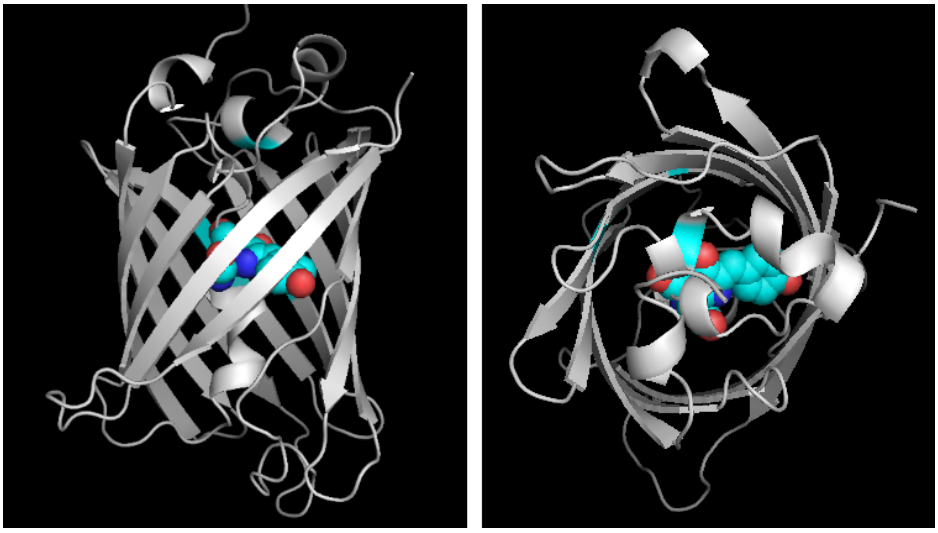


Figura 2: Representação da estrutura terciária da *Green Fluorescent Protein* - GFP (código de acesso 1GFL no *Protein Data Bank*) no formato de um barril composto por 11 folhas-beta e contendo uma hélice distorcida em seu interior (Vista frontal à esquerda e vista superior à direta). Em destaque, no interior, os átomos dos 3 aminoácidos formadores do cromóforo da proteína - do autor

controle sobre o cromóforo a partir de uma determinada direção (CHUDAKOV et al., 2010).

Como destaque, (CHUDAKOV et al., 2010) ressalta que experimentos mostram que mutações nestes aminoácidos mais próximos, causando por consequência mudanças nas cadeias laterais, podem alterar a faixa de cor e a emissão de fluorescência nas proteínas fluorescentes, inclusive tornando a proteína altamente fluorescente ou quase completamente não fluorescente.

2.4 Modelagem da Estrutura Tridimensional

Em muitos casos, necessita-se conhecer a estrutura tridimensional de uma proteína, mesmo que esta estrutura ainda não tenha sido experimentalmente resolvida. Neste contexto, surge a técnica de modelagem das estruturas tridimensionais de moléculas, que tem por objetivo modelar estruturas de proteínas a partir da sequência de aminoácidos, utilizando-se de métodos computacionais (VERLI, 2014).

Existem três classes de métodos para a predição da estrutura tridimensional de proteínas: modelagem por homologia (ou comparativa), *threading* (predição de enovelamento de proteínas) e predição *ab-initio* (VERLI, 2014) (HINCHLIFFE, 2008).

Na modelagem por homologia, a estrutura da sequência-alvo é modelada a partir de estruturas moldes, ou seja, estruturas resolvidas experimentalmente e que tenham identidade maior do que 25% (homólogas) com a sequência de aminoácidos da proteína-alvo. A estrutura-molde que apresentar a maior similaridade é a escolhida como molde para a

modelagem da proteína-alvo (ZAKI; BYSTROFF, 2008).

No método de *threading*, tenta-se ajustar a estrutura da proteína de interesse aos tipos de enovelamentos de proteínas conhecidas e depositados em bibliotecas de enovelamentos. É considerada uma forma intermediária entre o método de homologia e a predição *ab-initio*. Geralmente, é utilizado para sequências de proteínas que não possuem uma proteína homóloga no banco de dados de estruturas (HINCHLIFFE, 2008).

A predição *ab-initio* baseia-se nas propriedades físico-químicas conhecidas de cada aminoácido para a construção de funções de energia. Estas funções são minimizadas por algoritmos que realizam buscas no espaço de conformações que a proteína-alvo possa assumir. É um método ainda em desenvolvimento, normalmente utilizado para modelar pequenos trechos de sequências, devido ao alto custo computacional (HINCHLIFFE, 2008).

Dentre as três citadas, a modelagem por homologia é a metodologia que apresenta os melhores resultados, pois baseia-se em padrões gerais observados no processo de evolução biológica, demonstrando especialmente que a homologia entre sequências de aminoácidos implica em semelhança estrutural e funcional e que proteínas homólogas apresentam regiões internas conservadas (ZAKI; BYSTROFF, 2008) (VERLI, 2014).

Concluído o referencial teórico relacionado aos aspectos biológicos do presente trabalho, nos próximos capítulos é abordada a fundamentação teórica das análises e processos computacionais relacionados ao problema da predição da classe de cor de proteínas fluorescentes a partir de seus dados estruturais.

3 DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

Diante de um cenário de constante crescimento do volume de dados disponíveis em ambientes computacionais, surge nas organizações, seja de natureza empresarial ou acadêmica, a necessidade de analisar e utilizar do melhor modo possível o volume de dados disponíveis obtendo informações até então implícitas nestes dados. Como os volumes de dados crescem rapidamente, assim como sua complexidade, qualquer tipo de análise manual torna-se impraticável em vários domínios do conhecimento (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) (TAN; STEINBACH; KUMAR, 2006).

Segundo (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996), a Descoberta de Conhecimento em Banco de Dados (em inglês *Knowledge Discovery in Databases*, origem da sigla KDD) é um processo que surgiu junto com este contexto. O termo *Knowledge Discovery in Databases* foi formalizado em 1989, no primeiro *workshop* sobre o assunto.

Conforme a definição clássica, KDD é um processo não trivial, interativo e iterativo, para identificar padrões válidos, novos, potencialmente úteis e compreensíveis nos dados. E, por ser um processo, é composto por várias etapas (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Em outras palavras, é um processo cujo principal objetivo é extrair conhecimento a partir de bases de dados (TAN; STEINBACH; KUMAR, 2006).

Os termos da definição recém apresentada podem ser entendidos da seguinte forma: o termo interativo indica a atuação do homem no processo; o termo iterativo faz referência à possibilidade de repetições, seja de todo o processo ou apenas de algumas etapas, em busca de resultados satisfatórios; um padrão compreensível refere-se a uma representação do conhecimento que possa ser interpretada pelo homem; já um padrão válido indica que o conhecimento deve ser verdadeiro e estar contextualizado com a área da aplicação; finalmente um padrão, ou conhecimento útil, é aquele que pode ser aplicado e trazer benefícios à aplicação (HAN; KAMBER; PEI, 2011) (GOLDSCHMIDT; PASSOS, 2005).

Para iniciar um processo de KDD é fundamental que os envolvidos tenham entendimento do domínio da aplicação e objetivos bem definidos. Para tal, 3 categorias de pessoas estão geralmente envolvidas, são elas: o analista de dados, conhecedor das técnicas, algoritmos e ferramentas computacionais do processo de KDD; o especialista no domínio,

conhecedor do domínio ao qual os dados pertencem e sua aplicação; e, por último, os usuários, que utilizam os resultados do processo, ou seja, o conhecimento útil extraído (TAN; STEINBACH; KUMAR, 2006) (HAN; KAMBER; PEI, 2011).

A interdisciplinaridade do processo de KDD, já citada, é resultado da combinação de áreas do conhecimento como matemática, estatística, bancos de dados, inteligência artificial, visualização de dados, reconhecimento de padrões, dentre outras, que juntas fornecem teorias, técnicas e algoritmos que possibilitam a realização de todo o processo (HAN; KAMBER; PEI, 2011) (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Entre as áreas de aplicação onde mais cresce a utilização do processo de KDD estão marketing, gestão de qualidade, medicina, biologia, mercado financeiro, transporte e logística (ELMASRI; NAVATHE, 2003).

3.1 Tarefas de KDD

De acordo com os objetivos pretendidos com o processo de KDD, existem diferentes tarefas que podem ser executadas. Essas tarefas podem extrair diferentes tipos de conhecimento, o que torna necessário decidir no início do processo que tipo de conhecimento se deseja adquirir ao final do processo (HAN; KAMBER; PEI, 2011) (TAN; STEINBACH; KUMAR, 2006).

Evitando invadir demais a temática do Capítulo 4, uma vez que as técnicas de mineração estão diretamente ligadas à etapa de mineração de dados, se faz necessário abordar alguns conceitos para o correto entendimento deste tópico (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

É importante diferenciar o que é uma tarefa de KDD e o que é uma técnica de mineração de dados. A tarefa de KDD consiste na especificação do que se busca nos dados, ou seja, que tipo de padrão ou informação deseja-se obter. Já a técnica de mineração consiste na especificação de como conseguir os resultados pretendidos, ou seja, que tipos de métodos devem ser utilizados (neste escopo, técnicas e métodos são considerados sinônimos) (HAN; KAMBER; PEI, 2011) (TAN; STEINBACH; KUMAR, 2006) (PRATI, 2006).

Dessa forma, pode-se classificar as tarefas de KDD segundo a análise que se pretende realizar sobre um conjunto de dados: análise de amostragem, análise descritiva e análise de prognóstico (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) (GOLDSCHMIDT; PASSOS, 2005). A seguir, cada uma delas é explicada.

3.1.1 Tarefas de Amostragem

O objetivo deste tipo de análise é encontrar comportamentos que fogem muito à situação em geral, aumentando a confiança de determinada amostragem e, potencialmente, dos resultados (HAN; KAMBER; PEI, 2011). São tarefas de amostragem:

- *Detecção de desvios*

  Tarefa que tem por objetivo encontrar dados que não obedeçam ao comportamento geral do modelo de dados. Quando encontrados, podem ser tratados ou simplesmente descartados (HAN; KAMBER; PEI, 2011);

- *Análise de desvios*

  Similar à tarefa de detecção, porém, aqui uma medida de comparação é que define se um dado não obedece ao comportamento do modelo de dados, ou se já é um padrão estabelecido (HAN; KAMBER; PEI, 2011).

3.1.2 Tarefas Descritivas

As tarefas deste tipo de análise têm por objetivo estabelecer relações e associações entre os dados, capazes de descrever e caracterizar modelos de dados, possibilitando encontrar informações relevantes que seriam de difícil visualização no conjunto de dados (TAN; STEINBACH; KUMAR, 2006) (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). São caracterizadas por permitirem que se inicie o processo de KDD sem uma ideia ou hipótese previamente estabelecida (HAN; KAMBER; PEI, 2011). São tarefas descritivas:

- *Associações*

  Estas tarefas visam identificar grupo de fatos que ocorrem em conjunto ou de forma condicionada, encontrando associações e relacionamentos entre itens (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Geralmente, expressa-se os resultados através de regras de associação (TAN; STEINBACH; KUMAR, 2006);

- *Agrupamento*

  As tarefas de agrupamento caracterizam-se por separar o conjunto de dados em grupos de acordo com similaridade de certos atributos que direcionam esta ação. De certa foram, assemelha-se às tarefas de classificação, devido a formação de grupos (similar às classes), porém tais grupos são definidos durante a execução da tarefa, de acordo com o estabelecimento dos atributos que direcionam as semelhanças, e não previamente, como as classes das tarefas de classificação (GOLDSCHMIDT; PASSOS, 2005) (HAN; KAMBER; PEI, 2011);

- *Detecção de Sequências*

  Estas tarefas têm como objetivo estabelecer relacionamentos temporais (em sequência) entre fatos (BRAGA, 2005) (GOLDSCHMIDT; PASSOS, 2005);

- *Segmentação*

Tipo de tarefa que realiza a subdivisão do conjunto de dados em conjuntos menores de acordo com alguma distinção. Diferencia-se da tarefa de agrupamento por ser um passo intermediário, ou seja, utiliza-se a segmentação para depois aplicar uma nova tarefa sobre os dados segmentados (ZHANG; ZHANG; YANG, 2003) (PRATI, 2006)

3.1.3 Tarefas de Prognóstico

Categoria de tarefas que busca inferir um valor ou comportamento futuro ou estimar classes e valores desconhecidos, tendo como base as informações adquiridas em análises descritivas (HAN; KAMBER; PEI, 2011). Normalmente, inicia-se o processo de KDD já com uma ideia ou hipótese previamente estabelecida. São tarefas de prognóstico:

- *Classificação*

  Tarefa que consiste em categorizar os dados em classes previamente definidas de acordo com a similaridade de características nos dados. O modelo construído, então, é aplicado a dados não classificados a fim de categorizá-los posteriormente (HAN; KAMBER; PEI, 2011);

- *Estimação ou Regressão*

  Tarefa capaz de estimar determinado valor numérico, baseado nos demais atributos do conjunto de dados. Similar à classificação, porém só é utilizada para valores numéricos, e não classes (TAN; STEINBACH; KUMAR, 2006).

- *Predição*

  Nesta tarefa, similar às tarefas de classificação e regressão, o objetivo também é inferir o valor desconhecido de algo, só que neste caso, ela visa descobrir o valor futuro de determinado atributo do conjunto de dados (GOLDSCHMIDT; PASSOS, 2005).

3.2 Etapas de KDD

O processo de KDD é composto por uma série de etapas que compreendem inteiramente o fluxo percorrido pelos dados em análise, desde a seleção do conjunto de dados até a interpretação dos resultados e dos padrões gerados pelo processo (GOLDSCHMIDT; PASSOS, 2005).

A Figura 3, adaptada de (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996), fornece uma visão geral do processo de KDD, mostrando cada uma das cinco etapas do fluxo: seleção de dados, pré-processamento dos dados, transformação dos dados, mineração de

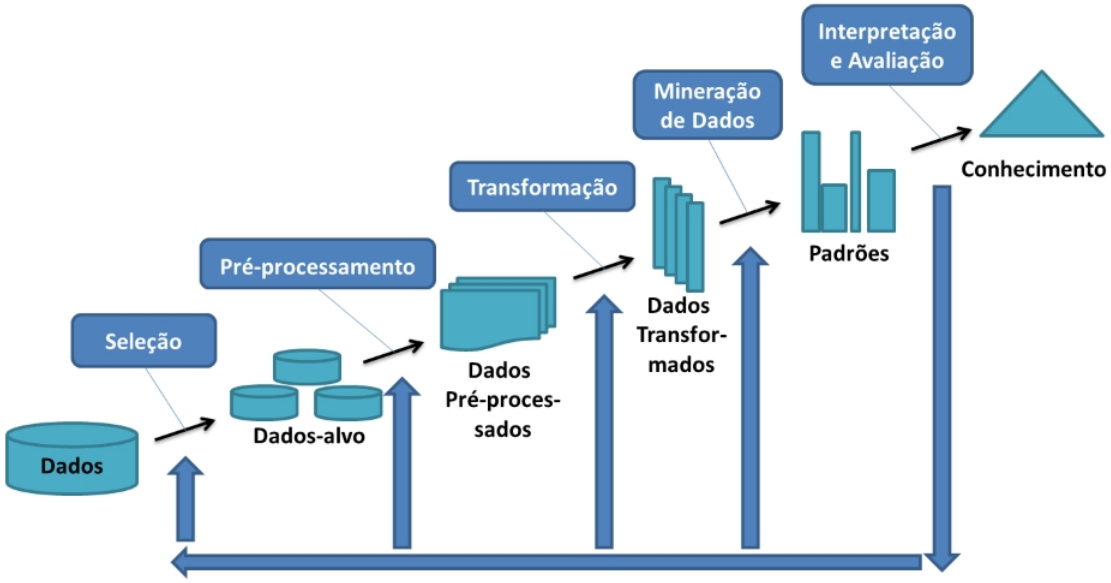


Figura 3: Visão geral das cinco etapas do processo de Descoberta de Conhecimento em Base de Dados - adaptada de (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996)

dados e a interpretação e avaliação dos resultados. Nota-se que é possível repetir o processo desde o início, ou somente a partir de alguma etapa anterior, a fim de se ajustar parâmetros ou revisar escolhas, com o objetivo de obter melhores resultados (HAN; KAMBER; PEI, 2011).

A seguir, as cinco etapas do processo de KDD são tratadas e explicadas em mais detalhes, com exceção da etapa de mineração de dados, que é abordada de maneira introdutória, já que, devido a sua importância (TAN; STEINBACH; KUMAR, 2006), é dedicado à ela todo um capítulo deste trabalho.

3.2.1 Seleção de Dados

Esta etapa, de acordo com o contexto da aplicação e objetivos traçados, busca o conjunto de dados apropriado nas bases existentes, ou seja, quais informações devem ser consideradas durante o processo, sejam estas bases internas ou externas (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). As bases internas são repositórios que já estão incorporados ao sistema da aplicação, por exemplo um *Data Warehouse* ou bases de dados operacionais. Já bases externas são oriundas de repositórios de outras localidades que não estão no sistema da aplicação, por exemplo, bases de dados públicas e documentos do especialista do domínio (JUBRAN et al., ????).

A maioria dos métodos da etapa de mineração de dados parte do princípio que os dados estão organizados em uma única, e possivelmente grande, estrutura. Há, basicamente, duas maneiras de agrupar as informações: na primeira, chamada junção direta, não há análise crítica quanto à contribuição dos atributos ou variáveis para o processo de

KDD, pois todos os atributos e registros da base de dados são incluídos na nova estrutura; na outra forma, chamada junção orientada, são escolhidos os atributos e registros que têm maior potencial para colaborar com o processo através de análise feita previamente pelo especialista do domínio da aplicação (ZHANG; ZHANG; YANG, 2003) (GOLDSCHMIDT; PASSOS, 2005).

Didaticamente, a seleção de dados, tal como descrita até o momento, é considerada a primeira etapa do fluxo de processo de KDD, porém é importante ressalvar que, na prática, a seleção de dados pode ocorrer em dois momentos (SOARES, 2007) (GOLDSCHMIDT; PASSOS, 2005). No primeiro momento ocorre a extração dos dados de diversas fontes para formar o conjunto de dados a ser analisado, como descrito anteriormente. Esta atividade também é denominada Coleta de Dados (ZHANG; ZHANG; YANG, 2003).

O importante nesta ressalva é o segundo momento no qual pode ocorrer uma seleção de dados, que é o de escolha dos atributos que serão efetivamente considerados na análise e que ocorre imediatamente antes da etapa de mineração de dados. Este segundo momento é chamado de Redução de Dados e é importante porque alguns atributos podem contribuir muito pouco ou nada com o processo, por não estarem de acordo com a semântica da aplicação ou possuírem forte restrição de unicidade, por exemplo (SOARES, 2007) (ZHANG; ZHANG; YANG, 2003) (BATISTA, 2003).

Considerando que os dados estejam em uma estrutura única, a Redução de Dados pode ser obtida sob dois enfoques distintos chamados de Redução Horizontal de Dados e Redução Vertical de Dados (BATISTA, 2003).

3.2.1.1 Redução Horizontal de Dados

Na Redução Horizontal ocorre a escolha de casos, ou seja, de registros. Esta operação pode ser feita escolhendo-se um ou mais atributos para guiar sua execução, sendo que, de acordo com estes atributos, é possível escolher casos para fazer parte ou não do conjunto final selecionado. Tais operações podem ser feitas, por exemplo, por instruções em linguagem de consulta estrutura (*Structured Query Language* - SQL), uma linguagem de pesquisa declarativa padrão para banco de dados relacionais (GOLDSCHMIDT; PASSOS, 2005) (BATISTA, 2003).

Outra maneira de fazer este tipo de seleção é por amostragem, na qual se seleciona registros de forma randômica, para formar um conjunto menor que o original. Os tipos de amostragem são classificados em Amostragem Aleatória Simples Com ou Sem Repetição, Amostragem em *Clusters* e Amostragem Estratificada (HAN; KAMBER; PEI, 2011).

Na Amostragem Aleatória Simples todos os registros possuem a mesma probabilidade de seleção, podendo ser permitida a possibilidade de repetição do registro selecionado ou não. Na Amostragem em *Clusters*, as tuplas são agrupadas em diferentes *clusters* e, em seguida, é realizada uma amostragem aleatória dentre os *clusters*. Já na Amostragem Estratificada, um conjunto de dados é segmentado em grupos disjuntos a partir de algum

atributo ou conjunto de atributos e, a partir daí, são selecionadas amostras aleatórias de cada grupo (HAN; KAMBER; PEI, 2011).

Uma última alternativa para se reduzir horizontalmente os dados é uma simples agregação de registros, quando isso for possível, porém sob pena de perda de detalhes (GOLDSCHMIDT; PASSOS, 2005).

3.2.1.2 Redução Vertical de Dados ou Seleção de Atributos

Esta técnica de seleção de dados trata da seleção de atributos. A Redução Vertical procura obter a combinação, com o mínimo de atributos, que deve ser considerada no processo de KDD. A seleção de atributos visa identificar e excluir o máximo de informações irrelevantes ou redundantes do conjunto de dados (MERSCHMANN, 2007).

Uma boa seleção de atributos pode levar, através de um conjunto bem selecionado, a modelos de conhecimento mais precisos e confiáveis. A eliminação de um atributo é muito mais significativa do que a eliminação de um registro no conjunto de dados, e por isso, é considerada mais importante (SOARES, 2007) (MERSCHMANN, 2007).

De acordo com a abordagem usada para avaliação de atributos, as técnicas de seleção de atributos podem ser classificadas como *filter* ou *wrapper*. No caso dos *wrappers*, a avaliação dos atributos é feita por um algoritmo de mineração de dados específico e o desempenho do subconjunto de atributos é medido de acordo com o desempenho do algoritmo (MERSCHMANN, 2007).

Por outro lado, *filters* trabalham de forma que os atributos são selecionados levando em conta apenas as características gerais dos dados, tal como a capacidade de discriminação dos atributos com relação à classe, pois operam independentemente de qualquer algoritmo de mineração de dados (MERSCHMANN, 2007). Assim, o método *wrapper* tem a vantagem de gerar um subconjunto de atributos que pode aumentar o desempenho e precisão do algoritmo de mineração de dados, porém é mais lento que o método *filter* e, ainda, o melhor subconjunto para um algoritmo de mineração pode não ser tão bom para outro (SOARES, 2007).

Existem também técnicas de seleção de atributos que consideram cada atributo individualmente, selecionando os melhores atributos para fazer parte do subconjunto no qual será aplicado o algoritmo de mineração de dados, através da sua capacidade preditiva, por exemplo (MERSCHMANN, 2007).

Outra maneira de se realizar a redução vertical de atributos é através da Eliminação Direta, uma técnica manual onde deve-se ter conhecimento prévio e especializado do problema para saber quais atributos realmente não são relevantes para o processo de KDD (necessária participação do especialista no domínio). Por exemplo, podem-se eliminar atributos com valores constantes ou que sejam somente elementos identificadores do registro (HAN; KAMBER; PEI, 2011).

Há ainda uma alternativa à redução de dados vertical, conhecida como redução de

valores. Este tipo de operação consiste em reduzir o número de valores distintos de cada atributo, não excluindo o atributo em si. Tal manipulação pode melhorar o desempenho do algoritmo de mineração de dados e diminuir o tempo de processamento, pois com menos valores distintos, menos comparações são feitas. Pode ser utilizado em atributos que apresentem valores discretos ou categóricos (GOLDSCHMIDT; PASSOS, 2005).

3.2.2 Pré-Processamento

A etapa de pré-processamento é formada por atividades que tratam da organização, tratamento e limpeza dos dados para a próxima etapa. Nesta etapa, existem atividades muito dependentes do domínio da aplicação, por isso é importante a participação do especialista do domínio (ZHANG; ZHANG; YANG, 2003) (TAN; STEINBACH; KUMAR, 2006).

De forma geral, o objetivo desta fase é corrigir a base de dados. Esta etapa envolve uma avaliação da consistência das informações, correção de possíveis erros, tratamento de valores ausentes ou redundantes, e ainda a eliminação de valores não pertencentes ao domínio (HAN; KAMBER; PEI, 2011).

Valores inconsistentes são aqueles que contêm alguma discrepância semântica entre si, já valores não pertencentes ao domínio são os valores que não estão entre os valores possíveis de um atributo. A limpeza de dados com estas características pode ocorrer com a exclusão dos atributos ou pela correção dos erros (HAN; KAMBER; PEI, 2011) (GOLDSCHMIDT; PASSOS, 2005).

Não existe consenso sobre a utilização de técnicas para preenchimento de valores ausentes, pois depende de caso a caso e da natureza do problema. Além da exclusão de atributos para eliminar valores ausentes, existem opções para o preenchimento destes valores. Tais valores podem ser preenchidos de forma manual, com valores globais constantes, ou com medidas estatísticas como a média ou a moda dos valores do mesmo atributo nos demais registros (TAN; STEINBACH; KUMAR, 2006) (BATISTA, 2003).

Também é possível o preenchimento através de métodos de mineração de dados, como a aplicação de técnicas de agrupamento para auxiliar na descoberta dos melhores valores, ou até mesmo predição de valores, proporcionando que eventuais relacionamentos entre atributos possam ser mantidos, pois os próprios atributos são utilizados para prever valores de outro atributo. Este último método citado é muito usado na prática (HAN; KAMBER; PEI, 2011).

Ainda constitui esta etapa, embora seja realizada, tipicamente, após a etapa de transformação, a operação de balanceamento das classes do conjunto de dados em problemas de classificação. O balanceamento é indicado na ocorrência de registros em grande quantidade representando determinada classe e, por outro lado, poucos registros representativos de outra classe. A fim de evitar que o modelo de classificação gerado seja muito especializado na classe majoritária, devido ao desequilíbrio no número de instâncias,

executa-se operações com o intuito corrigir, ou pelo menos minimizar, esse desbalanceamento entre as classes (HAN; KAMBER; PEI, 2011).

3.2.3 Transformação

A transformação dos dados, também chamada de codificação por alguns autores, tem por objetivo transformar a natureza dos valores dos atributos. Esta etapa é responsável pela forma como os dados serão representados durante o processo de KDD. Os dados devem ser codificados de maneira a atender às necessidades específicas dos algoritmos de mineração de dados a serem utilizados, facilitando assim o seu uso por esses algoritmos (HAN; KAMBER; PEI, 2011) (SOARES, 2007).

Existe a codificação classificada como Numérico-Categórica, quando divide-se valores de atributos contínuos em valores categóricos ou intervalos codificados, e a classificada como Categórico-Numérica, a qual representa valores categóricos em códigos numéricos (BATISTA, 2003). A técnica de Discretização, que é a representação de valores por intervalos discretos, é um tipo de transformação Numérico-Categórica, porém cabe ressalvar que alguns autores consideram esta técnica como uma Redução de Valores (GOLDSCHMIDT; PASSOS, 2005).

Uma operação comumente utiliza nesta etapa é a criação de atributos, que consiste em gerar atributos derivados de outros já existentes no conjunto de dados. A importância dessa operação é justificada pois os novos atributos, além de expressarem relacionamentos conhecidos entre os atributos existentes, podem reduzir o conjunto de dados, simplificando o processamento dos algoritmos de mineração de dados (GOLDSCHMIDT; PASSOS, 2005).

Além destas, também é comum as seguintes transformações nos dados: a generalização, que converte valores muito específicos em valores mais genéricos; a padronização, que matematicamente dispõe os valores em uma mesma escala; e a normalização, onde os valores assumem uma distribuição normal, dentro de intervalo específico (mais comumente $[0, 1]$ ou $[-1, 1]$), sendo normalizados pela média e pelo desvio padrão de cada atributo (HAN; KAMBER; PEI, 2011) (SOARES, 2007).

3.2.4 Mineração de Dados

A mineração de dados (MD) é a etapa do processo de KDD que consiste em aplicar algoritmos específicos, de análise e descoberta, para identificação de padrões sobre dados. Envolve a aplicação repetida e iterativa de métodos destes algoritmos, seja para construção de modelos de conhecimento ou para determinação de padrões sobre os dados analisados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Esta é a principal etapa do processo de KDD, pois é onde ocorre a busca efetiva por conhecimentos novos e úteis, surgindo assim o uso dos termos mineração de dados e KDD como sinônimos por parte de alguns autores (TAN; STEINBACH; KUMAR, 2006). Dada

a importância desta etapa para todo o processo de KDD, o próximo capítulo é dedicado em toda a sua extensão à esta etapa e alguns de seus métodos.

3.2.5 Interpretação e Avaliação

Nesta última etapa do processo de descoberta de conhecimento, o objetivo é analisar e interpretar os padrões obtidos na etapa de mineração de dados, para identificar quais destes padrões pode ser considerado um novo conhecimento referente ao domínio da aplicação (GOLDSCHMIDT; PASSOS, 2005). Após, o próximo passo consiste em consolidar o conhecimento gerado incorporando-o dentro de outros sistemas, documentando-o ou utilizando-o no auxílio à tomada de decisão (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Também é na etapa de interpretação e avaliação que o analista de KDD avalia a necessidade de reiniciar ou não a execução de qualquer uma das etapas anteriores na tentativa de buscar melhores resultados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). No que se refere a avaliação em si dos resultados, mais detalhes são tratados em capítulo específico, onde aborda-se os métodos de avaliação e as métricas de desempenho.

Quanto a interpretação, considera-se as seguintes tarefas como pertencentes a esta etapa do processo (GOLDSCHMIDT; PASSOS, 2005) (TAN; STEINBACH; KUMAR, 2006):

- Simplificações do Modelo de Conhecimento: Nesta tarefa o objetivo é reduzir o volume de padrões gerados, a fim de facilitar o entendimento do modelo gerado.

- Transformações do Modelo de Conhecimento: Esta tarefa tem por objetivo converter um modelo de conhecimento em outro modelo existente, de forma a tornar mais fácil o seu entendimento.

- Organização e Apresentação dos Resultados: O objetivo desta tarefa é visualizar em duas ou três dimensões o modelo de conhecimento gerado, através do uso de tabelas, árvores, gráficos, planilhas, entre outros, visando a facilidade no entendimento do modelo.

4 MINERAÇÃO DE DADOS

A etapa de mineração de dados é a responsável por efetivamente explorar o conjunto de dados previamente preprocessado e codificado, com o objetivo de estabelecer descrições, relações, associações e padrões entre os dados, transformando dados brutos em informação útil e conhecimento (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

De acordo com a tarefa de KDD que se deseja executar, na etapa de mineração de dados deve ser escolhida a técnica de mineração adequada para tal tarefa. Uma técnica de mineração de dados consiste na especificação dos métodos de como descobrir as relações e padrões que interessam, ou seja, são as diferentes abordagens utilizadas pelas tarefas para alcançar seus objetivos. As técnicas de mineração, por sua vez, se utilizam de diferentes tipos de algoritmos para implementar o método necessário para realizar suas ações (HAN; KAMBER; PEI, 2011) (TAN; STEINBACH; KUMAR, 2006).

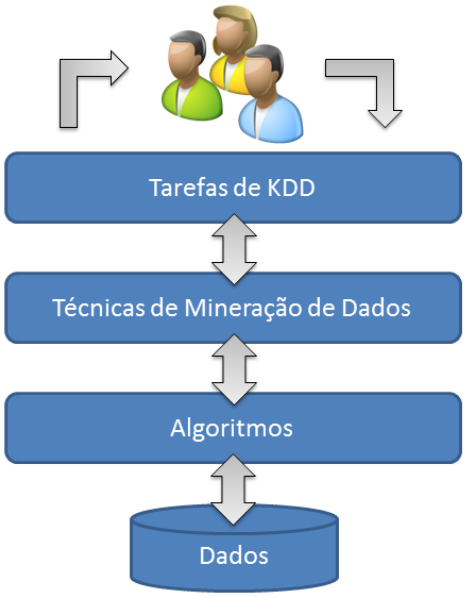


Figura 4: Representação da relação interativa e hierárquica entre as tarefas de KDD, as técnicas de mineração de dados e seus algoritmos - adaptada de (HAN; KAMBER; PEI, 2011) e (TAN; STEINBACH; KUMAR, 2006)

O esquema da relação interativa e hierárquica entre as tarefas de KDD, as técnicas de mineração de dados e seus algoritmos pode ser melhor visualizado na Figura 4. Nota-se que os analistas e especialistas definem a tarefa a ser realizada pelo processo de KDD, onde cada tarefa possui técnicas a disposição que, por sua vez, são implementadas por algoritmos específicos, os quais efetivamente acessam e utilizam o conjunto de dados. Após, de acordo com o objetivo inicial da tarefa de KDD, os resultados são visualizados e interpretados pelos analistas e especialistas do domínio, no topo do esquema representado (o esquema é somente uma representação das relações, não um fluxo de informações ou de dados) (HAN; KAMBER; PEI, 2011) (TAN; STEINBACH; KUMAR, 2006).

Dentre as principais técnicas de mineração de dados, exitem técnicas estatísticas, técnicas de aprendizado de máquina e técnicas baseadas em *crescimento-poda-validação*, dentre outras (HAN; KAMBER; PEI, 2011) (TAN; STEINBACH; KUMAR, 2006). Através da existência destas técnicas somadas à sua aplicação em conjuntos de dados evidencia-se a natureza interdisciplinar da mineração de dados, envolvendo áreas como Inteligência Artificial, Banco de Dados e Estatística, conforme ilustrado na Figura 5 (HAN; KAMBER; PEI, 2011).

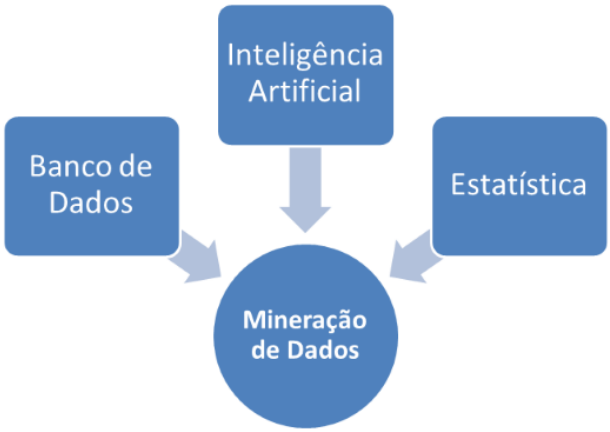


Figura 5: Representação da natureza interdisciplinar da mineração de dados, envolvendo as áreas de Inteligência Artificial, Banco de Dados e Estatística - adaptada de (HAN; KAMBER; PEI, 2011)

4.1 Aprendizado de Máquina

Dentro desse contexto, é importante destacar alguns conceitos relativos à Inteligência Artificial (IA), como o aprendizado de máquina, pois está intimamente ligado aos algoritmos de mineração de dados e sua capacidade de extração de padrões (HAN; KAMBER; PEI, 2011).

O termo aprendizado é classicamente definido como qualquer mudança em um sistema que melhore o seu desempenho na segunda vez que ele repetir a mesma tarefa (SI-

MON, 1983). Dessa forma, o aprendizado de máquina é uma parte da IA responsável pelo desenvolvimento de teorias computacionais com foco na criação de conhecimento artificial, ou ainda, é uma área da IA cujo objetivo é desenvolver técnicas computacionais sobre o processo de aprendizado (BISHOP, 2007) (RUSSELL; NORVIG, 2003).

Outro termo que é necessário ser definido é a indução, que é a inferência de conhecimento a partir dos dados. Assim, neste contexto, aprendizagem indutiva é o processo de construção de um modelo em que o ambiente (conjunto de dados) é analisado na busca por tendências e padrões. É importante ressaltar que o conjunto de dados pode ser dinâmico, logo o modelo deve ser adaptativo, isto é, deve ter a capacidade de aprender a partir do ambiente. A aprendizagem indutiva, então, pode ser realizada através de duas estratégias distintas: supervisionada e não supervisionada (BATISTA, 2003) (HAN; KAMBER; PEI, 2011).

- *Aprendizagem supervisionada*

  A aprendizagem supervisionada é feita a partir de exemplos, onde o analista contribui com o sistema na construção do modelo de dados, através da definição dos exemplos e a qual classe cada exemplo pertence. O sistema tem que determinar a descrição para cada classe, ou seja, o conjunto de propriedades comuns nos exemplos que lhe são fornecidos. Tendo por base uma tarefa de prognóstico, estando a descrição determinada, é possível formular regras de classificação que podem ser utilizadas para predizer a classe de um objeto que não tenha sido considerado nos exemplos da aprendizagem (HAN; KAMBER; PEI, 2011) (BATISTA, 2003) (RUSSELL; NORVIG, 2003);

- *Aprendizagem não supervisionada*

  Esta aprendizagem é realizada com base em observação e descoberta. Como não são definidas classes para os exemplos, o sistema necessita observar os exemplos e reconhecer os padrões por si próprio, resultando em um conjunto de grupos que apresentam padrões de similaridade. Geralmente, é necessário realizar uma análise posterior para definir o que cada agrupamento representa no contexto do conjunto de dados (HAN; KAMBER; PEI, 2011) (RUSSELL; NORVIG, 2003) (BATISTA, 2003).

4.2 Classificação

Devido aos objetivos e proposta do presente trabalho, fica evidente a opção de utilizar a tarefa de classificação para realizar as análises necessárias. Assim, nesta seção, aborda-se em mais detalhes esta tarefa e suas características.

A tarefa de classificação, em uma definição clássica, consiste em buscar uma função que mapeie os dados, ou seja, classifique corretamente registros de dados em uma dentre

diferentes classes de dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Em outras palavras, a tarefa de classificação busca padrões em registros de dados rotulando-os em um número finito de categorias previamente definidas, chamadas de classes, e cada registro pertence a uma classe de modo que o modelo gerado pela tarefa de classificação pode ser aplicado a registros ainda não classificados, com o objetivo de classificá-los segundo as classes existentes (HAN; KAMBER; PEI, 2011).

Sob o ponto de vista do aprendizado de máquina, a tarefa de classificação é o processo de aprendizagem de uma função que classifica um dado objeto de interesse em uma das possíveis classes (ELMASRI; NAVATHE, 2003). Devido à suas características de aprendizado supervisionado, o esquema de aprendizagem é apresentado como um conjunto de exemplos classificados, a partir do qual se espera aprender a classificar exemplos ainda não-vistos (RUSSELL; NORVIG, 2003).

A classificação, tomando-se uma descrição simplista, é a tarefa de aprender uma função alvo f que mapeie cada elemento do conjunto de atributos X para um dos rótulos do conjunto de classes Y pré-determinados. Essa função f é o modelo de classificação, conforme ilustra a Figura 6. Observa-se que x_i representa qualquer registro do conjunto de dados e y_i qualquer rótulo do conjunto de classes. Nos casos em que ocorrer a inferência de um registro do conjunto de dados em um rótulo do conjunto de classes, essa inferência é denominada classificação (TAN; STEINBACH; KUMAR, 2006) (GOLDSCHMIDT; PASSOS, 2005).

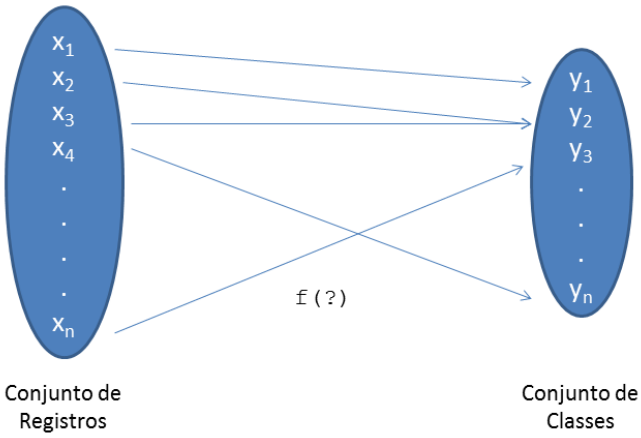


Figura 6: Representação das relações entre os registros de dados e suas respectivas classes - adaptada de (GOLDSCHMIDT; PASSOS, 2005)

Existe uma formalização para a tarefa de classificação, considerando um par $(x, f(x))$ como um exemplo, onde x é a entrada e $f(x)$ a função aplicada a x. A tarefa deve obter uma função h que se aproxime de f, dada uma coleção de exemplos de f. Segundo o autor, a função h obtida é uma hipótese, e toda a hipótese encontrada é chamada de classificador (TAN; STEINBACH; KUMAR, 2006).

A acurácia da função, ou hipótese h, retrata a quantidade de acertos da hipótese em classificar as entradas ainda não vistas, ou seja, aquelas que formam o conjunto de testes. O conjunto de entradas utilizadas para encontrar a melhor hipótese h é chamado de conjunto de treinamento. Depois de conhecido este conjunto, o classificador pode ser aplicado a novos registros para rotulá-los de acordo com as classes existentes (TAN; STEINBACH; KUMAR, 2006).

Neste contexto, é importante ressaltar que cada algoritmo de inferência possui um viés indutivo, que direciona o processo de construção dos classificadores. O viés indutivo de um algoritmo é o conjunto de fatores que, de forma coletiva, influenciam na seleção de hipóteses (UTGOFF, 1986).

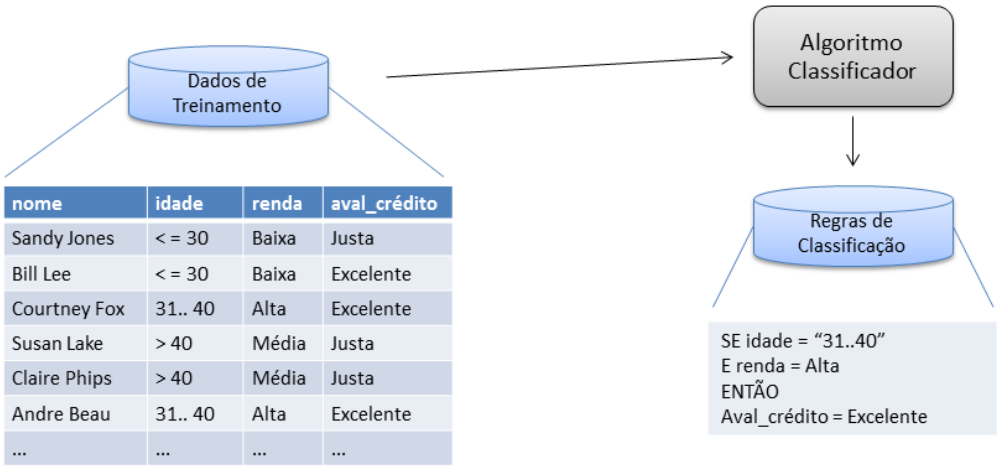
Na prática, o viés indutivo afeta o processo de inferência restringindo o espaço de hipóteses e impondo ordem de preferência sobre as hipóteses. Assim, conclui-se o que já está disposto no teorema NFL - *No Free Lunch* - (WOLPERT, 1996): não há um algoritmo de classificação que seja superior a todos os outros em qualquer problema de classificação (BENSUSAN, 1999). Tal afirmação é outro fator que justifica a proposta do presente trabalho, de comparar a performance de classificadores aplicados a um mesmo problema de classificação.

De maneira funcional, a maioria dos problemas de classificação tem duas fases: primeiramente uma etapa de treinamento ou aprendizado, executada a partir do conjunto de dados de treinamento; e posteriormente uma etapa de generalização, preferencialmente executada a partir de dados não contidos no conjunto de treinamento, chamados de dados de teste.

O primeiro passo da tarefa de classificação inicia na construção de um conjunto prédeterminado de classes e conceitos, através da análise dos elementos de tuplas (ou registros) de um banco de dados, descritos aqui como atributos. Assume-se que cada registro pertence a uma determinada classe, a qual é determinada por um dos atributos desse registro, denominado atributo de rótulo de classe. Neste cenário, as tuplas ou registros também podem ser chamados de amostras, exemplos ou objetos (HAN; KAMBER; PEI, 2011).

Estes registros selecionados a partir dos dados disponíveis, seguindo tarefas de préprocessamento e transformação, formam o conjunto de dados. Parte destes registros (ou todos, dependendo da metodologia utilizada) formam o conjunto de dados de treinamento e são utilizados pelo algoritmo de classificação para executar o aprendizado dos padrões, chamado de indução, possibilitando a extração de um conjunto de padrões (HAN; KAMBER; PEI, 2011). A Figura 7 ilustra o explanado neste primeiro passo, exemplificando a extração de padrões representados por regras de classificação.

No segundo passo da tarefa de classificação, chamada de generalização, o modelo é utilizado para a classificação do conjunto de dados de teste, a porção restante do conjunto de dados que não foi utilizada pelo treinamento. Dessa forma, é possível, através do teste pelos padrões extraídos, avaliar o modelo construído produzindo parâmetros de

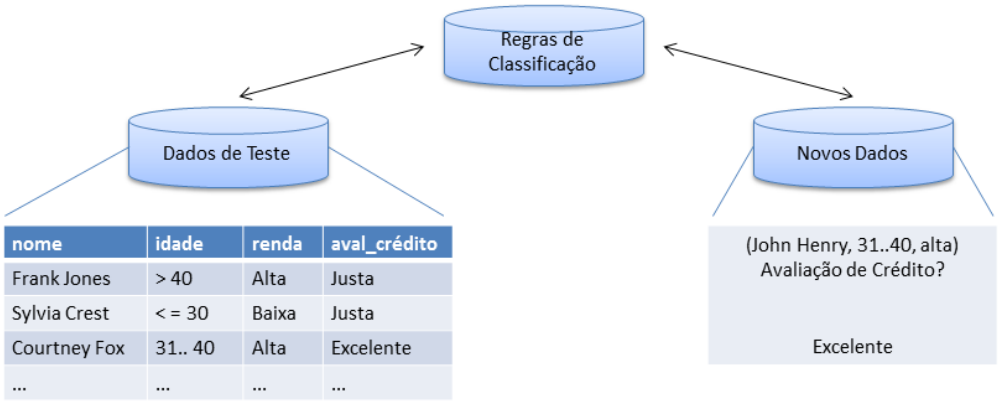


nome	idade	renda	aval_crédito
Sandy Jones	< = 30	Baixa	Justa
Bill Lee	< = 30	Baixa	Excelente
Courtney Fox	31.. 40	Alta	Excelente
Susan Lake	> 40	Média	Justa
Claire Phips	> 40	Média	Justa
Andre Beau	31.. 40	Alta	Excelente
...	...	...	...

Figura 7: Ilustração das atividades referentes à etapa de treinamento ou aprendizado (1ª etapa) em uma tarefa de classificação - adaptada de (HAN; KAMBER; PEI, 2011)

desempenho (HAN; KAMBER; PEI, 2011).

Caso as medidas de desempenho sejam consideradas aceitáveis, o modelo pode ser utilizado para classificar futuros exemplos dos quais se desconhece a classe. De acordo com os padrões contidos no modelo, o classificador é capaz de predizer a classe do exemplo submetido a ele (HAN; KAMBER; PEI, 2011). As ocorrências deste segundo passo estão ilustradas na Figura 8, novamente utilizando-se regras de classificação como o padrão extraído.



nome	idade	renda	aval_crédito
Frank Jones	> 40	Alta	Justa
Sylvia Crest	< = 30	Baixa	Justa
Courtney Fox	31.. 40	Alta	Excelente
...	...	...	...

Figura 8: Ilustração das atividades referentes à etapa de generalização (2ª etapa) em uma tarefa de classificação - adaptada de (HAN; KAMBER; PEI, 2011)

Mais detalhes sobre métodos de avaliação e métricas de desempenho em classificadores são abordados no Capítulo 5.

4.3 Técnicas e Algoritmos de Mineração de Dados

Como citado anteriormente, na etapa de Mineração de Dados, é necessário definir a técnica e os possíveis algoritmos que utilizam a técnica escolhida, em função da tarefa de KDD desejada. No caso deste trabalho, a tarefa de KDD executada é a Classificação, para a qual existem diversas técnicas e algoritmos que podem ser empregados, tais como os baseados em Árvores de Decisão, em Redes *Bayseanas*, em Redes Neurais Artificiais, em Máquinas de Vetores de Suporte, em Algoritmos Genéticos, dentre outros (HAN; KAMBER; PEI, 2011).

Segundo as justificativas apresentadas no Capítulo 1, descreve-se a seguir as três técnicas e algoritmos utilizados no presente trabalho: Árvores de Decisão (QUINLAN, 1986) (QUINLAN, 1993), Redes Neurais Artificiais (RUMELHART; HINTON; WILLIAMS, 1986) e Máquinas de Vetores de Suporte (VAPNIK, 1998).

4.3.1 Árvores de Decisão

A técnica de Árvore de Decisão (AD) é um modelo estatístico que utiliza aprendizado supervisionado para classificação e previsão de dados, através do qual a função aprendida é representada por uma árvore de decisão. Este modelo também pode ser representado como um conjunto de regras do tipo *se-então* para, assim, facilitar a compreensão e interpretação dos resultados. Devido a facilidade de entendimento do modelo, esta técnica é amplamente utilizada em problemas de classificação (QUINLAN, 1986) (TAN; STEINBACH; KUMAR, 2006).

Os algoritmos de indução de árvores de decisão caracterizam-se pelo uso de uma técnica chamada Divisão e Conquista em sua execução. Esta estratégia baseia-se na sucessiva divisão do problema em vários subproblemas de menores dimensões, até que uma solução para cada um dos subproblemas menores seja encontrada. Através dessa estratégia, os algoritmos de árvores de decisão buscam dividir sucessivamente o conjunto de dados em vários subconjuntos, até que cada subconjunto contemple apenas uma classe ou até que uma das classes demonstre uma clara minoria, injustificando novas divisões (QUINLAN, 1986).

Alguns conceitos sobre árvores de decisão precisam ser destacados, tendo como exemplo a árvore hipotética da Figura 9: o nó-raiz é o item localizado no topo da árvore, sua origem; os nós são todos os itens representados na árvore; os ramos são as ligações entre os nós; os nós-filhos são os nós logo abaixo do respectivo nó-pai; e, por último, os nós-folhas, são os nós que não possuem filhos, onde as ramificações da árvore se encerram (RUSSELL; NORVIG, 2003).

A associação entre tais conceitos e suas respectivas definições na utilização de árvores de decisão como um modelo de classificação se faz necessária a partir deste momento. Para realizar esta associação, antes define-se um conjunto de dados T contendo n regis-

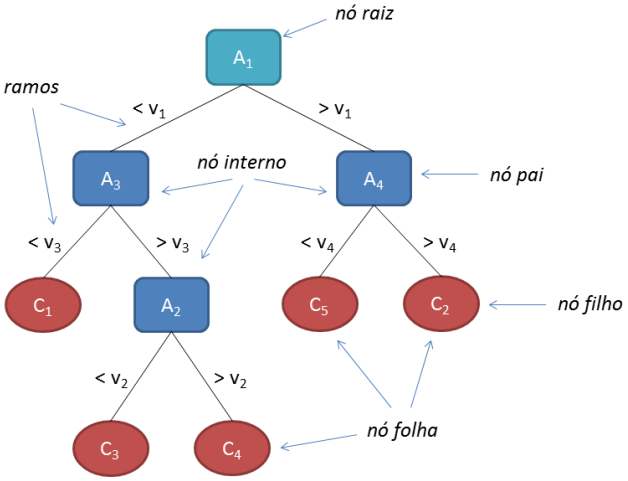


Figura 9: Árvore de Decisão hipotética que ilustra alguns conceitos em relação ao modelo - adaptada de (HAN; KAMBER; PEI, 2011)

tros, onde cada registro contém dois tipos de atributos: o atributo classe, que indica a classe a qual pertence o registro; e os atributos preditivos, os quais são analisados pelo algoritmo de aprendizado para descoberta de como se relacionam com o atributo classe (TAN; STEINBACH; KUMAR, 2006) (HAN; KAMBER; PEI, 2011).

A partir disso, em um modelo de classificação de árvore de decisão têm-se: cada nó (incluindo a raiz) representa um teste sobre um atributo preditivo; um ramo descendente partindo de um nó (incluindo a raiz) representa um possível resultado para o teste sobre o atributo que o nó em questão representa; um nó-folha representa um atributo rótulo de classe. A classificação de um novo registro ocorre percorrendo-se um caminho na árvore, ou seja, desde a raiz, testando-se os atributos dos nós, até atingir alguma folha, indicando a classe desse novo registro (QUINLAN, 1986).

A extração das regras *se-então* de um modelo de árvore de decisão é feito considerando um trajeto ou caminho do nó raiz até um nó-folha da árvore, conforme ilustra a área destacada na Figura 10. Os condicionais *se* de uma regra são formados pelos atributos preditivos que surgem ao longo do caminho percorrido, testando os valores que os definem, e as consequências *então* são formadas pelo atributo classe (TAN; STEINBACH; KUMAR, 2006).

Assim, a regra *se-então* extraída da árvore de decisão hipotética conforme destaca a Figura 10 pode ser assim interpretada: *se* $A_1 > v_1$ e $A_4 > v_4$, *então* C_2.

4.3.1.1 Indução de Árvores de Decisão

O algoritmo *Top-Down Induction of Decision Tree* (TDIDT) é bem conhecido e é utilizado como base para muitos algoritmos de indução de árvores de decisão, dentre eles os mais conhecidos ID3 (QUINLAN, 1986) e C4.5 (QUINLAN, 1993). O TDIDT

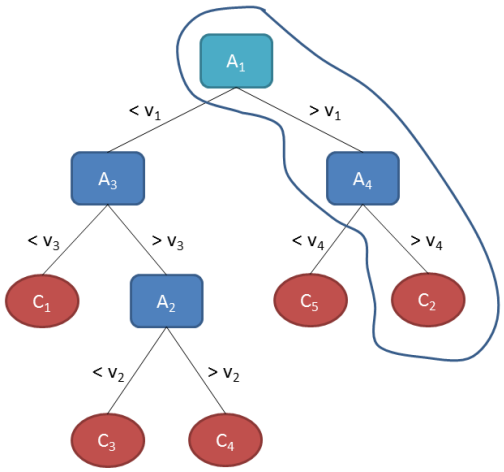


Figura 10: Representação da extração de uma regra *se-então* a partir de uma árvore de decisão hipotética - adaptada de (HAN; KAMBER; PEI, 2011)

produz regras de decisão de forma implícita numa árvore de decisão, a qual é construída por sucessivas divisões do conjunto de dados de acordo com os valores de seus atributos preditivos (QUINLAN, 1986).

De maneira formal, o algoritmo TDIDT baseia-se em três possibilidades, partindo de um conjunto de dados T contendo as classes $C_1, C_2, ..., C_k$ (QUINLAN, 1986) (HAN; KAMBER; PEI, 2011):

1. T contém um ou mais objetos, sendo todos da classe C_j. Assim, a árvore de decisão para T é um único nó-folha que identifica a classe C_j;

2. T não contém objetos. Dessa forma, a árvore para T também é um único nó-folha, mas a classe associada deve ser determinada por uma informação externa, através de conhecimento do domínio do problema;

3. T contém exemplos pertencentes a mais de uma classe. Neste caso, o objetivo é dividir T em subconjuntos que possuam, ou tendam a ter, registros pertencentes a uma única classe. Para isso, é escolhido um atributo preditivo A que possui um ou mais possíveis valores distintos $O_1, O_2, ..., O_n$. O conjunto T é particionado nos subconjuntos $T_1, T_2, ..., T_n$, onde T_i contém os exemplos de T que têm valor O_i para o atributo A. Assim, a árvore de decisão para T consiste de um nó de decisão identificando o teste sobre A e n ramos para cada um dos possíveis valores O_i.

Assim, os passos 1, 2 e 3 são aplicados recursivamente para cada subconjunto de dados T_i, para i de $1..n$, até construir toda a árvore de decisão (QUINLAN, 1986) (HAN; KAMBER; PEI, 2011).

De forma resumida, o algoritmo TDIDT é um algoritmo recursivo de busca gulosa que procura, sobre um conjunto de atributos, aqueles que melhor dividem o conjunto

de exemplos em subconjuntos menores. Inicialmente, todos os exemplos são colocados em um único nó, chamado de raiz. A seguir, um atributo preditivo é escolhido para representar o teste desse nó e, assim, dividir os exemplos em subconjuntos. Esse processo se repete recursivamente até que todos os exemplos já estejam classificados ou, então, até que todos os atributos preditivos já tenham sido utilizados (QUINLAN, 1986) (HAN; KAMBER; PEI, 2011).

A escolha do melhor atributo para as divisões do conjunto de exemplos segue critérios de seleção, que são definidos em termos da distribuição de classe dos registros antes e depois da divisão. Existem diferentes critérios de seleção e esta é uma das principais variações entre os diferentes algoritmos de indução de árvores de decisão (TAN; STEINBACH; KUMAR, 2006).

Estes critérios de seleção podem se basear em diferentes medidas em relação ao conjunto de dados, tais como impureza, distância e dependência. Contudo, a maioria dos algoritmos de indução de árvores de decisão tem por objetivo dividir os dados de determinado nó-pai de maneira a minimizar o grau de impureza dos nós-filhos. Ou seja, quanto menor o grau de impureza, mais desbalanceada é a distribuição de classe. Diz-se, para um dado nó, que a impureza é nula se todos os registros dele pertencerem a mesma classe e, de forma análoga, o grau de impureza é máximo se ocorrer o mesmo número de registros para cada classe possível (TAN; STEINBACH; KUMAR, 2006).

O algoritmo ID3 (QUINLAN, 1986), pioneiro na indução de árvores de decisão, utiliza como medida para o critério de seleção o Ganho de Informação, o qual baseia-se na entropia como medida de impureza, ou seja, a entropia quantifica a variação de um conjunto de exemplos em relação aos valores do atributo alvo (classe), conforme descreve a Equação 1 (QUINLAN, 1986) (TAN; STEINBACH; KUMAR, 2006).

$$\text{entp}(t) = -\sum_{i=1}^{c} p(i|t).\log_2 p(i|t) \tag{1}$$

onde $p(i|t)$ é a fração dos registros pertencentes à classe i no nó t e c é o número de classes.

Dessa forma, com o objetivo de determinar quão bem um atributo divide os exemplos, o Ganho de Informação calcula a diferença entre o grau de entropia do nó-pai antes da divisão com o grau de entropia dos nós-filhos após a divisão, conforme descreve a Equação 2. O atributo que atingir a maior diferença é escolhido como condição de teste naquele determinado nó da árvore (QUINLAN, 1986) (TAN; STEINBACH; KUMAR, 2006).

$$\text{ganho} = \text{entp(pai)} - \sum_{j=1}^{k} \frac{N(t_j)}{N}\text{entp}(t_j) \tag{2}$$

onde k é o número de nós-filhos, ou seja, o número de valores distintos para o atributo, N é o número total de exemplos do nó-pai e $N(t_j)$ é o número de exemplos associado ao

nó-filho t_j.

Como já mencionado, este critério seleciona como melhor atributo para divisão dos exemplos aquele que maximiza o Ganho de Informação, o que pode gerar um grande problema, uma vez que este critério dá preferência a atributos com muitos valores possíveis, como por exemplo, um atributo identificador. Embora irrelevante para o aprendizado e construção da árvore, na seleção segundo o critério do Ganho de Informação este atributo geraria um nó-filho para cada valor possível, ou seja, igual ao número de identificadores (QUINLAN, 1986) (TAN; STEINBACH; KUMAR, 2006).

Cada um desses nós-filhos teria apenas um registro vinculado, o qual pertence a uma única classe, o que resultaria em um valor de grau de entropia mínimo, pois a cada nó todos os exemplos (neste caso, somente um) pertencem à mesma classe. Assim, com todas os graus de entropia mínimo dos nós-filhos, este atributo identificador apresentaria um ganho de informação máximo, mesmo que sua contribuição no conjunto de dados seja inútil, e ele seria escolhido como o melhor atributo para divisão dos exemplos (QUINLAN, 1986) (TAN; STEINBACH; KUMAR, 2006).

Com o intuito de solucionar o problema do ganho de informação, uma nova medida para o critério de seleção de atributo foi proposta juntamente com o algoritmo C4.5, uma versão aprimorada do algoritmo ID3 (QUINLAN, 1993). A nova medida, chamada de Razão do Ganho, é calculada através da média ponderada do ganho em relação a entropia em determinado nó, conforme descreve a Equação 3 (TAN; STEINBACH; KUMAR, 2006).

$$\text{razao}(t) = \frac{\text{ganho}}{\text{entp}(t)} \tag{3}$$

onde t é o nó sendo avaliado.

É importante destacar, pela análise da Equação 3, que a razão do ganho favorece os atributos cuja entropia (denominador) apresente valores pequenos, desde que maiores do que zero. Pois, a razão do ganho torna-se indefinida quando o denominador é igual a zero, prevenindo assim o problema do ganho da informação citado anteriormente (TAN; STEINBACH; KUMAR, 2006).

Em relação ao algoritmo C4.5, cabe caracterizar que ele lida tanto com atributos categóricos (ordinais ou não-ordinais) quanto com atributos contínuos (QUINLAN, 1996). Para lidar com atributos contínuos, o algoritmo C4.5 define um limiar e então divide os exemplos de forma binária: aqueles cujo valor do atributo é maior que o limiar e aqueles cujo valor do atributo é menor ou igual ao limiar (TAN; STEINBACH; KUMAR, 2006). Além disso, o algoritmo trata valores desconhecidos para atributos, desconsiderando-os nos cálculos de entropia e ganho de informação (QUINLAN, 1993) (HAN; KAMBER; PEI, 2011).

Quanto ao método de poda da árvore, que tem o intuito de detectar e eliminar ramos

e subárvores que representem erros ou ruídos nos dados, a fim de melhorar a taxa de acerto do modelo de classificação para novos exemplos, o algoritmo C4.5 utiliza o método chamado de pós-poda (QUINLAN, 1993). Esse método de poda faz uma busca, de baixo para cima, após a árvore de decisão ser construída e transforma em nós-folha aqueles ramos que não apresentam ganho significativo, representando a classe mais frequente no ramo eliminado (HAN; KAMBER; PEI, 2011).

4.3.2 Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) são técnicas ou sistemas computacionais que constroem modelos matemáticos que objetivam emular um sistema neural biológico simplificado, com capacidades de aprendizado, generalização, associação e abstração (HAYKIN, 2001) (HAN; KAMBER; PEI, 2011).

As definições e conceitos computacionais apresentados sobre as redes neurais artificiais neste trabalho se baseiam nos trabalhos de (ROSENBLATT, 1958) e (RUMELHART; HINTON; WILLIAMS, 1986). De acordo com o modelo de rede *Perceptron* e seu algoritmo de treinamento, onde foi instituído os pesos ajustáveis nas conexões entre os neurônios, foi possível criar redes neurais passíveis de treinamento para tarefas de classificação (ROSENBLATT, 1958). Décadas mais tarde, através de um algoritmo de retropropagação para o treinamento, foi elaborada a rede neural *Perceptron* de Múltiplas Camadas (do inglês, *Multi Layer Perceptron* - MLP), onde a ideia do uso de várias camadas foi adicionada ao modelo da rede *Perceptron* (RUMELHART; HINTON; WILLIAMS, 1986).

Uma importante propriedade das redes neurais é a sua habilidade de aprender a partir de exemplos do ambiente no qual está inserida, ou ambiente de aprendizado, e assim melhorar o seu desempenho através da aprendizagem. O aprendizado em uma RNA ocorre por experiência, ou seja, diretamente a partir dos dados, através de um processo de repetidas apresentações dos dados à rede neural (HAYKIN, 2001).

De forma similar ao cérebro humano, uma rede neural artificial apresenta uma estrutura paralelizada composta por várias unidades de processamento de funcionamento bastante simples (neurônios artificiais) conectadas entre si. As conexões entre as unidades de processamento estão associadas a determinados pesos que, quando tem seus valores representativos alterados, influenciam o resultado de saída da rede neural (HAN; KAMBER; PEI, 2011) (HAYKIN, 2001).

A topologia de uma rede neural, dependendo do problema a ser tratado e dos dados envolvidos, pode variar. No entanto, especialmente nas aplicações em tarefas de mineração de dados, utiliza-se a seguinte topologia, dividida em camadas, conforme ilustra a Figura 11: a camada de entrada, na qual os dados são inseridos na rede; as camadas intermediárias (ou ocultas), nas quais é realizado grande parte do processamento e o aprendizado sobre os dados; e, por fim, a camada de saída, na qual o resultado é concluído

e apresentado ao meio externo à rede neural (GOLDSCHMIDT; PASSOS, 2005) (HAN; KAMBER; PEI, 2011).

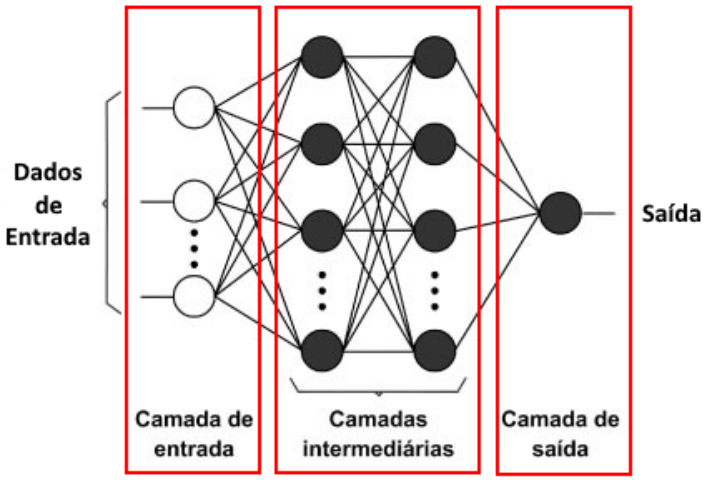


Figura 11: Representação de uma rede neural artificial hipotética, com múltiplas camadas em destaque - adaptada de (HAYKIN, 2001)

Sob o ponto de vista das tarefas de mineração de dados, em redes neurais com aprendizado supervisionado, a entrada corresponde aos atributos preditivos do conjunto de dados enquanto a saída da rede corresponde ao atributo alvo do problema ou classe, para os quais se deseja a construção de um modelo de classificação (HAN; KAMBER; PEI, 2011).

As redes *Perceptron* de Múltiplas Camadas têm como unidade básica de processamento o modelo matemático do neurônio artificial descrito em (MCCULLOCH; PITTS, 1943). Estas unidades de processamento são distribuídas na camada de entrada, camadas intermediárias e camada saída, de forma que cada unidade está conectada a todas as unidades da camada anterior (HAYKIN, 2001).

Formalmente, para cada neurônio artificial, é calculado o produto interno das conexões de entrada aplicadas com seus respectivos pesos, além de ser incorporada uma polarização aplicada externamente. A polarização é importante quando a soma ponderada dos neurônios da camada anterior é igual a zero. Então, a soma ponderada resultante do neurônio, chamada de nível de atividade interna ou potencial de ativação, é aplicada a uma função de ativação, que pode ser a saída da rede neural ou a entrada de outros neurônios da próxima camada. As funções de ativação mais utilizadas são as funções: linear, sigmóide tangente hiperbólica, dentre outras (HAYKIN, 2001).

O teorema da aproximação universal para as redes MLP (HORNIK; STINCHCOMBE; WHITE, 1989), embora não demonstre uma maneira de escolher o número de neurônios necessários para obter a aproximação de uma função, afirma que sempre existe uma rede MLP de três camadas capaz de aproximar qualquer função não-linear e contínua. Além disso, sabe-se que quanto mais complexa a função para aproximação

através da rede neural, mais unidades ocultas de processamento, aquelas pertencentes às camadas intermediárias, são necessárias (HAYKIN, 2001).

As falhas mais comuns na obtenção de modelos baseados em redes neurais artificiais estão relacionadas à escolha do número de neurônios necessários na rede (HORNIK; STINCHCOMBE; WHITE, 1989). Uma solução para a questão do tamanho da rede neural é o teste por tentativa e erro, até que se obtenha um nível arbitrário de aproximação. No entanto, é importante observar que o desempenho de uma rede não deve ser medido em função da sua quantidade de neurônios, e sim pela sua capacidade de mapeamento dos dados e consequente generalização (HAN; KAMBER; PEI, 2011).

O algoritmo de retropropagação do erro (do inglês *Back-Propagation*) é o algoritmo de aprendizado supervisionado base para as redes neurais MLP. O principal objetivo do algoritmo de retropropagação é minimizar a função de erro entre a saída gerada pela rede neural e a saída real desejada, utilizando o método do gradiente descendente (HAN; KAMBER; PEI, 2011) (HAYKIN, 2001).

O método do gradiente descendente é um método de otimização numérica que busca o mínimo local de uma função usando a informação local do gradiente e fazendo com que a busca dirija-se na direção negativa do gradiente, indicada pela informação contida no seu vetor gradiente (HAN; KAMBER; PEI, 2011).

De maneira informal, a minimização do erro é realizada através da estimativa de erro, ou distância, entre a saída produzida pela rede e a saída desejada. A estimativa de erro de saída é calculada e esta é retroalimentada para as camadas intermediárias, possibilitando o ajuste do valor dos pesos das conexões da rede neural, a fim de tornar a saída real tão próxima quanto seja possível da saída desejada (HAN; KAMBER; PEI, 2011) (HAYKIN, 2001).

Como é possível concluir, devido às características aqui apresentadas, modelos neurais baseados em redes MLP com o algoritmo de retropropagação são muito úteis para tarefas de classificação, mesmo a rede neural sendo considerada uma técnica *caixa-preta*, pois o conhecimento identificado nos dados estão codificados internamente no equacionamento do modelo (HAN; KAMBER; PEI, 2011).

4.3.3 Máquina de Vetores de Suporte

As Máquinas de Vetores de Suporte (do inglês, *Support Vector Machines* - SVMs) constituem uma importante técnica de aprendizado que tem apresentado destaque na comunidade de aprendizado de máquina. Os resultados da aplicação dessa técnica são comparáveis e até mesmo superiores aos obtidos por outros métodos já consagrados, como as redes neurais artificiais (HAN; KAMBER; PEI, 2011).

Este método de aprendizado é baseado na Teoria de Aprendizado Estatístico (VAPNIK, 1998), a qual estabelece uma série de princípios (condições matemáticas) que devem ser seguidos na obtenção de classificadores com boa capacidade de generalização, ou

seja, que apresente bom desempenho em prever corretamente as classes a que pertencem novos dados do mesmo domínio onde ocorreu o aprendizado (BURGES, 1998).

A técnica de SVM foi originalmente concebida para lidar com classificações binárias, motivo pelo qual as definições, conceitos e exemplos tratados nesta seção apresentam classes binárias, entretanto a maior parte dos problemas reais requer múltiplas classes (TAN; STEINBACH; KUMAR, 2006). Para se utilizar uma SVM para classificar múltiplas classes é necessário transformar o problema original em vários problemas de classes binárias, como demonstrado em (CHANG; LIN, 2011).

O objetivo das SVMs é encontrar um hiperplano, tipo especial de modelo linear, que separe corretamente dois conjuntos de pontos linearmente separáveis no espaço. Aqueles pontos mais próximos do hiperplano são os chamados vetores de suporte. Existe sempre, pelo menos, um vetor de suporte para cada classe. A Figura 12 ilustra um exemplo de um hiperplano separador de pontos de duas classes distintas (HAN; KAMBER; PEI, 2011) (BURGES, 1998).

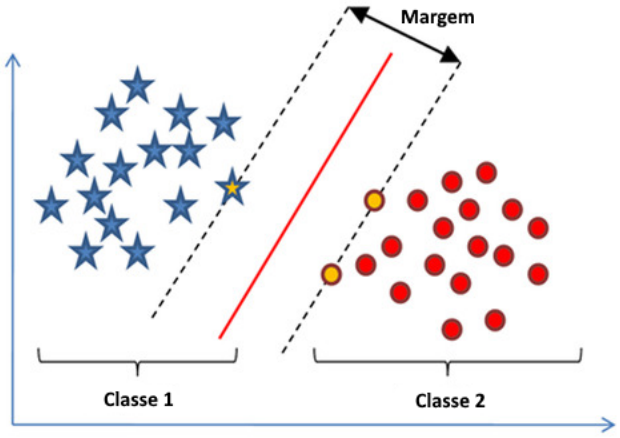


Figura 12: Representação de um hiperplano separador entre os pontos de classes binárias, com destaque para os pontos utilizados como vetores de suporte - adaptada de (HAN; KAMBER; PEI, 2011)

É possível concluir que podem existir infinitos hiperplanos que separam dois conjuntos de pontos linearmente separáveis no espaço. Além disso, já foi demonstrado que o hiperplano cuja margem para os pontos mais próximos apresenta a maior distância é o hiperplano que minimiza o risco de se classificar erroneamente um novo ponto (HAN; KAMBER; PEI, 2011). O desafio da técnica de SVM é, portanto, encontrar o hiperplano ótimo que tenha a maior margem para os pontos mais próximos a ele, o que resulta em um problema de otimização quadrático, com restrições, solucionado através de ampla teoria matemática já estabelecida em (SCHÖLKOPF; SMOLA, 2002).

As SVMs, em sua concepção original e formal, são ditas de margem rígida, umas vez que restrições são impostas de maneira a assegurar que não haja dados de treinamento

entre as margens de separação das classes, conforme ilustra a Figura 13-A. Contudo, em situações reais, os dados geralmente não são linearmente separáveis, devido a presença de ruídos e *outliers* ou devido à própria natureza do problema, que pode ser não-linear. Assim, as SVMs lineares de margens rígidas são estendidas para lidar com conjuntos de treinamento mais gerais, permitindo que alguns dados possam violar a restrição da margem. A aplicação desse procedimento suaviza as margens do classificador linear, permitindo que alguns dados permaneçam entre os hiperplanos referentes aos vetores de suporte (hiperplanos que demarcam a margem de cada classe) e, permitem também, a ocorrência de alguns erros de classificação, conforme ilustra a Figura 13-B. Por esse motivo, as SVMs obtidas neste caso também podem ser referenciadas como SVMs com margens suaves (HAN; KAMBER; PEI, 2011) (TAN; STEINBACH; KUMAR, 2006).

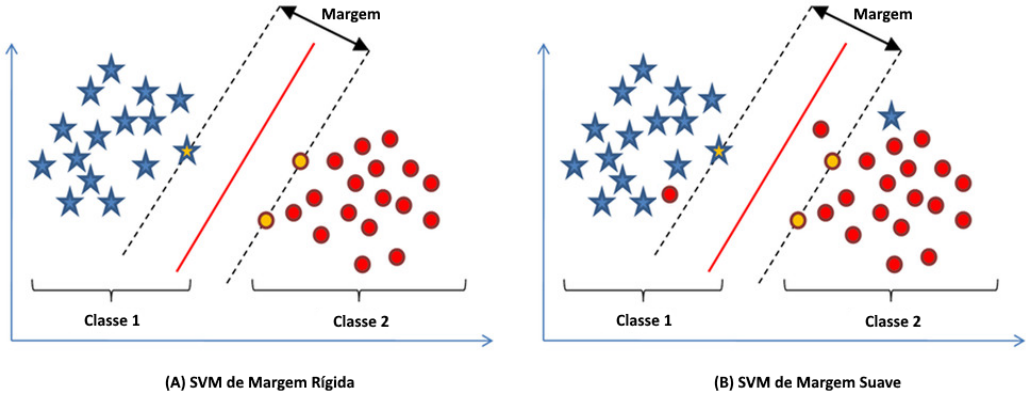


Figura 13: (A) Representação de uma SVM de Margem Rígida. (B) Representação de uma SVM de Margem Suave - adaptadas de (HAN; KAMBER; PEI, 2011)

Entretanto, é importante considerar que em muitos problemas reais as classes não são linearmente separáveis mesmo utilizando a folga instituída pelas SVMs de margens suaves. Nestes casos, então, a abordagem utilizada pela SVM para resolver esse tipo de problema consiste em mapear os dados para um espaço de dimensão maior, ou seja, a técnica de SVM utiliza uma função *kernel* para mapear os dados em um espaço diferente em que um hiperplano pode ser utilizado para fazer a separação linear. A Figura 14 demonstra um exemplo de dados não-linearmente separáveis (A) que foram submetidos ao um mapeamento segundo uma função *kernel*, para então ser possível a separação linear em um novo espaço (B) (BURGES, 1998).

O conceito de utilização de uma função *kernel* é importante porque, além de mapear os dados de maneira a permitir a utilização de um hiperplano separador, permite que o modelo realize separações mesmo com fronteiras bastante complexas entre os dados. A SVM possui quatro tipos de funções *kernel*: linear, quadrática, polinomial e função de base radial. Cada função possui seus respectivos parâmetros que devem ser determinados pelo analista. A Função de Base Radial (FBR) é uma das mais utilizadas pois apresenta os

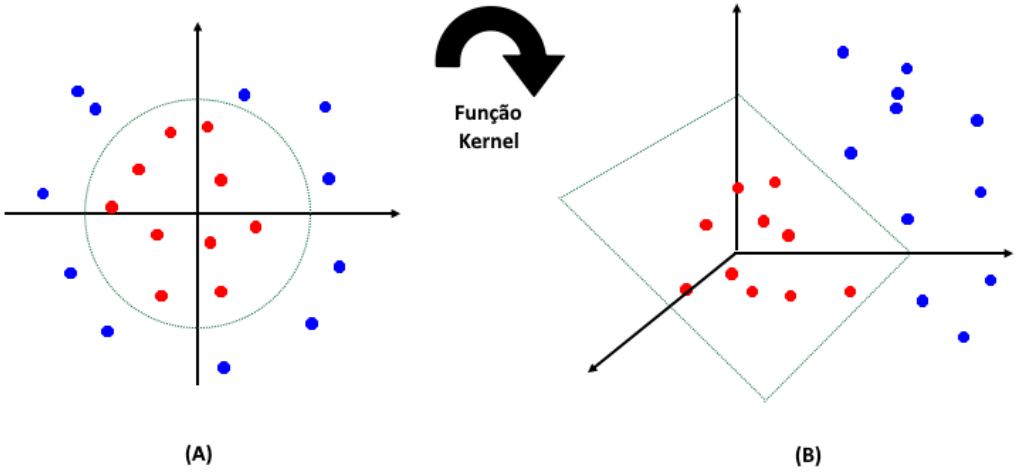


Figura 14: (A) Representação de dados não-linearmente separáveis por uma SVM. (B) Representação dos mesmos dados, linearmente separáveis, após o mapeamento segundo uma função *kernel* - adaptadas de (BURGES, 1998)

melhores resultados na separação ótima de classes, e também porque as demais funções são variações da própria FBR (BURGES, 1998).

Por fim, cabe ressaltar que, da mesma forma que as redes neurais, a técnica de SVM é considerada *caixa-preta*, ou seja, o conhecimento extraído dos dados através desta técnica encontra-se codificado em equações de difícil interpretação, ao contrário de outros modelos de classificação, como por exemplo, do modelo gerado por técnicas de árvores de decisão (HAN; KAMBER; PEI, 2011).

5 AVALIAÇÃO DE DESEMPENHO

O grau de relevância das informações adquiridas no processo de KDD pode ser observado por meio das avaliações de desempenho dos classificadores, geralmente representada pelas taxas de acerto e de erros resultantes da classificação (FRANK et al., 2004).

Uma classificação é dita correta quando o classificador indica para determinado exemplo a mesma classe que está indicada em seu atributo rótulo. De forma análoga, o erro ocorre quando a classe indicada pelo classificador difere da classe a qual o exemplo pertence. O detalhamento desse desempenho é mostrado através da matriz de confusão, onde as contagens estão tabuladas em uma matriz bidimensional (HAN; KAMBER; PEI, 2011).

A matriz de confusão possui uma linha e uma coluna para cada classe existente. Cada elemento da matriz mostra o número de exemplos de teste para o qual a classe representada na linha é a classe real e a classe representada na coluna é a classe na qual o exemplo foi classificado. Ou seja, a matriz de confusão evidencia, para cada classe, o número de classificações corretas em relação ao número de classificações indicadas pelo modelo (HAN; KAMBER; PEI, 2011).

Embora a matriz de confusão forneça as informações para determinar o desempenho de classificação dos modelos, para a comparação entre diferentes modelos, é conveniente a existência de uma medida de valor numérico único, que represente a sumarização do desempenho do modelo, tal como a acurácia ou a taxa de erros (TAN; STEINBACH; KUMAR, 2006).

A acurácia de um modelo de classificação para um determinado conjunto de dados é a sua taxa de acertos, ou seja, taxa de amostras que o modelo classifica de maneira correta em relação a todas as classificações realizadas sobre o conjunto de teste. De forma equivalente, o desempenho pode ser medido considerando-se a taxa de erros, quando a métrica é tratada de forma contrária, computando a taxa de amostras que o modelo classifica incorretamente em relação ao total de classificações (TAN; STEINBACH; KUMAR, 2006).

Em relação a avaliação de um classificador, além da acurácia preditiva, existem outros parâmetros que podem ser considerados. Tais critérios não são comumente utilizados pois

nem todos são quantitativamente calculáveis. Mas, a critério do analista e do especialista do domínio, mesmo os critérios qualitativos podem e devem ser levados em consideração na avaliação da performance de classificadores (TAN; STEINBACH; KUMAR, 2006).

Estes outros parâmetros que podem ser analisados são: a velocidade, a qual refere-se aos custos computacionais para construção e uso do modelo; a robustez, que refere a habilidade do modelo em realizar predições corretas mesmo na existência de dados com ruídos ou ausentes; a escalabilidade, que consiste na capacidade de construir modelo eficientes mesmo na presença de grandes quantidades de dados; e, por último, a interpretabilidade, a qual refere-se ao nível de informação útil e conhecimento descoberto e extraído pelo modelo que é passível de ser interpretado pelo ser humano (HAN; KAMBER; PEI, 2011).

5.1 Métricas de Avaliação

As métricas de avaliação abordadas inicialmente na seção introdutória do capítulo são melhor explicadas a seguir. Além disso, métricas mais elaboradas para a avaliação de modelos de classificação também são abordadas. É importante destacar que todas as métricas são calculadas com base nas contagens de acerto ou erro de classificação contidas na matriz de confusão (HAN; KAMBER; PEI, 2011).

- *Matriz de Confusão*

  A matriz de confusão de um classificador oferece uma medida efetiva do modelo de classificação ao mostrar o número de classificações corretas versus as classificações preditas para cada classe, dispondo os resultados em duas dimensões: classes verdadeiras e classes preditas (TAN; STEINBACH; KUMAR, 2006).

  Formalizando a construção de uma matriz de confusão, conforme a Tabela 1, para um conjunto de dados T, com k classes diferentes $\{C_1, C_2, ..., C_k\}$: cada elemento $M(C_i, C_j)$ da matriz, sendo $i, j = 1, 2, ..., k$, representa o número de exemplos do conjunto de dados que realmente pertencem à classe C_i, mas foram classificados como sendo pertencentes à classe C_j (MONARD; BARANAUSKAS, 2003).

  O número de acertos, para cada classe, localiza-se na diagonal principal $M(C_i, C_j)$, para $i = j$. Os demais elementos $M(C_i, C_j)$, para $i \neq j$, representam erros na classificação. Dessa forma, a matriz de confusão de um classificador ideal possuiria todos estes elementos iguais a zero, pois nenhum erro de classificação seria cometido (MONARD; BARANAUSKAS, 2003).

  Para abordar os cálculos das demais métricas, por simplicidade, considera-se uma matriz de confusão para um problema de duas classes chamadas de positivo C_+ e negativo C_-, ou seja, uma estrutura de predição de ocorrência ou não de dado evento, conforme Tabela 2. Neste caso, os dois erros possíveis são chamados de

classe	predita C_1	predita C_2	...	predita C_k
verdadeira C_1	$M(C_1, C_1)$	$M(C_1, C_2)$	...	$M(C_1, C_k)$
verdadeira C_2	$M(C_2, C_1)$	$M(C_2, C_2)$	...	$M(C_2, C_k)$
...	...	...	...	...
verdadeira C_k	$M(C_k, C_1)$	$M(C_k, C_2)$	...	$M(C_k, C_k)$

Tabela 1: Exemplo formal de uma matriz de confusão para um classificador de um conjunto de dados com k classes - adaptada de (MONARD; BARANAUSKAS, 2003)

falso positivo (F_p) e falso negativo (F_n). Por outro lado, o número de exemplos positivos e exemplos negativos classificados corretamente, respectivamente, são chamados de V_p e V_n. E, por fim, o total de exemplos é dado por $n = V_p + V_n + F_p + F_n$ (MONARD; BARANAUSKAS, 2003).

classe	predita C_+	predita C_-
verdadeira C_+	V_p	F_n
verdadeira C_-	F_p	V_n

Tabela 2: Exemplo de uma matriz de confusão para um classificador de um conjunto de dados com duas classes: positivo C_+ e negativo C_- - adaptada de (MONARD; BARANAUSKAS, 2003)

- *Acurácia e Taxa de Erro*

  A acurácia e a taxa de erro são métricas básicas para avaliação do desempenho de classificadores. A acurácia consiste na taxa de acertos do classificador, ou seja, na taxa de exemplos positivos e negativos corretamente classificados dentre todos os exemplos do conjunto de dados, conforme descreve a Equação 4. Já a taxa de erro consiste na taxa de registros positivos e negativos classificados incorretamente dentre todos os exemplos existentes, conforme a Equação 5 (HAN; KAMBER; PEI, 2011).

  É importante observar que na presença de um conjunto de dados que apresente classes desbalanceadas, ou seja, número muito desigual entre registros de diferentes classes, a acurácia e a taxa de erro tornam-se medidas não confiáveis. Isso se explica pelo fato do viés do modelo em classificar corretamente a classe majoritária e assim obter uma alta acurácia e uma baixa taxa de erros, mesmo não conseguindo classificar nenhum item da classe minoritária (TAN; STEINBACH; KUMAR, 2006).

$$Ac = \frac{V_p + V_n}{n} \tag{4}$$

$$Er = \frac{F_p + F_n}{n} \tag{5}$$

- *Precisão e Revocação*

Em complemento às métricas de acurácia e taxa de erro, outras medidas também são utilizadas na avaliação: Precisão (*Precision*) e Revocação (*Recall*). Uma vez que estas métricas são mais suscetíveis ao desbalanceamento entre as classes, podem ser consideradas métricas mais adequadas na presença de classes desbalanceadas e, portanto, mais confiáveis para a avaliação de classificadores (TAN; STEINBACH; KUMAR, 2006).

A precisão é a taxa de exemplos corretamente classificados como positivo (V_p) dentre todos os exemplos classificados como positivo (V_p e F_p), conforme descreve a Equação 6, ou seja, indica a proporção (ou probabilidade) de, caso um exemplo seja classificado como de uma determinada classe por um classificador, ele realmente pertença a esta determinada classe (TAN; STEINBACH; KUMAR, 2006) (HAN; KAMBER; PEI, 2011).

Ainda é possível dizer que a precisão é a capacidade do classificador em reconhecer as instâncias de uma classe de interesse e rejeitar as demais (MONARD; BARANAUSKAS, 2003).

$$\text{Prec} = \frac{V_p}{V_p + F_p} \tag{6}$$

A revocação (*recall*) é a taxa de exemplos corretamente classificados como positivo (V_p) dentre todos os exemplos que realmente são positivos (V_p e F_n), conforme descreve a Equação 7, ou seja, indica a proporção (ou probabilidade) de um exemplo de determinada classe ser classificado como tal (TAN; STEINBACH; KUMAR, 2006) (HAN; KAMBER; PEI, 2011).

Em outras palavras, revocação é a capacidade do classificador em reconhecer todas as instâncias de uma classe de interesse (MONARD; BARANAUSKAS, 2003).

$$\text{Revoc} = \frac{V_p}{V_p + F_n} \tag{7}$$

- *F-Measure*

  A *F-Measure* (*Medida-F*) é a média harmônica entre as medidas de Precisão e Revocação, conforme descreve a Equação 8. Tais medidas, quando examinadas separadamente, podem ser enganosas, já que uma precisão elevada geralmente indica sacrificar um bom resultado de revocação e vice-versa. Dessa forma, através da média harmônica entre ambas, é possível obter uma avaliação de desempenho mais realista (TAN; STEINBACH; KUMAR, 2006) (HAN; KAMBER; PEI, 2011).

  Cabe destacar que a média harmônica entre dois números tende a ser próxima ao mínimo entre estes números, portanto a *F-Measure* somente será elevada na

presença de medidas de precisão e revocação razoavelmente altas (TAN; STEINBACH; KUMAR, 2006) (HAN; KAMBER; PEI, 2011).

$$\text{Fmeasure} = \frac{2 * \text{Prec} * \text{Revoc}}{\text{Prec} + \text{Revoc}} = \frac{2}{\frac{1}{\text{Prec}} + \frac{1}{\text{Revoc}}} \tag{8}$$

- *Índice Kappa*

  O índice *Kappa* mede a fração de concordância observada entre as classes preditas por um classificador e as classes verdadeiras. É uma medida de concordância ajustada, ou seja, informa a proporção de concordância não aleatória, além da concordância já esperada pelo acaso. É considerada uma maneira de expressar a confiabilidade de um classificador (TAN; STEINBACH; KUMAR, 2006).

  Este índice utiliza todos os elementos da matriz de confusão (Tabela 1) no seu cálculo: é uma medida da concordância real (indicada pelos elementos diagonais da matriz de confusão) menos a concordância por chance (indicada pelo produto entre os totais marginais da matriz de confusão) (TAN; STEINBACH; KUMAR, 2006), conforme exposto na Equação 9.

$$\text{Kappa} = \frac{n \sum_{i=1}^{r} M(C_i, C_i) - \sum_{i=1}^{r} M(C_i, *)M(*, C_i)}{n^2 - \sum_{i=1}^{r} M(C_i, *)M(*, C_i)} \tag{9}$$

  onde n é o número total de instâncias, r é o número total de classes, $M(C_i, C_i)$ é o total de instâncias corretamente classificadas, $M(C_i, *)$ é o total de instâncias preditas como da classe i e $M(*, C_i)$ é o total de instâncias da classe i.

  Os valores do índice *Kappa* podem variar de -1 até $+1$. Quanto maior o valor, mais forte é a concordância, conforme indica a Tabela 3 com a escala completa dos intervalos de valores e sua concordância. O significado de alguns casos específicos do índice são (HAN; KAMBER; PEI, 2011) (LANDIS; KOCH, 1977):

  - *Kappa* $= 1$: a concordância é perfeita;
  - *Kappa* $= 0$: a concordância é a mesma que seria esperada pelo acaso;
  - *Kappa* < 0: a concordância é pior que o esperado pelo acaso, o que raramente ocorre.

  Normalmente, conforme consenso dos estatísticos, prefere-se valores de índice *Kappa* maiores que 0,6, sendo ideal aqueles superiores a 0,7. Este índice é considerado como uma medida apropriada da exatidão de um classificador, pois ele representa inteiramente a matriz de confusão, ou seja, tanto as classificações corretas quanto as incorretas são consideradas no cálculo (LANDIS; KOCH, 1977) (GOLDSCHMIDT; PASSOS, 2005).

Índice Kappa	Concordância
$<$ 0,00	Nenhuma
0,00 - 0,20	Ruim
0,21 - 0,40	Fraca
0,41 - 0,60	Média
0,61 - 0,80	Boa
0,81 - 0,99	Excelente

Tabela 3: Escala de concordância do índice *Kappa* - adaptada de (LANDIS; KOCH, 1977)

5.1.1 Fenômeno de Overfitting

Após o entendimento sobre a matriz de confusão, as informações que ela demonstra e os cálculos das taxas de acerto e de erro é importante ressaltar alguns conceitos sobre os tipos de erro existentes e o que representam (HAN; KAMBER; PEI, 2011).

Existem dois tipos de erros considerados nas tarefas de classificação: erro de treinamento e erro de generalização. O erro de treinamento, ou erro aparente, é o número de classificações incorretamente realizadas sobre os registros de treinamento, ao passo que o erro de generalização é o erro estimado do modelo para registros desconhecidos, baseado nas classificações realizadas sobre o conjunto de teste (TAN; STEINBACH; KUMAR, 2006).

Após a abordagem das definições dos tipos de erros pode-se definir a qualidade de um classificador em relação a ambos: considera-se bom um modelo de classificador quando o mesmo se ajusta bem aos dados de treinamento e também possui boa acurácia para classificar exemplos não vistos. Em outras palavras, um bom classificador deve apresentar tanto um erro de treinamento quanto um erro de generalização baixos. Isso é importante pois nem sempre um modelo bem ajustado aos dados de treinamento, também apresenta um baixo erro de generalização (TAN; STEINBACH; KUMAR, 2006).

Dessa forma, é importante ressaltar que ao induzir um classificador podem ocorrer os fenômenos de *overfitting* ou *underfitting*. O primeiro fenômeno ocorre quando o classificador é muito específico para o conjunto de treinamento utilizado, ou seja, quando se ajusta excessivamente ao conjunto de treinamento (apresentando erro de treinamento baixo) mas não consegue obter bom desempenho para o conjunto de teste (erro de generalização alto). Por outro lado, o segundo fenômeno ocorre quando o classificador se ajusta muito pouco ao conjunto de treinamento, não obtendo bom desempenho tanto para os dados de treino quanto para os de teste (TAN; STEINBACH; KUMAR, 2006) (HAN; KAMBER; PEI, 2011).

5.1.2 Estimativa do Erro de Generalização

Apesar da principal razão de ocorrer o fenômeno de *overfitting* ainda ser objeto de debate, há um consenso de que o nível de complexidade do modelo tem impacto sobre esse ajuste excessivo, evidenciando que determinar a complexidade correta de um modelo

tem grande importância (TAN; STEINBACH; KUMAR, 2006).

Sabe-se que a complexidade ideal é a do modelo que apresentar o menor erro de generalização, porém durante a construção do classificador o algoritmo de aprendizagem só tem acesso aos dados de treinamento (responsáveis por produzir o erro aparente). Dessa forma, a melhor maneira de conseguir determinar a complexidade correta do modelo é estimando o seu erro de generalização (HAN; KAMBER; PEI, 2011).

Dentre os principais métodos que realizam a estimativa do erro de generalização na construção de modelos de classificação estão aqueles que utilizam as abordagens de: erro de resubstituição, incorporar a complexidade do modelo, limites estatísticos e conjunto de validação. Todos os métodos amplamente abordados em (TAN; STEINBACH; KUMAR, 2006).

5.2 Métodos de Avaliação

Conforme já mencionado anteriormente, o erro de generalização de um modelo é estimado através de vários métodos durante o processo de treinamento. Esta estimativa de erro contribui para os algoritmos de aprendizagem realizarem a seleção de modelos, ou seja, encontrar um modelo com a complexidade correta que não seja suscetível ao ajuste excessivo, chamado de (*overfitting*) (TAN; STEINBACH; KUMAR, 2006).

Uma vez construído, o modelo de classificação pode ser aplicado ao conjunto de dados de teste para classificar registros não vistos anteriormente. É muito útil medir o desempenho do modelo no conjunto de teste, pois tal medida fornece uma boa estimativa do erro de generalização do modelo (TAN; STEINBACH; KUMAR, 2006) (HAN; KAMBER; PEI, 2011).

Outro fator bem importante é que as métricas de desempenho calculadas a partir do conjunto de teste podem ser utilizadas para comparar o desempenho relativo de classificadores para um mesmo domínio, desde que se conheça a classe a que pertencem os registros contidos no conjunto de teste (TAN; STEINBACH; KUMAR, 2006).

A seguir, são apresentados os métodos mais utilizados para estimar e avaliar o desempenho de modelos de classificação. São métodos que aplicam técnicas para a divisão do conjunto de dados em conjuntos de treinamento e teste a fim de se maximizar a obtenção de boas estimativas de medidas de desempenho (HAN; KAMBER; PEI, 2011).

- *Holdout*

  Neste método, divide-se o conjunto de dados em dois conjuntos disjuntos, um para o treinamento e outro para os testes. O classificador é induzido com o conjunto de treinamento e depois seu desempenho é medido através do conjunto de testes. Dessa forma, as métricas de desempenho do modelo de classificação são estimadas baseando-se nas métricas do classificador aplicado nos dados de teste. A proporção

de divisão dos dados fica a critério do analista, podendo ser $50 - 50\%$ ou, usualmente, $2/3$ para o treinamento e $1/3$ para o teste. Este método, porém, apresenta algumas limitações: menos registros estão disponíveis para o treinamento, visto que parte deles ficam retidos para os testes, o que resulta na possibilidade de o modelo não ser tão bom quanto seria utilizando-se todos os registros para o treinamento; outra limitação é referente a possível forte dependência do modelo em relação à composição dos conjuntos de treinamento e de teste, pois quanto menor o conjunto de treinamento maior a variância do modelo, porém, por outro lado, caso o conjunto de treinamento seja muito grande, menos confiável será a estimativa das métricas calculadas no conjunto de teste pequeno (diz-se que é uma estimativa que apresenta um extenso intervalo de confiança); a última limitação refere-se aos conjuntos de treinamento e teste tornarem-se dependentes entre si e, como ambos são subconjuntos dos dados originais, uma classe que está super-representada em um subconjunto, estará automaticamente sub-representada no outro, e vice-versa (HAN; KAMBER; PEI, 2011).

- *Holdout Repetido*

  Esta abordagem consiste em repetir algumas vezes o método *holdout* com o intuito de melhorar a estimativa de desempenho do classificador. As métricas de desempenho do modelo são obtidas a partir da média das métricas das repetições. A metodologia de subamostragem aleatória do *holdout* enfrenta os mesmos problemas associados ao método *holdout*, já que não utiliza o máximo de dados possível no treinamento. Como também não há controle sobre o número de vezes que cada registro é utilizado para treinamento ou teste, alguns registros podem ser mais frequentemente utilizados no treinamento do que outros, gerando uma possibilidade de viés indutivo (TAN; STEINBACH; KUMAR, 2006).

- *Validação Cruzada*

  É um método alternativo ao *holdout*, pois consiste em um processo estatístico de partição do conjunto de dados em subconjuntos disjuntos, ou seja, cada registro é utilizado o mesmo número de vezes para o treinamento e uma única vez para teste. A especificação que generaliza este método é chamada de validação cruzada em k*-fold* (validação cruzada em k partições). Esta técnica consiste em particionar o conjunto de dados em k partições. Uma destas k partições é retida para representar o conjunto de dados de teste, enquanto as demais $k-1$ partições são utilizadas como conjuntos de treinamento do modelo. O processo de validação é então repetido, até ser realizado k vezes, com cada uma das k partições sendo utilizada somente uma vez como conjunto de teste. Ao final, os k resultados são combinados e é calculada a média de todos, para se obter um valor único das métricas de desempenho do classificador. Usualmente opta-se pela partição em 10 subconjuntos (10 partições),

pois essa se mostrou a mais eficiente em testes empíricos. Existe ainda um caso particular de validação cruzada em k partições, chamado de *leave-one-out*, onde o número de partições k é o número de registros do conjunto de dados, assim, cada conjunto de teste contém um único registro. As principais vantagens dessa abordagem são: o máximo de dados possíveis são utilizados para o treinamento e os conjuntos de dados são mutuamente exclusivos e cobrem todo o conjunto de dados. Por outro lado, a grande desvantagem é o custo computacional para realizar o número de repetições conforme o número de registros do conjunto de dados. Além disso, como cada conjunto de teste possui somente um registro, a variância da estimativa da medida de desempenho tende a ser alta (HAN; KAMBER; PEI, 2011) (TAN; STEINBACH; KUMAR, 2006).

- *Bootstrap*

  Os demais métodos assumem que os dados de treinamento são amostras sem reposição dos registros, portanto não há registros duplicados nos conjuntos de treinamento e teste. Já no método *bootstrap* as amostras dos registros para treinamento são criadas com reposição, ou seja, um registro já escolhido para o conjunto de treinamento pode novamente ser escolhido para compor este mesmo conjunto com a mesma probabilidade que os demais registros disponíveis. A amostragem é repetida diversas vezes para gerar tantas amostras quantas forem necessárias. As métricas de desempenho do modelo de classificação são calculadas pela combinação das métricas de cada amostragem, seguindo fórmulas específicas, dependendo da variação de amostragem utilizada. Este método é considerado a melhor maneira de se ter uma estimativa das métricas apenas para pequenos conjuntos de dados (HAN; KAMBER; PEI, 2011).

5.3 Teste Estatístico t-Student

Um teste estatístico pode ser utilizado em situações nas quais é necessário saber se alguma medida é realmente diferente entre dois grupos de valores, ou se essa diferença ocorre meramente ao acaso, ou seja, não há diferença estatística (BUSSAB; MORETTIN, 2010).

Nesse cenário, o *teste t-Student* ou simplesmente *teste-t* é um teste de hipótese que usa conceitos estatísticos para rejeitar ou não uma hipótese nula quando a estatística de teste (*t*) segue uma distribuição *t-Student*. Essa premissa é normalmente usada quando a estatística de teste, na verdade, segue uma distribuição normal, mas a variância da população é desconhecida. Nesse caso, é usada a variância amostral e, com esse ajuste, a estatística de teste passa a seguir uma distribuição *t-Student* (BUSSAB; MORETTIN, 2010).

Alguns conceitos relevantes em relação aos testes de hipóteses são (BUSSAB; MO-

RETTIN, 2010):

- Hipótese nula (h_0): é a hipótese assumida como verdade para a construção do teste. É o efeito ou teoria de interesse que se deseja testar;
- Hipótese alternativa (h_1): é o considerado caso a hipótese nula não tenha evidência estatística que a defenda;
- Erro do tipo I: é a probabilidade de rejeitarmos a hipótese nula quando ela é efetivamente verdadeira;
- Erro do tipo II: é a probabilidade de rejeitarmos a hipótese alternativa quando ela é efetivamente verdadeira.

O *teste-t* pareado, mais especificamente, testa se existem diferenças entre médias de valores quando se tem um mesmo grupo de dados, testados em duas situações distintas. Ou seja, tipicamente um teste de diferenças entre médias, onde a hipótese nula h_0 consiste na afirmação de que não há diferença entre as médias de valores das amostras e, consequentemente, a hipótese alternativa h_1 afirma que há diferença entre as médias dos valores das amostras (VIEIRA, 2015).

Através das médias dos valores e da variância da amostra, o *valor-t* é calculado. Utilizando este *valor-t* e certo nível de significância (normalmente 5%), obtém-se a probabilidade destas duas amostras apresentarem diferença estatisticamente significativa por meio da consulta à tabela de distribuição *t-student* (VIEIRA, 2015).

Dessa forma, a utilização do *teste-t* pareado para diferenças entre médias possibilita afirmar, com certo nível de confiança (dado por: 100 - nível de significância), se as médias dos valores dos dois grupos amostrais são significativamente diferentes ou não, rejeitando-se ou não a hipótese nula h_0 (VIEIRA, 2015).

Concluído o referencial teórico relacionado às tarefas e técnicas do processo de KDD, aos classificadores e algoritmos aplicados no problema da predição da classe de cor em proteínas fluorescentes e às formas de avaliação e comparação das métricas de desempenho destes classificadores, nos próximos capítulos são abordados a metologia proposta para o trabalho, as ferramentas utilizadas para implementação desta metodologia e os resultados obtidos, devidamente discutidos tendo em vista a fundamentação teórica e os objetivos traçados no presente trabalho.

6 METODOLOGIA

Com o intuito de atingir o objetivo deste trabalho, ou seja, comparar métodos de classificação de cor de proteínas fluorescentes a fim de se investigar a performance dos classificadores em predizer a classe de cor de determinada proteína fluorescente a partir de sua estrutura terciária, foi proposta a metodologia a ser abordada neste capítulo, composta de quatro etapas principais, conforme ilustra a Figura 15.

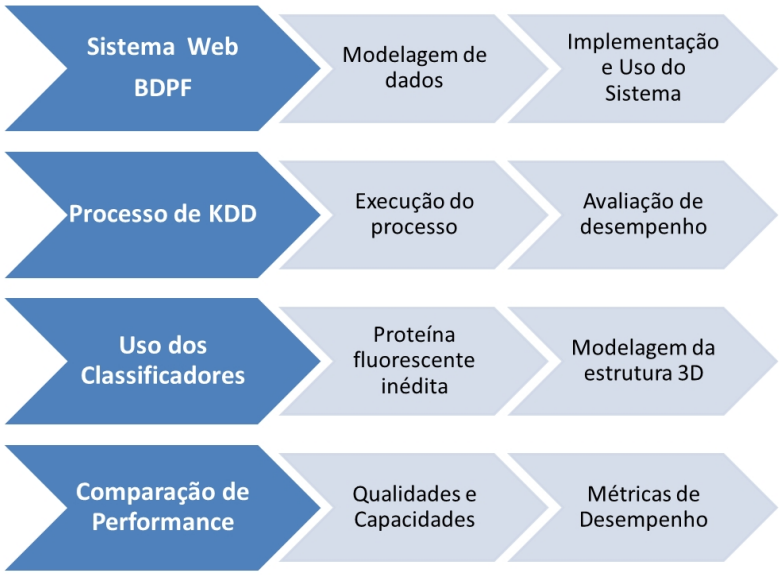


Figura 15: Representação das quatro etapas da metodologia proposta para a comparação de performance dos classificadores - do autor

A primeira etapa da metodologia consiste em desenvolver um sistema de informação para manipular, organizar e armazenar informações relativas às proteínas fluorescentes. Para tanto, três atividades são propostas:

- *modelagem conceitual*: consiste na descrição formal do banco de dados, a qual inclui um modelo relacional, em uma descrição abstrata do conjunto de relacionamentos entre os dados no contexto (ELMASRI; NAVATHE, 2003);
- *implementação*: consiste na implantação do modelo de banco de dados em um

Sistema Gerenciador de Banco de Dados (SGBD) e o desenvolvimento do sistema de informação, para que este sistema de informação proposto possa manipular e gerenciar tais dados (ELMASRI; NAVATHE, 2003);

- *carga*: consiste na inserção de dados no Sistema Gerenciador de Banco de Dados através da interface do sistema de informação (ELMASRI; NAVATHE, 2003).

Em uma seção dedicada inteiramente ao sistema *Banco de Dados de Proteínas Fluorescentes* (BDPF), a seguir, esta primeira etapa da metodologia é abordada de forma mais ampla, contendo mais detalhes em relação a modelagem de dados e ao desenvolvimento e funcionamento do sistema.

Na segunda etapa, a metodologia propõe que, a partir de dados armazenados no sistema *Banco de Dados de Proteínas Fluorescentes*, realize-se o processo de descoberta de conhecimento em bases de dados, com o intuito de construir os classificadores e avaliar o desempenho das técnicas de classificação para a predição da classe de cor de proteínas fluorescentes.

Mais especificamente, esta segunda etapa da metodologia propõe o emprego de três técnicas (AD, RNA e SVM) referentes à tarefa de classificação, as quais utilizam como dados de entrada informações armazenadas no banco de dados do sistema de informação desenvolvido, tais como: a sequência de aminoácidos, propriedades fluorescentes, a classe de cor a que pertence e a estrutura tridimensional das proteínas fluorescentes.

Cabe ressaltar que, além do processo de KDD utilizar como dados de entrada os dados armazenados no sistema BDPF, o próprio sistema realiza as etapas iniciais do processo de KDD, onde ocorre toda a preparação dos dados para o efetivo uso pelos algoritmos de classificação.

A terceira etapa da metodologia objetiva utilizar os classificadores construídos na etapa anterior para a classificação de um exemplo inédito, neste caso, uma proteína fluorescente inédita produzida pelo projeto Peixes Transgênicos Fluorescentes. Nesta etapa também é possível avaliar a acurácia dos classificadores em relação ao exemplo não visto.

Para que seja possível utilizar a proteína fluorescente inédita dessa forma, no entanto, é necessário modelar sua estrutura tridimensional através de ferramentas próprias para tal função, uma vez que ela ainda não foi experimentalmente resolvida.

A quarta e última etapa desta metodologia consiste em realizar a comparação de performance entre os três classificadores. Para tanto, o teste estatístico *t-student* é realizado sobre as métricas de desempenho geradas pelos classificadores. Além disso, considera-se também nesta comparação o desempenho na classificação da proteína fluorescente inédita e as qualidades e capacidades de cada modelo de classificação avaliado.

O resultado esperado nesta metodologia é:

- o desenvolvimento de um sistema de informação online que faça corretamente a

manipulação e armazenamento de dados referentes às proteínas fluorescentes, e realize todas as etapas de preparação dos dados para as técnicas de KDD;

- a construção de modelos de classificação de bom desempenho no processo de KDD segundo as técnicas de avaliação apresentadas no capítulo 5, a partir dos quais seja possível predizer a classe de cor de proteínas fluorescentes a partir de sua estrutura terciária;

- a comparação da performance dos classificadores construídos através do teste estatístico sobre as métricas de desempenho e das qualidades e capacidades de cada modelo de classificação.

6.1 Banco de Dados de Proteínas Fluorescentes

O Banco de Dados de Proteínas Fluorescentes (BDPF) é um sistema de informação web, desenvolvido exclusivamente para este trabalho, utilizando o framework CodeIgniter (GABARDO, 2012), através da linguagem de programação PHP (DALL'OGLIO, 2009) e do sistema gerenciador de banco de dados (SGBD) MySQL (MILANI, 2010).

A principal tarefa desse sistema é permitir o gerenciamento e organização de dados referentes à proteínas fluorescentes, tais como o comprimento de onda emitido, a classe de cor, a intensidade de brilho, dentre outros, a serem tratados de forma completa na seção sobre a modelagem de dados. Além destes dados que caracterizam as proteínas fluorescentes, também são manipulados e armazenados dados relativos à sequência de aminoácidos e às estruturas terciárias destas proteínas.

A motivação para o desenvolvimento do sistema BDPF reside em três fatores:

- permitir a organização, manipulação e armazenamento dos dados relativos às proteínas fluorescentes estudadas no projeto Peixes Transgênicos Fluorescentes;

- o caráter inédito de um sistema web com este intuito, pois nenhum similar foi encontrado nas pesquisas realizadas;

- preprocessar, codificar e preparar os dados utilizados nas tarefas de KDD, através de *scripts* próprios, gerando automaticamente arquivos de entrada para as técnicas de mineração de dados.

6.1.1 Modelagem de Dados

Nesta seção é abordada a modelagem dos dados para posterior inclusão e estruturação no sistema gerenciador de banco de dados MySQL.

O primeiro passo para a modelagem dos dados é o levantamento de quais dados relativos às proteínas fluorescentes se deseja armazenar no SGBD. Para tanto, os pesquisadores parceiros no projeto Peixes Transgênicos Fluorescentes, especialistas do domínio da

aplicação, elencaram dados que eles julgam essenciais na manipulação e armazenamento do sistema BDPF, conforme segue:

- Classe (*class*): classificação geral para separar as classes de cores, baseada no comprimento de onda emitido pela proteína;
- Nome da Proteína (*name*): nome atribuído a proteína;
- Organismo (*organism*): organismo do qual a proteína foi originalmente extraída;
- Excitação (*excitation maximum*): comprimento de onda de excitação máxima (em nm);
- Emissão (*emission maximum*): comprimento de onda de emissão máxima (em nm);
- Rendimento quântico (*quantum yield*): número de vezes que um evento específico ocorre por fóton absorvido;
- Coeficiente de extinção (*maximal extinction coefficient*): refere-se à absorção de luz em um dado comprimento de onda por densidade de massa ou concentração molar;
- Brilho (*relative brightness*): refere-se ao brilho da fluorescência, que é determinado pelo produto do coeficiente de extinção pelo rendimento quântico;
- Fotoestabilidade ou Maturação (*photostabitity or maturation half-time*): é o tempo necessário para a proteína fluorescente produzir metade do seu máximo de fluorescência (em minutos ou até horas);
- pK_a: valor de pH no qual o brilho da proteína fluorescente é igual a 50% do brilho máximo medido no pH ótimo;
- Oligomerização (*oligomerization*): classificação de acordo com o número de cadeias polipeptídicas;
- Código PDB (*protein data bank code*): código de acesso para o registro PDB (*Protein Data Bank*), quando houver;
- Sequência de aminoácidos (*aminoacid sequence*): sequência de aminoácidos da proteína;
- Mutações (*mutation*): indicação de quais posições na sequência de aminoácidos sofreram modificações em relação a sequência da proteína selvagem, quando houver.

Os dados listados anteriormente são especialmente importantes sob o ponto de vista dos pesquisadores do projeto Peixes Transgênicos Fluorescentes, uma vez que o sistema BDPF tem por objetivo também servir como ferramenta de apoio ao projeto. Contudo,

além destes dados, também são importantes os dados relativos à estrutura terciária (tridimensional) das proteínas fluorescentes, pois são dados essenciais aos objetivos deste trabalho, de relacionar a estrutura terciária às classes de cores das proteínas fluorescentes.

Com este intuito, além de possuir o código de acesso ao registro do *Protein Data Bank*, também é necessário ser armazenada no sistema BDPF a estrutura terciária contida no arquivo oriundo do PDB, quando houver disponibilidade. O arquivo contendo estas informações apresenta-se na extensão *.pdb*.

O arquivo principal do PDB consiste apenas de estruturas tridimensionais determinadas experimentalmente, por isso nem toda proteína existente, o que também se aplica às proteínas fluorescentes, possui sua estrutura tridimensional depositada no PDB.

O conteúdo dos arquivos do *PDB* é constituído por centenas ou milhares de linhas rotuladas que indicam informações sobre o nome da molécula, sua fonte natural, sua preparação para o experimento, análises estatísticas em relação a qualidade do modelo 3D e fontes bibliográficas, dentre outras. Por fim, há uma longa lista de linhas, rotuladas com o termo *ATOM*, que indicam as coordenadas espaciais (x, y e z) de cada átomo, listadas por tipo de átomo, tipo de aminoácido e número de cadeia (BERMAN et al., 2000).

A partir da análise de todos estes dados e como se relacionam, foi elaborado o modelo relacional conforme esquema ilustrado na Figura 16. O modelo relacional consiste em um modelo composto por tabelas ou relações, onde cada tabela é um conjunto não-ordenado de linhas (tuplas) que, por sua vez, são uma série de campos (atributos) (ELMASRI; NAVATHE, 2003). Para uma melhor visualização das relações, objetivo maior deste diagrama, os atributos das tabelas foram suprimidos. A lista completa dos atributos das relações encontra-se no Anexo A.

É necessário ressalvar que, embora a análise dos dados oriundos dos arquivos do *PDB* indique a necessidade de atributos vinculados a informações do *PDB*, nem todos são obrigatórios. Assim, um arquivo do tipo *.pdb* obtido através de métodos de modelagem é passível de ser incluído no banco de dados representado pelo modelo relacional citado anteriormente.

Relação *protein*: representa a proteína em si, contendo os atributos relacionados a fluorescência da proteína e outros dados relativos ao arquivo *PDB*. No entorno desta tabela, na porção superior do esquema do modelo relacional, estão localizadas relações que representam classificações desta proteína. Relação *class*: representa a classe de cor da proteína fluorescente. Relação *organism*: representa o organismo de origem da proteína fluorescente. Relação *oligomerization*: representa o número de cadeias que compõe a molécula da proteína fluorescente.

Relação *chain*: representa as cadeias de aminoácidos que compõem as proteínas fluorescentes, por isso encontra-se associada à tabela *protein*. Relação *sequence*: representa a sequência de aminoácidos de cada cadeia das proteínas fluorescentes. Relação *mutation*:

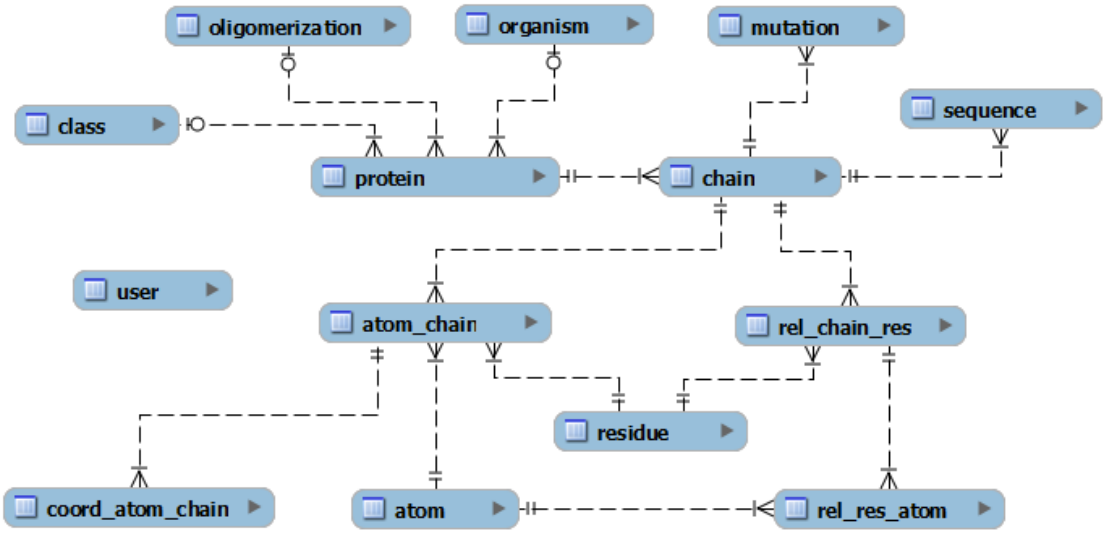


Figura 16: Esquema do modelo relacional que ilustra as relações existentes no banco de dados do sistema *Banco de Dados de Proteínas Fluorescentes* - do autor

representa as mutações nestas sequências de aminoácidos.

Relações *atom, residue, rel_chain_res, rel_res_atom, atom_chain, coord_atom_chain*: referem-se às informações sobre os aminoácidos e átomos, extraídas do arquivo PDB, contendo a relação de quais aminoácidos formam cada cadeia de uma proteína, a relação de quais átomos formam cada aminoácido e as coordenadas espaciais de tais átomos.

Relação *user*: refere-se ao cadastro dos usuários que podem acessar o sistema e, desta forma, não relaciona-se com nenhuma tabela existente no esquema do modelo relacional.

Com a modelagem de dados concluída, realiza-se a transposição do modelo relacional para o SGBD *Mysql*, criando-se efetivamente o banco de dados segundo o esquema ilustrado pela Figura 16. Dessa forma, com o sistema de informação desenvolvido e o sistema gerenciador de banco de dados apto a receber dados, o sistema BDPF encontra-se pronto para entrar em funcionamento, como tratado na próxima seção.

6.1.2 Funcionamento do Sistema BDPF

Como tratado anteriormente, o sistema BDPF tem como uma de suas tarefas o gerenciamento e organização de dados referentes à proteínas fluorescentes. O acesso ao sistema é realizado através de qualquer aplicativo do tipo navegador, porém somente pessoas autorizadas podem acessar as informações nele armazenadas, uma vez que o acesso é autenticado via login e senha.

A principal operação disponível em um sistema que se propõe a armazenar e organizar dados de qualquer natureza é a operação de adicionar dados. A inserção de informações no sistema BDPF, que realiza o preenchimento dos registros do banco de dados vinculados ao sistema, pode ser realizada de duas formas, dependendo da situação e de quais dados

estão disponíveis:

- *Preenchimento de formulário*: nesta situação, o usuário do sistema não possui o arquivo *PDB* da estrutura tridimensional da proteína a ser inserida (nem todas proteínas encontram-se disponíveis no PDB). Dessa forma, os campos referentes às informações sobre a fluorescência da proteína, sua classe, sequência de aminoácidos (se possuir), dentre outros dados, são inseridos via o preenchimento de formulário específico no sistema. As demais informações sobre a estrutura tridimensional da proteína são ignoradas neste preenchimento. Este caso, especificamente, atende à uma necessidade de uso do sistema BDPF como ferramenta de apoio ao projeto *Peixes Transgênicos Fluorescentes*.

- *Leitura do arquivo PDB*: neste outro caso, ao contrário do anterior, o usuário possui o arquivo *PDB* relativo à proteína a ser adicionada ao sistema. Assim, através de um formulário específico, o usuário indica o arquivo *PDB* e o sistema faz a leitura de informações contidas nele. O termo *leitura* refere-se a ação de busca e identificação de dados no arquivo, seguindo padrões previamente estabelecidos, o que possibilita extrair de forma automática os dados necessários referentes à estrutura tridimensional da proteína e realizar a inserção no BDPF. Após concluir esta inserção, a critério do usuário, pode ser realizada a inserção das informações relativas à fluorescência da proteína, caso as possua, uma vez que tais dados inexistem de forma padrão no arquivo *PDB*.

Consequentemente à operação de inserção, o sistema também dispõe de operações para alteração e visualização dos dados e remoção de um registro já existente. Existe ainda uma operação para visualização, no próprio sistema, da estrutura tridimensional de uma proteína armazenada, implementada através do *plugin* Jmol (HANSON, 2010), uma ferramenta para visualização de estruturas química em três dimensões.

Por fim, cabe destacar a importante função do sistema BDPF de gerar automaticamente o arquivo de entrada para execução das técnicas de KDD. Para as proteínas disponíveis no sistema que possuam seus dados estruturais devidamente armazenados no banco de dados, o sistema BDPF realiza a seleção, os cálculos e formata os dados seguindo o padrão específico para a compilação do arquivo que representa o conjunto de dados de treinamento nos algoritmos de mineração de dados. Este procedimento é explicado em mais detalhes na seção a seguir, na qual a seleção e formatação dos dados de treinamento têm um item específico.

6.2 Processo de KDD

A segunda etapa da metodologia consiste na utilização do processo de KDD, ou seja, realizar a seleção e preparação dos dados e a aplicação das três técnicas escolhidas refe-

rentes à tarefa de classificação. Cada uma das técnicas produz um modelo de classificação próprio e métricas de desempenho posteriormente avaliadas.

Esta seção aborda em mais detalhes, nos itens a seguir, a formação do conjunto de dados de treinamento através das etapas de seleção, pré-processamento e transformação nos dados. Inicialmente, porém, aborda-se a metodologia utilizada na definição das classes de cores das proteínas fluorescentes.

É necessário ressaltar, desde já, que o conjunto de dados de treinamento segue a mesma formatação e padrão para os três métodos de mineração de dados, sendo representados por atributos referentes às estruturas primária e terciária e por propriedades fluorescentes das proteínas fluorescentes.

6.2.1 Classes de Cores

Antes de iniciar a abordagem sobre os passos do processo de KDD aplicados aos dados do trabalho, é preciso explicar a escolha da classe de cor como atributo representante das propriedades fluorescentes. A opção pelo uso da classe de cor das proteínas como atributo rótulo de classe foi uma decisão baseada na análise dos dados disponíveis, pois uma vez que a proposta é relacionar a estrutura tridimensional com alguma propriedade fluorescente das proteínas, é evidente que obrigatoriamente somente registros com as informações da estrutura terciária podem ser utilizados.

Porém, estes registros, em sua totalidade adquiridos a partir de arquivos *PDB*, não possuem de maneira completa os dados de propriedades fluorescentes. A classe de cor, entretanto, é um atributo identificável analisando-se os campos de comentários do arquivo *PDB*, especialmente com a colaboração de especialistas do domínio, tornando possível completar todos os registros do BDPF com este atributo de forma correta.

A classe de cor de uma proteína fluorescente é obtida a partir do comprimento de onda do espectro de luz visível emitido pela proteína quando excitada por um determinado comprimento de onda, ambos calculados em nanômetros (nm), conforme os intervalos aproximados de emissão da Tabela 4 (OLENYCH et al., 2007). Esta divisão de classes de cores representa a aplicação da técnica de discretização, que é a representação de intervalos de valores numéricos por valores discretos (BATISTA, 2003).

Embora seja representada por um valor discreto, a classe de cor *Long Stokes* foge à regra de discretização imposta às demais classes, uma vez que sua característica não é emitir exclusivamente em determinada faixa de valores. Esta classe caracteriza-se por apresentar uma diferença entre o seu comprimento de onda de excitação e o seu comprimento de onda de emissão maior do que 100nm, diferente de todas as outras classes, onde este intervalo é em torno de 50nm (OLENYCH et al., 2007).

Classe de cor	Intervalo de comprimento de onda de emissão
Blue	440nm - 470nm
Cyan	470nm - 500nm
Green	500nm - 525nm
Yellow	525nm - 555nm
Orange	555nm - 580nm
Red	580nm - 630nm
Far-Red	630nm - 700nm
Long-Stokes	$>$ 570nm

Tabela 4: Representação da discretização do comprimento de onda, no espectro de luz visível, de proteínas fluorescentes em classes de cores - dados de (OLENYCH et al., 2007)

6.2.2 Seleção de Dados

O próprio desenvolvimento do BDPF já exemplifica a etapa de seleção de dados do processo de KDD. A unificação de bases de dados externas, com dados relativos às propriedades fluorescentes das proteínas oriundos da literatura e de documentos do especialista do domínio e dados relacionados às propriedades estruturais depositados no *Protein Data Bank* demonstram a tarefa de organizar os dados em um única estrutura.

Este agrupamento de informações seguiu a metodologia da junção orientada, uma vez que foram analisados e escolhidos atributos representativos das propriedades fluorescentes que tem o potencial de colaborar no processo de KDD, dentre dados com diversas propriedades existentes. As propriedades fluorescentes (já definidas na seção 6.1.1) escolhidas são: comprimento de onda de excitação, comprimento de onda de emissão, rendimento quântico, coeficiente de extinção, brilho, maturação ou fotoestabilidade e pK_a.

Outro fator que caracteriza a junção orientada nesta seleção de dados foi a escolha somente dos registros do PDB que representam proteínas fluorescentes e que têm a possibilidade de identificação de sua classe de cor, dentre todos os registros disponíveis. Esta seleção resultou uma lista de 109 arquivos *PDB* unificados ao BDPF, todos oriundos do *Protein Data Bank*. A listagem com o código de acesso a estes registros no PDB está disponível no Anexo B.

Assim, após esta seleção inicial de dados, encontra-se organizada a estrutura unificada de dados, representada pelo sistema BDPF.

6.2.3 Pré-processamento

Em sua totalidade, os registros armazenados no BDPF oriundos de arquivos do tipo *PDB*, essenciais para o objetivo do trabalho, não apresentam preenchidos os atributos referentes às propriedades fluorescentes das proteínas, visto que tais informações não encontram-se disponíveis no PDB. Esta situação caracteriza um dos problemas que são tratados nesta etapa do processo de KDD: valores ausentes nos dados.

O tratamento dos valores ausentes do atributo classe de cor, imprescindível ao objetivo

do trabalho, foi realizado através do preenchimento manual, com o auxílio dos especialistas do domínio. Os demais atributos vinculados às propriedades fluorescentes não eram passíveis de preenchimento, uma vez que nenhuma técnica pode ser aplicada. Com a ausência destes dados em outros registros, técnicas como o uso de medidas de média ou moda ou mesmo de técnicas de mineração de dados tornam-se inaplicáveis neste caso.

Dessa forma, com o tratamento dos valores ausentes do atributo classe de cor nos registros que possuem as informações estruturais das proteínas, esta etapa do processo de KDD é concluída.

6.2.4 Transformação

O próprio atributo que representa o rótulo de classe (classe de cor) já representa uma transformação na natureza dos dados, como já abordado anteriormente. Ele representa a discretização do comprimento de onda em classes de cores. Porém, a natureza original deste atributo, quando da inclusão no BDPF, já era discretizada.

A transformação foi uma importante etapa da preparação dos dados, visto que para lidar com a estrutura tridimensional da proteína e seus dados complexos, foi preciso implementar uma tarefa de criação de atributos a partir dos dados de coordenadas espaciais dos átomos, da relação de átomos e da relação de aminoácidos existentes nas proteínas. O método para a criação de atributos explicado a seguir e a posterior formatação do arquivo *CSV* de entrada contendo o conjunto de dados do processo de KDD foram implementados de forma automática através de um algoritmo, incorporado ao sistema BDPF como uma operação acionada por um botão na interface.

Neste ponto, a questão central do processo de KDD foi definir uma maneira de utilizar os dados referentes à estrutura terciária da proteína de forma padronizada para todos os registros disponíveis no banco de dados. Para elaborar uma possibilidade de uso desses dados, dois fatores foram levados em consideração: (1) sabe-se que mutações na sequência de aminoácidos de proteínas fluorescentes podem provocar mudanças de brilho, estabilidade e na classe de cor destas proteínas; (2) também é conhecida a influência que os aminoácidos espacialmente localizados nas estruturas no entorno do grupo cromóforo tem sobre este grupo de átomos e sobre a definição de cor das proteínas fluorescentes (CHUDAKOV et al., 2010).

Dessa forma, através da fundamentação teórica anterior e de consultas aos especialistas do domínio, elaborou-se uma estratégia para utilizar a relação de aminoácidos e a estrutura tridimensional das proteínas de forma padronizada no processo de KDD: a utilização de uma matriz de distância para o cálculo das distâncias entre todos os aminoácidos de uma proteína e os respectivos aminoácidos do seu grupo cromóforo.

Para cada um dos aminoácidos existentes em uma proteína, mediu-se a distância destes para os aminoácidos do grupo cromóforo, localizado na região central da molécula (estrutura 3D). Após, para cada um dos 20 tipos de aminoácidos formadores das proteínas

(conforme tabela presente no Anexo C), a menor distância calculada foi vinculada a cada um deles, por tipo. Ao fim, para cada registro do sistema BDPF criou-se 20 atributos que representam os 20 tipos de aminoácidos existentes, onde cada atributo foi preenchido com a menor distância calculada para aquele determinado tipo de aminoácido em relação ao cromóforo.

Por exemplo, em um dado registro, uma molécula de uma proteína fluorescente possui em sua sequência 3 aminoácidos Triptofano. Para cada um desses 3 aminoácidos foi calculada a distância entre eles e os aminoácidos do grupo cromóforo. Ao final, o atributo Triptofano deste registro apresenta como valor a menor dessas 3 distâncias calculadas.

A unidade de medida de comprimento utilizada para o cálculo das distâncias foi o *Angstrom*, originária do termo *Ångström*. A relação com a unidade de medida metro é tal que $1\text{Å} = 1^{-10}\text{m}$ (VERLI, 2014).

É importante abordar como foi efetuado o cálculo das distâncias entre cada aminoácido da sequência de uma proteína e os aminoácidos do grupo cromóforo. Este cálculo foi realizado considerando-se a distância euclidiana entre as posições espaciais de cada átomo do aminoácido em análise para cada átomo dos aminoácidos do cromóforo, conforme mostra a Equação 10. A menor distância encontrada entre os átomos é escolhida como a distância daquele aminoácido em relação ao cromóforo.

A escolha da menor distância entre os átomos é justificada pela maior influência dos aminoácidos próximos ao cromóforo na definição de cor das proteínas fluorescentes, especialmente devido às cadeias laterais destes aminoácidos (CHUDAKOV et al., 2010). Assim, ao utilizar a menor distância entre átomos, busca-se considerar que a mínima interação entre as os átomos das cadeias laterais dos aminoácidos próximos com os átomos do cromóforo já apresenta potencial de relacionar-se com a definição de cor e emissão de fluorescência nestas proteínas.

Para dois pontos tridimensionais $P(p_x, p_y, p_z)$ e $Q(q_x, q_y, q_z)$, a distância euclidiana d é calcula por:

$$d = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2} \tag{10}$$

A Tabela 5 mostra uma parcela destes atributos e registros para um melhor entendimento de sua estrutura. Os rótulos dos atributos utilizam a nomenclatura de uma letra dos aminoácidos a fim de simplificar a demonstração (a nomenclatura não abreviada pode ser consultada no Anexo C). É necessário destacar que em alguns casos no conjunto de dados, existem proteínas fluorescentes que não possuem algum dos 20 aminoácidos, o que impossibilitou o cálculo da sua respectiva distância para o cromóforo. É o caso demonstrado na linha *#2* da Tabela 5, onde o atributo K encontra-se em branco. Para seguir a lógica mencionada anteriormente para a criação dos atributos e como os algoritmos utilizados têm a capacidade de lidar com valores não disponíveis, decidiu-se por manter no conjunto de dados estes atributos em branco.

	A	C	D	E	F	G	H	I	K	L	M	..	Y	*classe*
#1	5,5	5,4	10,4	3,1	3,4	7,5	2,0	5,1	6,5	1,3	6,2	..	2,4	Green
#2	4,9	11,6	5,8	3,4	0	6,6	6,9	3,7		3,2	3,8	..	3,8	Red
#n	..	..	..	..	..	..	..	..	..	..	..	..	..	..

Tabela 5: Demonstração de uma parcela do conjunto de dados do processo de KDD após as etapas de seleção, pré-processamento e transformação dos dados - do autor

Em resumo, a estratégia da criação de atributos leva em consideração, na potencial definição da classe de cor de uma proteína fluorescente, tanto a influência dos aminoácidos, transformados em atributos, quanto a influência da estrutura espacial da molécula, com o cálculo das distâncias entre os aminoácidos e o cromóforo sendo o valor destes atributos.

Após as operações desta etapa, o conjunto de dados para a aplicação das técnicas de mineração de dados encontra-se corretamente formatado, conforme mostra a Tabela 6, contendo a lista de atributos final do conjunto de dados.

Atributo	Formato	Definição
A	Numérico	Distância* do aminoácido Alanina
C	Numérico	Distância* do aminoácido Cisteína
D	Numérico	Distância* do aminoácido Aspartato
E	Numérico	Distância* do aminoácido Glutamato
F	Numérico	Distância* do aminoácido Fenilalanina
G	Numérico	Distância* do aminoácido Glicina
H	Numérico	Distância* do aminoácido Histidina
I	Numérico	Distância* do aminoácido Isoleucina
K	Numérico	Distância* do aminoácido Lisina
L	Numérico	Distância* do aminoácido Leucina
M	Numérico	Distância* do aminoácido Metionina
N	Numérico	Distância* do aminoácido Asparagina
P	Numérico	Distância* do aminoácido Prolina
Q	Numérico	Distância* do aminoácido Glutamina
R	Numérico	Distância* do aminoácido Arginina
S	Numérico	Distância* do aminoácido Serina
T	Numérico	Distância* do aminoácido Treonina
V	Numérico	Distância* do aminoácido Valina
W	Numérico	Distância* do aminoácido Triptofano
Y	Numérico	Distância* do aminoácido Tirosina
Class	Categórico	Classe de cor da proteína (*Blue, Cyan, Green, Yellow, Orange, Red, Far-Red* e *Long-stokes*)

* menor distância do aminoácido em questão para o cromóforo da proteína

Tabela 6: Lista de atributos do conjunto de dados do processo de KDD - do autor

O conjunto de dados de treinamento para os algoritmos de mineração de dados apresenta ao todo 109 registros, após a execução das tarefas de seleção, pré-processamento e transformação. Os atributos desses registros estão preenchidos e formatados adequadamente para a execução das técnicas de mineração de dados.

É possível observar na Tabela 7, no entanto, que o número de instâncias que representam cada classe está desbalanceado, inclusive de forma bem acentuada entre as classes

Long-Stokes e *Green*, por exemplo. Mesmo sendo indicado, neste caso, o balanceamento entre as classes, optou-se pela não aplicação de nenhum método com este intuito. Uma vez que o possível efeito de *overfitting* para a classe majoritária influencia todos os classificadores, este fato não prejudica o objetivo de comparação entre os mesmos (TAN; STEINBACH; KUMAR, 2006).

Classe	Número de instâncias
Blue	4
Cyan	13
Green	27
Yellow	17
Orange	7
Red	17
Far-Red	21
Long-Stokes	3
TOTAL	109

Tabela 7: Quantidade de instâncias representantes de cada classe no conjunto de dados após as etapas de seleção, pré-processamento e transformação nos dados - do autor

Embora seja conhecida a importância das etapas que antecedem a mineração de dados e o quanto elas, muitas vezes, são mais decisivas na obtenção de melhores modelos de classificação do que a escolha ou parametrização dos algoritmos de classificação (HAN; KAMBER; PEI, 2011) (TAN; STEINBACH; KUMAR, 2006), o foco do presente trabalho não está em obter o melhor classificador possível para o problema. Este posicionamento claro relacionado aos objetivos justifica a opção por não aplicar mais métodos que objetivam manipular e adequar o conjunto de dados, como o balanceamento das classes e a normalização de atributos numéricos, métodos que frequentemente demonstram melhora no desempenho de algumas técnicas de classificação.

Essa fato justifica-se também pela intenção de os pesquisadores do projeto Peixes Transgênicos Fluorescentes possuírem autonomia futura em gerar o conjunto de dados a partir do BDPF e em construir e utilizar os classificadores na ferramenta WEKA de maneira simples e direta, somente executando os algoritmos sobre o arquivo que representa o conjunto de dados gerado pelo BDPF, mesmo sem um conhecimento mais amplo nesta área e no software WEKA.

Além disso, como a base de dados do BDPF é dinâmica e tende a crescer com o tempo, quando novas proteínas serão inseridas, qualquer método de balanceamento de classes aplicado ao conjunto atual poderia não necessariamente melhorar resultados com as novas instâncias inseridas neste conjunto futuramente.

No que se refere a normalização, em especial no modelo de classificação baseado em AD, a normalização dos dados numéricos descaracteriza os valores dos atributos representativos dos cálculos das distâncias, impedindo a direta interpretação do modelo de classificação.

Dessa forma, devido a todo o exposto, fez-se a opção por aplicar as técnicas de

mineração de dados diretamente sobre o conjunto de dados extraído do BDPF, que já possui métodos de pré-processamento e transformação dos dados aplicados. A partir deste conjunto de dados, prossegue-se para a etapa de execução das técnicas da mineração de dados.

6.2.5 Execução das Técnicas de Mineração de Dados

A execução das técnicas de mineração de dados deste trabalho utiliza algoritmos implementados no software WEKA, em sua versão 3.7.13. A construção dos classificadores, a partir destes algoritmos, é realizada através do ambiente de operação *Explorer*. Este ambiente de operação constrói o classificador, permitindo inclusive a visualização do mesmo, quando o modelo é representável graficamente. Além disso, possibilita também a utilização do classificador construído para a classificação de exemplos ainda não vistos (BOUCKAERT et al., 2015).

Para a técnica que utiliza AD, é utilizado o algoritmo *J48*, que por sua vez é a implementação do algoritmo *C4.5* na ferramenta WEKA. Já para a técnica baseada em RNA, utiliza-se a implementação de uma rede *Perceptron* multicamadas com retropropagação, chamada no software WEKA de *MultilayerPerceptron*. Por fim, para a técnica baseada em SVM, utiliza-se o algoritmo LibSVM, que implementa na ferramenta WEKA uma SVM de margens suaves que permite a utilização de múltiplas classes (BOUCKAERT et al., 2015).

Cada um destes algoritmos disponibiliza alguns parâmetros para a configuração de sua execução. Após vários testes empíricos realizados com diversas variações destes parâmetros e consultas à literatura, chegou-se aos valores que apresentam os melhores resultados. Na maioria dos casos, porém, como já abordado em (FRANK et al., 2004), os valores padrões dos parâmetros geram modelos de classificação com bons resultados. Embora não seja uma regra, pois está intimamente relacionado ao conjunto de dados de treinamento e a escolha do algoritmo, é algo que deve ser considerado (WITTEN; FRANK, 2005).

A avaliação de desempenho dos modelos de classificação é realizada através do método de validação cruzada em 10 partições, também disponível no software WEKA. A opção pelo particionamento em 10 se deve a testes empíricos, abordados na literatura tradicional, os quais demonstram ser o particionamento mais eficiente (WITTEN; FRANK, 2005).

6.3 Uso dos Classificadores

Durante o desenvolvimento do presente trabalho, o projeto Peixes Transgênicos Fluorescentes produziu uma variante inédita de uma proteína fluorescente. Através de processos de biologia molecular, com o uso de mutações na sequência de aminoácidos, foi

produzida esta variante. Devido ao sigilo por razões de registro de patente, não são fornecidas informações detalhadas em relação a esta variante e tampouco ela foi incluída na base de dados do BDPF, ou seja, ela não foi utilizada no conjunto de dados de treinamento dos algoritmos de classificação.

Devido também à inexistência da estrutura tridimensional da nova variante, uma vez que a produção dela foi realizada por meio de manipulações na sequência de aminoácidos e sua estrutura não foi experimentalmente determinada, ela não é candidata ao conjunto de dados de treinamento das técnicas de mineração de dados.

A nova proteína fluorescente é oriunda de três mutações realizadas sobre a sequência de aminoácidos de uma proteína pertencente à classe de cor *Red* que já possui estrutura tridimensional experimentalmente resolvida e encontra-se depositada no PDB. É preciso destacar que esta nova variante emite fluorescência em duas faixas distintas, dependendo do comprimento de onda no qual é excitada. Isso a faz pertencer, de forma excepcional, à duas classes de cor distintas simultaneamente: *Red* e *Green*.

Como a nova proteína é inédita no universo das proteínas fluorescentes e também é um exemplo não visto para os classificadores construídos, modelou-se sua estrutura tridimensional com o auxílio de softwares especializados. Através dos modelos das estruturas tridimensionais que representam a nova proteína é possível testar a capacidade dos classificadores em predizer corretamente a classe de cor desta nova variante, o que constitui a terceira etapa da metodologia.

Dessa forma, partindo-se de uma estrutura já existente e da sequência de aminoácidos da nova proteína, métodos de modelagem por homologia, de *threading* e de mutação de aminoácidos foram executados em 3 ferramentas distintas para a construção de modelos da estrutura tridimensional da nova variante.

Ao total, 4 estruturas da nova proteína no formato de um arquivo do tipo *PDB* foram modeladas e inseridas ao BDPF (conforme Tabela 8) para, a partir dele, gerar um conjunto de dados de teste, no mesmo formato do conjunto de treinamento dos classificadores.

Registro	*Estrutura*
Nº 1	Representa a estrutura modelada na ferramenta Modeller (WEBB; SALI, 2014)
Nº 2	Representa a estrutura modelada na ferramenta i-Tasser (YANG et al., 2015)
Nº 3	Representa a estrutura modelada na ferramenta Phyre2 (KELLEY et al., 2015)
Nº 4	Representa a estrutura modelada no algoritmo de mutação da ferramenta Modeller (WEBB; SALI, 2014)

Tabela 8: Identificação do conjunto de teste, contendo 4 instâncias que representam as estruturas da proteína inédita realizadas pelas ferramentas de modelagem - do autor

Ao concluir a execução das etapas de pré-processamento, transformação de dados e criação de atributos já citadas na seção anterior, um conjunto de dados de teste contendo estas 4 estruturas da proteína inédita foi gerado. Logo após, este conjunto de teste foi submetido aos classificadores construídos no ambiente *Explorer* para classificação.

Dada a excepcionalidade da nova proteína pertencer a duas classes de cor ao mesmo

tempo, é considerada correta a classificação dessa proteína caso seja indicada pelos classificadores tanto a classe *Green* quanto a *Red*.

6.4 Comparação de Performance

A comparação de performance dos classificadores, quarta e última etapa da metodologia, é realizada sob dois pontos de vista. Um deles é através de uma avaliação qualitativa em relação aos classificadores em si: o modelo de classificação gerado pelo classificador e sua aplicabilidade no conjunto de dados, sua interpretabilidade e a capacidade de extração de conhecimento e características dos dados, dentre outras. Tecnicamente, este primeiro ponto de vista não representa uma comparação propriamente dita, mas permite a afirmação de que, se determinado classificador possuir mais qualidades desejáveis ao seu âmbito de aplicação, ele apresenta vantagem sobre os demais (TAN; STEINBACH; KUMAR, 2006).

O outro ponto de vista é estritamente quantitativo, baseado nas métricas de desempenho geradas pelos métodos de avaliação e também pela acurácia em relação às classificações do conjunto de teste. Pode-se comparar de forma puramente numérica os valores das métricas de desempenho, analisando-se qual deles é superior ou inferior, porém, a maneira mais correta de comparação numérica formal é através de testes estatísticos realizados sobre os valores das métricas (TAN; STEINBACH; KUMAR, 2006).

O uso dos classificadores para a classificação de uma proteína fluorescente inédita desenvolvida no projeto Peixes Transgênicos Fluorescentes, utilizada como conjunto de teste, inclusive, simula uma futura aplicabilidade deste tipo de ferramenta no apoio à pesquisa: através da sequência de aminoácidos de uma nova variante de proteína fluorescente desenvolvida no projeto de pesquisa, é possível modelar sua estrutura tridimensional e submeter este conjunto de dados ao classificador para que ele realize a predição da classe de cor desta nova proteína. O que possibilita diminuir a necessidade da medição em laboratório do comprimento de onda emitido por esta nova variante, se a classe de cor predita não for do interesse dos pesquisadores.

Em relação às métricas de desempenho dos classificadores, sua avaliação e comparação é realizada através de um experimento no ambiente de operação *Experimenter* do software WEKA, onde o conjunto de dados de treinamento é submetido por 1000 vezes aos 3 algoritmos de mineração de dados com o método de avaliação de validação cruzada em 10 partições.

Na conclusão do experimento, cada par de classificadores (AD x RNA, RNA x SVM e SVM x AD) é testado e comparado estatisticamente através do *teste-t* pareado corrigido para os valores médios das seguintes métricas de desempenho geradas nas 1000 iterações dos 3 algoritmos: acurácia, precisão, revocação e *F-measure*. O *teste-t* indica se a hipótese da não-diferença entre as médias das métricas de desempenho de cada par de classificador

deve ser rejeitada ou não.

Como a métrica de desempenho índice *Kappa* possui uma escala de significado própria para a interpretação de seu valor (Tabela 3), ela não é incluída no teste estatístico, pois a comparação entre valores desta métrica não produz nenhum resultado objetivo. O valor médio calculado para o índice *Kappa* é interpretado e comparado diretamente segundo a sua escala de valores.

Encerrada a explicação sobre as 4 etapas que compõem a metodologia proposta para o trabalho, no próximo capítulo aborda-se as ferramentas computacionais que possibilitaram a execução desta metodologia.

7 FERRAMENTAS

Neste capítulo, são abordadas as ferramentas e tecnologias utilizadas para a modelagem de dados, desenvolvimento e funcionamento do sistema BDPF, execução das técnicas de mineração de dados do processo de KDD e da modelagem da estrutura tridimensional da proteína fluorescente inédita, já citadas na metodologia do presente trabalho.

7.1 Protein Data Bank

O *Protein Data Bank* (PDB) é um banco de dados de estruturas tridimensionais experimentais de macromoléculas biológicas, como as proteínas. Ele faz parte do *The Worldwide Protein Data Bank*, um consórcio de organizações para o depósito, processamento e distribuição dos dados do PDB. Estas organizações têm por missão manter um único arquivo *PDB* de dados estruturais de macromoléculas de forma pública e gratuita para a comunidade mundial (BERMAN et al., 2000).

O arquivo principal do PDB consiste apenas de estruturas tridimensionais determinadas experimentalmente por cristalografia por difração de raios-X, Ressonância Magnética Nuclear, Microscopia Eletrônica ou combinações dessas técnicas experimentais (BERMAN et al., 2000).

7.2 Desenvolvimento do Sistema Web

Nesta seção, algumas definições e conceitos são apresentados com o intuito de abordar as tecnologias e ferramentas utilizadas no desenvolvimento do sistema BDPF. O PHP é uma linguagem de *script*, de código aberto, muito utilizada, e especialmente adequada para o desenvolvimento web, pois pode ser embutida dentro de uma interface HTML (linguagem de marcação utilizada para produzir páginas na web) (DALL'OGLIO, 2009). Estas características, somadas a experiência do autor no desenvolvimento web na linguagem PHP, tornam esta linguagem a escolha natural para o desenvolvimento do sistema.

Como o sistema BDPF tem por um de seus objetivos armazenar dados, em conjunto com a linguagem PHP, é necessária a utilização de um Sistema Gerenciador de Banco de Dados. O MySQL é um SGBD relacional de código aberto largamente utilizado em

aplicações web para gerir bases de dados. Ele utiliza a linguagem SQL (linguagem de consulta estruturada) como interface para inserir, acessar e gerenciar os dados armazenados nos bancos de dados (MILANI, 2010). Da mesma forma que a linguagem PHP, o gerenciador de banco de dados MySQL também foi uma escolha natural pelo seu correto funcionamento em conjunto com a linguagem PHP.

O sistema BDPF foi implementado sobre o *CodeIgniter*, um *framework* de código aberto para desenvolvimento de aplicações e sistemas com linguagem PHP. O *CodeIgniter* é um *toolkit* (conjunto de ferramentas) cujo objetivo é permitir o rápido desenvolvimento de aplicações, quando comparado a utilização de nenhuma ferramenta (GABARDO, 2012).

Uma importante característica do *framework*, e parte da agilidade no desenvolvimento das aplicações decorre disso, é a utilização da abordagem conhecida como *Modelo-Visão-Controle* (MVC), a qual permite uma forte separação entre a lógica e a apresentação, em camadas, da seguinte maneira (GABARDO, 2012):

- Modelo: cria a comunicação da aplicação com o sistema de banco de dados, possibilitando a realização das operações de criação de tabelas na base de dados, além de ações de leitura, atualização e remoção de registros nas tabelas.
- Visão: constitui todas as informações apresentadas ao usuário na interface, neste caso, uma página web HTML.
- Controle: é o intermediário entre a camada Modelo e a camada Visão, além de realizar o processamento de requisições para a geração de páginas.

A maior velocidade no desenvolvimento dos sistemas através do *CodeIgniter* é também possibilitada devido a um conjunto de bibliotecas para tarefas comuns que o *framework* possui, diminuindo assim a quantidade de linhas de código necessárias para a realização das ações desejadas. Na prática, as bibliotecas são formadas por conjuntos de classes que já possuem estruturas de atributos e métodos que facilitam a maioria das tarefas comuns como a conexão com o sistema de banco de dados, tratamento e consulta de dados, criação da interface visual da aplicação, dentre outras (GABARDO, 2012).

7.3 Software WEKA

O software WEKA é uma coleção de algoritmos de aprendizagem de máquina e ferramentas de pré-processamento de dados desenvolvida pela *University of Waikato*, Nova Zelândia. O nome da ferramenta é uma sigla para *Waikato Environment for Knowledge Analysis*. O sistema da ferramenta foi implementado em *Java*, é multiplataforma e é distribuído sob os termos da licença GNU - *General Public License* (BOUCKAERT et al., 2015).

O programa fornece implementações de algoritmos de aprendizagem, com uma interface uniforme para eles, juntamente com métodos para pré e pós-processamento de dados e para avaliação de resultados. Inclui métodos para classificação, regressão, clusterização, regras de associação e seleção de atributos. Para todos os algoritmos, o arquivo de entrada é representado na forma de uma tabela relacional incluída em um arquivo no formato *ARFF*, porém também suporta a abertura direta de arquivos no formato CSV, dentre outros (BOUCKAERT et al., 2015).

Existem quatro possibilidades de interface que podem ser utilizadas, sendo que estas podem ser executadas diretamente via código *Java*, são elas: *Simple Cient*, onde a interação com o usuário ocorre por linhas de comando, por isso, exige conhecimento profundo do programa; *Explorer*, esta é a interface mais comum, que disponibiliza separadamente as etapas de pré-processamento, mineração de dados e pós-processamento; *Experimenter*, ambiente no qual pode ser avaliado o desempenho dos algoritmos de aprendizagem através de avaliações estatísticas; *KnowledgeFlow*, ferramenta gráfica que permite criação de um fluxo de processos de KDD (FRANK et al., 2004).

Mais informações e características da ferramenta WEKA podem ser encontradas em (BOUCKAERT et al., 2015).

A seguir, são abordados cada um dos três algoritmos e os principais parâmetros utilizados em sua execução.

7.3.1 Algoritmo J48

O algoritmo *J48* é a implementação do software WEKA para o algoritmo de indução de árvores de decisão C4.5 (QUINLAN, 1993). Nesta implementação, o *J48* tem a capacidade de lidar com classes nominais (categóricas), classes binárias e, até, com valores faltantes nas classes. Em relação aos atributos, é capaz de lidar com atributos nominais (categóricos), numéricos, unários, binários e datas, além de tratar atributos categóricos vazios e valores faltantes (BOUCKAERT et al., 2015).

Os principais parâmetros de execução do algoritmo *J48*, disponibilizados pelo software WEKA, estão definidos na Tabela 9.

7.3.2 Algoritmo MultilayerPerceptron

O algoritmo *MultilayerPerceptron* da ferramenta WEKA é a implementação de uma RNA do tipo *Perceptron* com possibilidade de múltiplas camadas ocultas ou intermediárias. Este algoritmo é capaz de lidar com classes nominais, binárias, numéricas, datas e com valores faltantes. Além disso, tem a capacidade de aceitar atributos nominais, de datas, binários, unários, numéricos e tratar valores faltantes e atributos nominais em branco (BOUCKAERT et al., 2015).

Os principais parâmetros de execução do algoritmo *MultilayerPerceptron*, disponibilizados pelo software WEKA, estão definidos na Tabela 10.

Parâmetro	*Descrição*
unpruned	Se a poda não é realizada
confidenceFactor	Nível de confiança usado para a poda (valores menores incorrem em mais poda)
reducedErrorPruning	Se o método de Poda de Redução de Erro é usada ao invés do método de poda original do algoritmo C4.5
subtreeRaising	Se considera a operação de elevação da subárvore na poda
binarySplits	Se usa divisão binária em atributos nominais durante a construção das árvores
minNumObj	Número mínimo de instâncias por folha da árvore
useMDLcorrection	Se a correção MDL (Minimum Description Length) é usada quando encontrar divisão de atributos numéricos
collapseTree	Se partes que não reduzem o erro de treinamento são removidas

Tabela 9: Parâmetros disponíveis na execução do algoritmo *J48* no software WEKA - dados de (BOUCKAERT et al., 2015)

Parâmetro	*Descrição*
seed	Usado para inicializar o gerador de número aleatório (utilizado para preencher o peso inicial das conexões e também para embaralhar os dados de treinamento)
momentum	Valor aplicado aos pesos durante a atualização dos mesmos no algoritmo de retropropagação
normalizeAttributes	Indica se realiza a normalização de atributos
hiddenLayers	Define as camadas ocultas da rede neural, através de uma lista de números inteiros, um para cada camada
decay	Causa o decréscimo na taxa de aprendizagem, dividindo a taxa inicial de aprendizagem pelo número de período, para determinar a taxa de aprendizagem corrente
validationSetSize	A porcentagem do tamanho do conjunto de dados de validação. Se for 0, nenhum conjunto de validação é usado e ao invés disso a rede irá treinar por um número específico de ciclos
trainingTime	Número de ciclos de treinamento
autoBuild	Adiciona e conecta as camadas ocultas na rede
learningRate	Taxa de aprendizagem, ou seja, o quanto os pesos são atualizados no algoritmo de retropropagação
reset	Permite reiniciar com uma taxa de aprendizagem menor. Se a rede diverge da resposta, ela será automaticamente reiniciada com uma taxa de aprendizagem menor e começar o treinamento novamente

Tabela 10: Parâmetros disponíveis na execução do algoritmo *MultilayerPerceptron* no software WEKA - dados de (BOUCKAERT et al., 2015)

7.3.3 Algoritmo LibSVM

O algoritmo LibSVM é oriundo de um pacote que abriga diversas ferramentas referentes à implementações de classificadores baseados em SVM. Este algoritmo lida com classes nominais, binárias e contendo valores faltantes. Quantos aos atributos, aceita nominais, binários, unários, numéricos e trata valores faltantes (BOUCKAERT et al., 2015).

É importante destacar que originalmente a técnica de SVM é aplicável diretamente somente em problemas binários, ou seja, de duas classes. Neste trabalho, como já observado, existem 8 classes distintas, o que não impede o uso do algoritmo *LibSVM*, já que há o tratamento para esta limitação inicial, através da redução do problema multiclasses

em diversos problemas binários. A estratégia um-contra-um é aplicada, e o algoritmo constrói um classificador para cada par de classes, então para C classes são construídos $C(C-1)/2$ classificadores. A classificação é feita por uma estratégia de votação, onde cada classificador atribui uma das duas classes para uma instância, assim, a classe mais votada determina a classificação dessa instância (CHANG; LIN, 2011) (SCHÖLKOPF; SMOLA, 2002).

Os principais parâmetros de execução do algoritmo *LibSVM*, disponibilizados pelo software WEKA, estão definidos na Tabela 11.

Parâmetro	*Descrição*
seed	O número base aleatório a ser utilizado
SVMType	O tipo de SVM utilizada
kernelType	Tipo de função *kernel* utilizada
gamma	Parâmetro gamma do *kernel* FBR para uso, se 0 utiliza 1/nº instâncias
shrinking	Se usa a heurística de diminuição
eps	A tolerância de critério de parada
cost	O parâmetro custo C de violação de restrição para C-SVC
weights	O peso para usar nas classes - se em branco, todas as classes utilizam o mesmo peso
normalize	Se normaliza os dados
probabilityEstimates	Se gera estimativas de probabilidade em vez de -1 / + 1 para problemas de classificação

Tabela 11: Parâmetros disponíveis na execução do algoritmo *LibSVM* no software WEKA - dados de (BOUCKAERT et al., 2015)

7.3.4 Ambiente Experimenter - Teste Estatístico

O ambiente de operação *Experimenter*, do software WEKA, possibilita execuções consecutivas de algoritmos de mineração de dados para um ou mais conjuntos de dados. Após a execução dos algoritmos e avaliação do classificadores segundo o método de avaliação escolhido, o teste estatístico do tipo *teste-t* pareado é efetuado sobre as médias das métricas de desempenho geradas (BOUCKAERT et al., 2015).

É importante ressaltar que o *teste-t* pareado, conforme abordado no capítulo 5, assume que as amostras dos dois grupos de médias de valores não apresentam reamostragem, o que não se aplica nos casos das métricas de desempenho dos classificadores, pois os conjuntos de treinamento e teste se sobrepõem numa proporção de 10% devido ao uso do método de avaliação de validação cruzada em 10 partições (VIEIRA, 2015) (FRANK et al., 2004).

Contudo, o ambiente *Experimenter* do software WEKA disponibiliza o *teste-t* pareado corrigido, o qual ajusta o *teste-t* pareado para estes casos, através da inclusão da fração de dados reamostrados no cálculo do *valor-t* (BOUCKAERT et al., 2015).

Assim, a avaliação e comparação das métricas de desempenho dos classificadores é realizada através do ambiente de operação *Experimenter*. O conjunto de dados de treinamento foi submetido aos 3 algoritmos de mineração de dados, através de um experimento

configurado no software neste ambiente de operação.

Ao final, os valores das médias das métricas de desempenho são testados, através do *teste-t* pareado corrigido, para indicar se há ou não diferença estatisticamente significativa entre tais médias.

7.4 Modelagem de Estruturas Tridimensionais

A modelagem da estrutura tridimensional da proteína fluorescente inédita foi realizada através de softwares especializados, a fim de se obter modelos de estruturas tridimensionais que possibilitem a utilização destes dados como conjunto de teste dos classificadores.

No software *Modeller* (WEBB; SALI, 2014) foram construídas duas estruturas para a nova proteína, baseando-se em dois processos distintos disponibilizados. No primeiro processo, de modelagem por homologia, tendo como base a estrutura molde na qual a nova proteína é baseada, os algoritmos do software executam o alinhamento entre as sequências de aminoácidos (da proteína molde e da que se deseja modelar) e, a partir da identificação de trechos homólogos, realiza a construção do modelo da nova proteína a partir das estruturas identificadas na proteína molde.

Já o segundo processo é baseado em uma operação de modelagem de mutação (WEBB; SALI, 2014), também a partir da estrutura tridimensional na qual a nova variante foi criada. Uma vez que a nova proteína é resultado de 3 mutações, este processo de mutação de um único aminoácido foi executado sucessivamente por 3 vezes, com a estrutura resultante da execução anterior sendo a base para a posterior, até que a estrutura apresente as 3 mutações necessárias para representar a nova proteína. Este processo ocorre pela otimização no posicionamento espacial da proteína devido à mutação indicada a cada execução. Mesmo este sendo um processo pouco ortodoxo neste objetivo de modelar uma estrutura que represente a nova proteína, é mais uma possibilidade de modelagem para aplicar aos classificadores.

Como resultado destes dois processos, o software *Modeller* gerou duas estruturas tridimensionais, uma referente a cada processo, que representam a estrutura terciária da nova variante de proteína fluorescente criada.

Além destes, utilizou-se duas ferramentas online de modelagem de estruturas. A ferramenta de modelagem por homologia *Phyre2* (KELLEY et al., 2015), a qual, através da sequência de aminoácidos da nova proteína, executa algoritmos que buscam estruturas homólogas em diversas estruturas de proteínas experimentalmente já resolvidas e utiliza estas partes como moldes para construir o modelo completo da estrutura alvo desejada. Também utilizou-se a ferramenta *i-Tasser* (YANG et al., 2015) que realiza a modelagem da estrutura alvo através do método *threading*, ou seja, um método de predição *ab-initio* que utiliza informação estrutural em seu processo. Estas ferramentas resultaram em mais dois modelos de estruturas tridimensionais referentes à nova proteína.

8 RESULTADOS

De acordo com a metodologia e as ferramentas utilizadas, abordadas nos capítulos anteriores, os classificadores foram construídos, avaliados e geraram as métricas de desempenho a partir do método de validação cruzada em 10 partições. Os resultados obtidos na construção dos modelos de classificação, no cálculo das métricas de desempenho para avaliação dos 3 classificadores, no teste estatístico para comparação das métricas e no uso dos classificadores para predição de novos exemplos são mostrados ao longo do presente capítulo.

8.1 Construção dos Modelos de Classificação

O modelo de classificação de Árvore de Decisão foi construído pelo algoritmo J48, através do ambiente *Explorer*, segundo os parâmetros especificados na Tabela 12.

Parâmetro	*Valor*
unpruned	Falso
confidenceFactor	0.25
reducedErrorPruning	Falso
subtreeRaising	Verdadeiro
binarySplits	Falso
minNumObj	2
useMDLcorrection	Verdadeiro
collapseTree	Verdadeiro

Tabela 12: Parâmetros utilizados na execução do algoritmo *J48* no software WEKA - do autor

É possível notar a partir do parâmetro *useMDLcorrection* (= Verdadeiro) que o algoritmo utiliza o princípio da Descrição de Comprimento Mínimo (do inglês *Minimum Description Lenght* - MDL) quando se depara com a necessidade de divisão de atributos numéricos (QUINLAN, 1996).

Dessa forma, para encontrar o melhor limiar que divide binariamente um atributo numérico, é utilizada uma função de discretização que considera a medida de entropia para buscar o limiar de divisão. Isso é feito recursivamente até ser satisfeito um critério de parada. O critério de parada baseia-se no princípio MDL e compara se o ganho de

informação obtido com a criação de um novo ponto de corte é melhor que o anterior (FAYYAD; IRANI, 1993) (TAN; STEINBACH; KUMAR, 2006).

Outro ponto a ser abordado é referente a poda no algoritmo J48. A partir do parâmetro *reducedErrorPruning* (= Falso), foi feita a opção pelo método original de poda do algoritmo C4.5, que realiza a pós-poda (após a árvore inicial ser construída) com base numa estimativa do erro no conjunto de treinamento (poda baseada no erro), segundo o nível de confiança estabelecido pelo parâmetro *confidenceFactor* (= 0.25). Esta estratégia de poda assume uma distribuição binomial para os exemplos de um nó. Assim, se a estimativa do erro no nó é menor ou igual à soma das estimativas de erros dos nós descendentes o nó é transformado em folha, ocorrendo a poda (QUINLAN, 1993) (HAN; KAMBER; PEI, 2011).

A representação gráfica do modelo criado está ilustrada na Figura 17, extraída do software WEKA, na qual é possível notar que o modelo de classificação possui 27 nós, sendo 14 deles nós-folhas.

De acordo com o modelo, dentre todos os atributos do conjunto de dados de treinamento, aqueles destacados como os mais importantes para a definição da classe de cor de uma proteína fluorescente são: *W, T, V, E, I, K, Y, F* e *H*. Estes são os atributos representados pelo nós de decisão (formato circular) na árvore.

As classes de cores são os nós-folhas ilustrados em retângulos contendo o nome da classe de cor que ele representa e um par de valores numéricos. O primeiro valor numérico deste par representa o número de instâncias preditas como pertencentes à classe que o nó-folha representa, segundo o método de avaliação, já o segundo valor representa o número destas instâncias classificadas incorretamente. Quando todas as instâncias de determinado nó-folha são corretamente preditas, o segundo valor é suprimido na representação do nó-folha (WITTEN; FRANK, 2005). A existência de valores decimais em alguns pares destes valores deve-se à divisão do conjunto de dados pelo método da validação cruzada.

Interligando todos estes nós de decisão e nós-folhas estão os ramos que representam a discretização binária dos valores das distâncias do conjunto de dados, de acordo com um determinado limiar calculado para cada nó de decisão.

Com o intuito de demonstrar a extração das regras *se-então* a partir da árvore de decisão, duas regras são extraídas. Partindo-se do nó raiz *W*, a regra 1 é extraída percorrendo-se sempre os ramos à esquerda dos nós até encontrar um nó-folha, e a regra 2 é extraída percorrendo-se sempre os ramos à direita dos nós até encontrar um nó-folha. As regras extraídas são:

- *Regra 1*: SE o valor de $W <= 6.8$ e o valor de $T <= 4.9$ e o valor de $E <= 4.5$ e o valor de $F <= 1.4$, ENTÃO o registro é da classe *Red*.
- *Regra 2*: SE o valor de $W > 6.8$ e o valor de $V > 1.5$ e o valor de $Y > 3.8$, ENTÃO o registro é da classe *Green*.

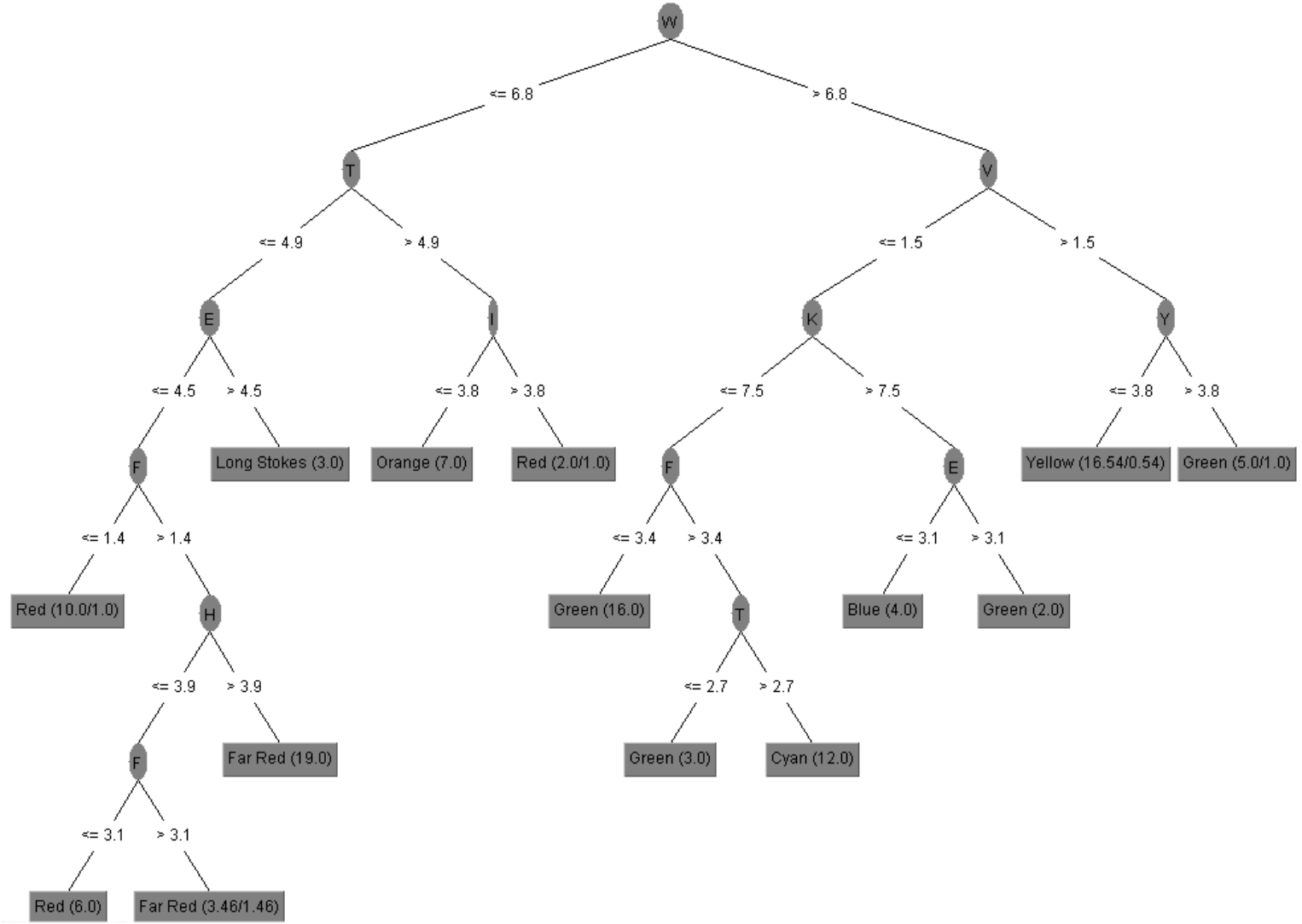


Figura 17: Representação gráfica do modelo de classificação de Árvore de Decisão construído pelo algoritmo J48 no software WEKA - do autor

O algoritmo *MultilayerPerceptron* construiu um modelo de rede neural artificial através do ambiente *Explorer*, conforme os parâmetros elencados na Tabela 13.

Parâmetro	*Valor*
seed	0
momentum	0.2
normalizeAttributes	Falso
hiddenLayers	a,a
decay	Falso
validationSetSize	0
trainingTime	500
autoBuild	Verdadeiro
learningRate	0.3
reset	Verdadeiro

Tabela 13: Parâmetros utilizados na execução do algoritmo *MultilayerPerceptron* no software WEKA - do autor

O parâmetro *hiddenLayers* demonstra a utilização de duas camadas ocultas, contendo em cada uma delas o número de neurônios a, que é a média aritmética entre o tamanho da entrada (20 atributos) e o tamanho da saída (8 classes), ou seja, 14 neurônios. A quantidade de camadas e o número de neurônios contidos nelas são definições empíricas, estabelecidas após inúmeros testes empíricos variando-se seus valores, porém a literatura estabelece que para a resolução de problemas de classificação uma rede neural com uma ou duas camadas ocultas é normalmente suficiente (MIIKKULAINEN, 2010) (FIESLER, 1996). O aumento desmedido das camadas ocultas nem sempre é vantajoso, visto que a retropropagação da estimação do erro utilizada para a atualização dos pesos das conexões perde sua precisão quando é realizada dentre muitas camadas, pois a cada camada ela torna-se a estimativa da estimativa e assim por diante (HAYKIN, 2001).

O parâmetro *learningRate* tem grande influência durante o processo de treinamento da RNA. Uma taxa de aprendizado muito baixa torna o aprendizado da rede muito lento, ao passo que uma taxa de aprendizado muito alta provoca oscilações no treinamento e impede a convergência do processo de aprendizado (HAYKIN, 2001). Geralmente seu valor varia de 0.1 a 1.0. Com o parâmetro *decay* (= Falso), a taxa de aprendizado é fixa, fato que explica a utilização de um valor não muito alto (= 0.3) (MIIKKULAINEN, 2010). Já o parâmetro *normalizeAttributes* (= Falso) indica a opção pela não normalização dos atributos, fato devidamente justificado na metodologia do presente trabalho.

A inclusão do termo *momentum* tem por objetivo aumentar a velocidade de treinamento da RNA e reduzir o perigo de instabilidade. Este termo pode ou não ser utilizado durante o treinamento e seu valor varia de 0.0 (não utilização) a 1.0. Um valor empiricamente recomendado para o parâmetro é 0.2. Também é importante destacar o critério de parada para o aprendizado da rede, neste caso o número de ciclos de treinamento, conforme define o parâmetro *trainingTime* (500 ciclos), ou seja, o número de vezes em que o conjunto de treinamento é apresentado à rede. Um número excessivo de ciclos pode levar

a rede à perda do poder de generalização (*overfitting*). Por outro lado, com um pequeno número de ciclos a rede pode não chegar ao seu melhor desempenho (*underfitting*). Empiricamente demonstrou-se que valores entre 500 e 3000 ciclos de treinamento são boas sugestões, dependendo da complexidade do problema (HAYKIN, 2001) (MIIKKULAINEN, 2010).

A RNA ilustrada na Figura 18, extraída do software WEKA, contém 20 neurônios na camada de entrada, dispostos verticalmente, em verde, representados pelos rótulos dos atributos do conjunto de dados: *A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y*. Ao centro, também dispostos verticalmente, em vermelho, encontram-se os 28 neurônios formadores das duas camadas ocultas, 14 em cada uma das camadas. Por fim, dispostos em tom amarelo, mais a direita, encontram-se os 8 neurônios que compõem a camada de saída, cada um conectado diretamente a uma classe, representadas por retângulos contendo o rótulo da classe.

A nomenclatura dos neurônios dispostos nas camadas do modelo de RNA construído pelo software WEKA é realizada automaticamente pelo software. Para o correto entendimento das conexões entre os neurônios das diferentes camadas, tendo por base o modelo da Figura 18, a nomenclatura dos neurônios nas diferentes camadas segue a seguinte forma:

- Camada de entrada: seus neurônios encontram-se diretamente conectados aos neurônios da camada oculta 1. A nomenclatura segue os rótulos dos atributos do conjunto de dados, de cima para baixo, do *A* ao *Y*;

- Camada Intermediária ou Oculta 1: seus neurônios encontram-se diretamente conectados aos neurônios da camada de entrada e da camada oculta 2. A nomenclatura segue uma numeração consecutiva, de cima para baixo, de 22 até 35, ou seja, o neurônio 22 é o mais ao topo, o neurônio 23 é o que está abaixo dele, e assim sucessivamente até o neurônio 35, localizado mais abaixo;

- Camada Intermediária ou Oculta 2: seus neurônios encontram-se diretamente conectados aos neurônios da camada oculta 1 e da camada de saída. A nomenclatura segue uma numeração consecutiva, de cima para baixo, de 8 até 21, ou seja, o neurônio 8 é o mais ao topo, o neurônio 9 é o que está abaixo dele, e assim sucessivamente até o neurônio 21, localizado mais abaixo;

- Camada de Saída: seus neurônios encontram-se diretamente conectados aos neurônios da camada oculta 2. A nomenclatura também segue uma numeração consecutiva, de cima para baixo, de 0 até 7, ou seja, o neurônio 0 é o mais ao topo (conectado à saída *Blue*), o neurônio 1 é o que está abaixo dele (conectado à saída *Cyan*), e assim sucessivamente até o neurônio 7, localizado mais abaixo (conectado à saída *Yellow*).

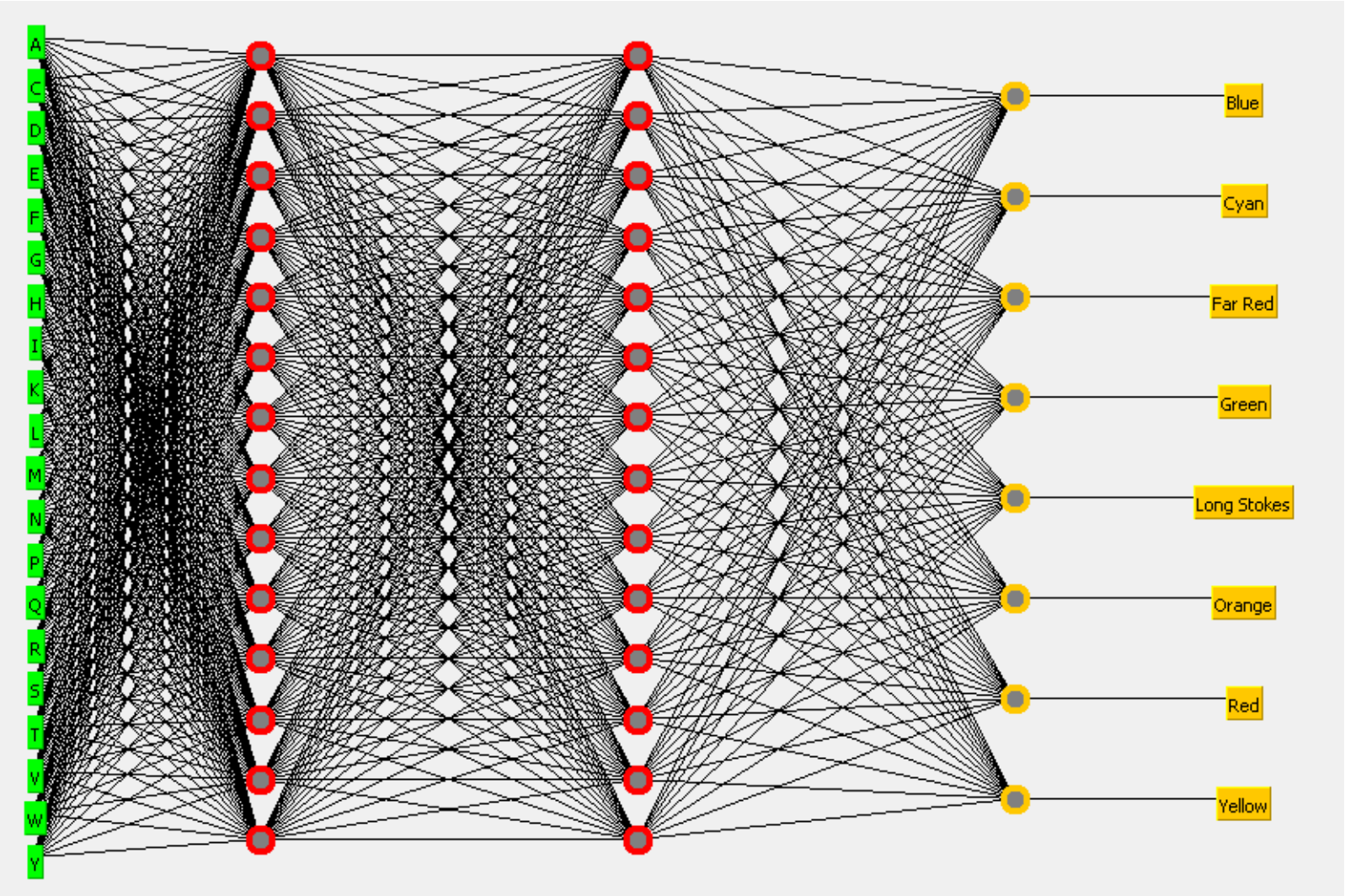


Figura 18: Representação gráfica da Rede Neural Artificial construída pelo algoritmo *MultilayerPerceptron* no software WEKA - do autor

O algoritmo LibSVM construiu o classificador utilizando a técnica de SVM através do ambiente *Explorer* do software WEKA, conforme os parâmetros listados na Tabela 14.

Parâmetro	*Valor*
seed	1
SVMType	C-SVC
kernelType	FBR
gamma	0.03125
shrinking	Verdadeiro
eps	0.001
cost	2.0
weights	(em branco)
normalize	Falso
probabilityEstimates	Falso

Tabela 14: Lista de parâmetros utilizados na execução do algoritmo *LibSVM* no software WEKA - do autor

O tipo de SVM utilizada, como indica o parâmetro *SVMType*, foi a C-SVC, ou seja, uma SVM de Margens Suaves própria para tarefas de classificação. Esta SVM se caracteriza por admitir que algumas instâncias sejam erroneamente classificadas, através de folgas de tolerância nas margens separadoras das classes (parâmetro *eps*, utilizado aqui como critério de parada). Uma penalização é imposta quando isso ocorrer, dada pelo parâmetro *cost* que representa o custo dessa violação de restrição, (CHANG; LIN, 2011) (SCHÖLKOPF; SMOLA, 2002).

O tipo de função *kernel* utilizada foi a FBR (ou *kernel* Gaussiano), conforme indica o parâmetro *kernelType*. Esta é a função mais utilizada em problemas de classificação, além de possuir somente um parâmetro para configuração, o *gamma*. Testes empíricos demonstram que a efetividade do algoritmo de SVM depende da seleção do *kernel*, dos parâmetros do *kernel* (*gamma*, neste caso) e do parâmetro de custo do tipo de SVM (*cost*, neste caso) (CHANG; LIN, 2011) (LORENA; CARVALHO, 2007).

Como estes parâmetros *cost* e *gamma* variam em algumas ordens de grandeza, pode-se usar uma função *gridsearch* com valores crescentes para selecioná-los. A função *GridSearch*, também implementada no software WEKA, testa valores de cada um dos dois parâmetros dentro de uma faixa específica de busca, usando passos geométricos, ou seja, busca pelos melhores valores desses parâmetros através da análise dos resultados obtidos com a execução do próprio algoritmo *LibSVM* para um intervalo de valores (CHANG; LIN, 2011) (LORENA; CARVALHO, 2007). Dessa forma, para o conjunto de dados apresentado, a função selecionou os valores: $cost = 2$ e $gamma = 0.03125$.

Além destes parâmetros, cabe destacar também o parâmetro *shrinking* (= Verdadeiro), responsável por definir a aplicação de uma heurística de diminuição. Esta heurística trata da redução do tamanho do problema, temporariamente eliminando elementos que dificilmente serão selecionados para o problema de otimização, uma vez que já atingiram os seus limites durante as iterações (JOACHIMS, 1998).

8.2 Métricas de Desempenho

A primeira métrica a ser abordada é a Matriz de Confusão, que contabiliza o número de classificação corretas e incorretas para cada classe existente nos dados realizadas sobre o próprio conjunto de dados através do método de avaliação de validação cruzada em 10 partições. A matriz de confusão de cada um dos classificadores foi obtida após a construção dos mesmos no ambiente *Explorer* do software WEKA.

As Tabelas 15, 16 e 17 representam as matrizes de confusão dos três classificadores, obtidas através do método de avaliação de validação cruzada em 10 partições utilizando-se o próprio conjunto de dados para estimar estas métricas de desempenho.

Classificado como –>	Blue	Cyan	Far Red	Green	L. Stokes	Orange	Red	Yellow
Blue	**2**	1	0	1	0	0	0	0
Cyan	1	**10**	0	1	0	0	0	1
Far Red	0	0	**17**	0	1	0	3	0
Green	1	2	0	**20**	0	0	1	3
L. Stokes	0	0	0	0	**3**	0	0	0
Orange	0	0	0	0	0	**7**	0	0
Red	0	0	4	0	0	1	**11**	1
Yellow	1	0	0	0	0	1	0	**15**

Tabela 15: Matriz de Confusão obtida pela avaliação, através do método de validação cruzada em 10 partições, do classificador construído pelo algoritmo J48 - do autor

Classificado como –>	Blue	Cyan	Far Red	Green	L. Stokes	Orange	Red	Yellow
Blue	**0**	1	0	3	0	0	0	0
Cyan	0	**9**	1	3	0	0	0	0
Far Red	0	0	**16**	1	0	0	4	0
Green	1	4	0	**15**	0	0	1	6
L. Stokes	0	0	0	0	**2**	0	0	1
Orange	0	0	2	0	0	**4**	1	0
Red	0	0	3	0	0	1	**12**	1
Yellow	0	0	0	2	0	0	1	**14**

Tabela 16: Matriz de Confusão obtida pela avaliação, através do método de validação cruzada em 10 partições, do classificador construído pelo algoritmo *MultilayerPerceptron* - do autor

Classificado como –>	Blue	Cyan	Far Red	Green	L. Stokes	Orange	Red	Yellow
Blue	**2**	0	0	2	0	0	0	0
Cyan	0	**11**	0	2	0	0	0	0
Far Red	0	0	**16**	1	0	1	3	0
Green	1	1	0	**25**	0	0	0	0
L. Stokes	0	0	0	0	**3**	0	0	0
Orange	0	0	1	0	0	**6**	0	0
Red	0	0	1	1	0	0	**14**	1
Yellow	0	0	0	3	0	0	1	**13**

Tabela 17: Matriz de Confusão obtida pela avaliação, através do método de validação cruzada em 10 partições, do classificador construído pelo algoritmo LibSVM - do autor

As demais métricas de desempenho, tratadas a seguir, foram calculadas e obtidas utilizando-se o ambiente *Experimenter* no software WEKA, através da execução de 1000 iterações consecutivas de cada um dos três algoritmos, avaliados pelo método de validação cruzada em 10 partições. Os valores apresentados representam a média das métricas para as 1000 iterações realizadas.

Os valores das métricas de desempenho Acurácia (porcentagem de classificações corretas) e Erro (porcentagem de classificações incorretas), para os três classificadores em avaliação, estão listados na Tabela 18.

Classificador	Acurácia	Erro
J48	0.7865 ou 78.65 %	0.2135 ou 21.35 %
MultilayerPerceptron	0.6831 ou 68.31 %	0.3169 ou 31.69 %
LibSVM	0.8219 ou 82.19 %	0.1781 ou 17.81 %

Tabela 18: Resultados das métricas de desempenho Acurácia e Erro para os classificadores avaliados - do autor

Os valores das métricas de desempenho Precisão, Revocação e *F-Measure*, para os três classificadores em avaliação, estão listados na Tabela 19.

Classificador	Precisão	Revocação	*F-Measure*
J48	0.815	0.786	0.773
MultilayerPerceptron	0.699	0.683	0.673
LibSVM	0.824	0.822	0.800

Tabela 19: Resultados das métricas de desempenho Precisão, Revocação e *F-Measure* para os classificadores avaliados - do autor

Os valores da métrica de desempenho Índice *Kappa*, para os três classificadores em avaliação, estão listados na Tabela 20.

Classificador	Índice *Kappa*
J48	0.744
MultilayerPerceptron	0.643
LibSVM	0.781

Tabela 20: Resultados da métrica de desempenho Índice *Kappa* para os classificadores avaliados - do autor

Com a finalidade de facilitar a visualização comparativa das métricas obtidas na avaliação de desempenho dos classificadores, o gráfico de colunas ilustrado na Figura 19 foi elaborado, agrupando as 4 métricas a serem comparadas com seus respectivos pares.

Observando-se o gráfico ilustrado na Figura 19 é possível descrever duas características em relação às métricas de desempenho calculadas. Primeiramente, percebe-se uma aparente superioridade das métricas de desempenho referentes aos classificadores baseados em SVM e AD frente às métricas do classificador baseado em RNA. Além disso, nota-se que, dentre as diferente métricas de desempenho de um mesmo classificador, os valores das métricas encontram-se dentro de um mesmo patamar numérico. Este

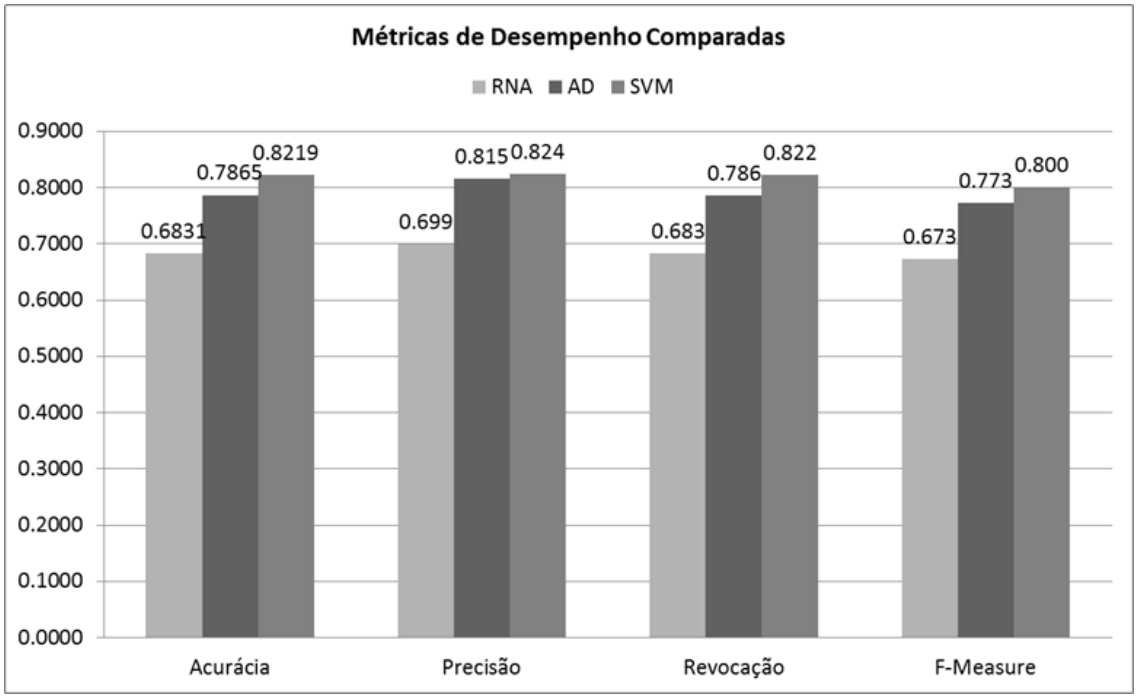


Figura 19: Representação gráfica em colunas das métricas de desempenho para os 3 classificadores construídos - do autor

fato sugere que este conjunto de métricas de desempenho potencialmente demonstra o desempenho geral dos classificadores.

8.3 Teste Estatístico

O *teste-t* pareado corrigido, realizado no ambiente *Experimenter* da ferramenta WEKA, através de um experimento, comparou os conjuntos de valores das métricas de desempenho acurácia, precisão, revocação e *F-measure* de cada par de classificadores.

Este experimento foi configurado estipulando 1000 iterações para cada algoritmo, com o método de avaliação de validação cruzada em 10 partições. Como o particionamento dos conjuntos na validação cruzada é aleatório, optou-se por um grande número de iterações dos algoritmos para minimizar qualquer vício nas métricas de desempenho obtidas pelo método de validação cruzada.

Ao final da execução do experimento, os valores das métricas de desempenho apresentados como resultado pelo software representam a média dos valores calculados para cada métrica nas 1000 iterações de cada algoritmo (BOUCKAERT et al., 2015).

A partir destes conjuntos de valores foi realizada a comparação em si dos classificadores, através do *teste-t* pareado corrigido, para indicar se há ou não diferença estatisticamente significativa entre as médias de valores de cada métrica para cada par de classificadores.

Os resultados desse teste estatístico se apresentam no seguinte formato nas tabelas a

seguir, *(x/y/z)*, onde *x*, *y* e *z* podem assumir 0 ou 1. Comparando-se dois classificadores, este resultado indica se o valor da métrica do segundo classificador é estatisticamente melhor (*x*), o mesmo (*y*) ou pior (*z*), que o valor da métrica do primeiro classificador, para um nível de confiança de 95%.

Apesar deste formato de resultado disponibilizado pelo ambiente *Experimenter*, sob o ponto de vista formal, este *teste-t* pareado corrigido realiza o seguinte teste de hipóteses:

- hipótese nula h_0: não há diferença entre as médias dos valores das métricas de desempenho dos classificadores;
- hipótese alternativa h_1: há diferença entre as médias dos valores das métricas de desempenho dos classificadores.

Na Tabela 21 estão agrupados os resultados do *teste-t* realizado comparando-se as métricas do classificador baseado em Redes Neurais Artificiais com as métricas do classificador baseado em Árvores de Decisão, de acordo com o disponibilizado pelo ambiente *Experimenter*.

Métrica	*Valor AD*	*Valor RNA*	*Resultado Teste-t*
Acurácia	0.786	0.683	(0 / 0 / **1**)
Precisão	0.815	0.699	(0 / 0 / **1**)
Revocação	0.786	0.683	(0 / 0 / **1**)
F-Measure	0.773	0.673	(0 / 0 / **1**)

Tabela 21: Resultados do *teste-t* pareado corrigido comparando as métricas de desempenho do classificador baseado em RNA com as métricas do classificador baseado em AD - do autor

Na Tabela 22 estão agrupados os resultados do *teste-t* realizado comparando-se as métricas do classificador baseado em Máquinas de Vetores de Suporte com as métricas do classificador baseado em Redes Neurais Artificiais, de acordo com o disponibilizado pelo ambiente *Experimenter*.

Métrica	*Valor RNA*	*Valor SVM*	*Resultado Teste-t*
Acurácia	0.683	0.822	(**1** / 0 / 0)
Precisão	0.699	0.824	(**1** / 0 / 0)
Revocação	0.683	0.822	(**1** / 0 / 0)
F-Measure	0.673	0.800	(**1** / 0 / 0)

Tabela 22: Resultados do *teste-t* pareado corrigido comparando as métricas de desempenho do classificador baseado em SVM com as métricas do classificador baseado em RNA - do autor

Na Tabela 23 estão agrupados os resultados do *teste-t* realizado comparando-se as métricas do classificador baseado em Árvores de Decisão com as métricas do classificador baseado em Máquinas de Vetores de Suporte, de acordo com o disponibilizado pelo ambiente *Experimenter*.

Métrica	*Valor SVM*	*Valor AD*	*Resultado Teste-t*
Acurácia	0.822	0.786	(0 / **1** / 0)
Precisão	0.824	0.815	(0 / **1** / 0)
Revocação	0.822	0.786	(0 / **1** / 0)
F-Measure	0.800	0.773	(0 / **1** / 0)

Tabela 23: Resultados do *teste-t* pareado corrigido comparando as métricas de desempenho do classificador baseado em AD com as métricas do classificador baseado em SVM - do autor

Pode-se observar que o resultado do *teste-t* foi sempre o mesmo para os mesmos pares de classificadores, independente de qual métrica de desempenho estava sendo testada. Então, em resumo, os resultados obtidos no *teste-t* pareado corrigido são os seguintes:

- AD x RNA: com um nível de confiança de 95%, rejeita-se a hipótese nula h_0 de que não há diferença entre as médias das métricas de desempenho dos classificadores baseados em AD e RNA. Assim, rejeita-se h_0 em favor de h_1, o que indica que as médias das métricas de desempenho são diferentes;

- RNA x SVM: com um nível de confiança de 95%, rejeita-se a hipótese nula h_0 de que não há diferença entre as médias das métricas de desempenho dos classificadores baseados em RNA e SVM. Assim, rejeita-se h_0 em favor de h_1, o que indica que as médias das métricas de desempenho são diferentes;

- SVM x AD: com um nível de confiança de 95%, não rejeita-se a hipótese nula h_0 de que não há diferença entre as médias das métricas de desempenho dos classificadores baseados em SVM e AD. Assim, não rejeita-se h_0, o que indica que as médias das métricas de desempenho não têm diferença.

8.4 Classificação da Proteína inédita

Após a construção dos 3 classificadores, submeteu-se a eles, através do ambiente *Explorer* do software WEKA, um conjunto de dados de teste contendo 4 instâncias que representam as modelagens realizadas via software para a proteína inédita produzida pelo projeto de pesquisa Peixes Transgênicos Fluorescentes.

Registro	*Modelagem*	*Classificador AD*	*Classificador RNA*	*Classificador SVM*
#1	*Modeller*	Red	Green	Green
#2	*i-Tasser*	Red	Red	Green
#3	*Phyre2*	Red	Red	Green
#4	*Mutação Modeller*	Red	Red	Red

Tabela 24: Relação das classificações realizadas pelos 3 classificadores construídos sobre o conjunto de teste que representa 4 modelagens estruturais de uma proteína inédita - do autor

Esta proteína inédita, de maneira excepcional, emite fluorescência em duas faixas

de cores distintas, o que a faz pertencer simultaneamente às classes *Red* e *Green*. Assim, a predição tanto da classe *Green* quanto da classe *Red* é considerada correta nesta classificação.

O resultado da classificação realizada por cada um dos 3 classificadores para cada um dos 4 registros do conjunto de testes estão reunidos na Tabela 24. Nesta tabela, a coluna Modelagem refere qual das modelagens da proteína inédita o registro representa. Também é possível notar que os classificadores receberam a nomenclatura das técnicas que representam, de forma a facilitar sua identificação.

9 DISCUSSÃO

O objetivo do presente trabalho é comparar três classificadores, já utilizados separadamente em trabalhos relacionados ao tema, para investigar sua performance no problema de predição da classe de cor de proteínas fluorescentes.

Paralelamente, também é necessário destacar a contribuição do processo completo de KDD aplicado na construção dos três classificadores, com cada uma das 5 etapas do processo devidamente abordadas em relação aos dados e ao problema da predição da classe de cor em proteínas fluorescentes.

A vinculação do trabalho como parte do projeto Peixes Transgênicos Fluorescentes, onde pretende-se que os pesquisadores do projeto possam construir e utilizar um classificador como ferramenta de apoio à decisão no desenvolvimento de novas variantes de proteínas fluorescentes, orientou a decisão de limitar algumas tarefas de pré-processamento no conjunto de dados de treinamento dos classificadores aqui comparados.

De maneira geral, a intenção é que a partir do sistema BDPF, que armazena e organiza os dados das proteínas fluorescentes, os pesquisadores gerem o arquivo representativo do conjunto de dados de treinamento dos classificadores com total autonomia. Sempre que registros de novas proteínas fluorescentes são adicionados no BDPF, novos classificadores podem ser gerados tendo como base de treinamento este novo e mais completo conjunto de dados.

Essa intenção de uso do sistema BDPF e do classificador de forma conjunta evidencia que qualquer tarefa de pré-processamento aplicada especificamente ao conjunto de treinamento das 109 instâncias utilizado neste trabalho, não seria necessariamente adequada quando da existência de um novo conjunto de dados com mais instâncias. Este fato justificou a opção de utilizar para o treinamento e posterior avaliação de desempenho dos três classificadores o conjunto de dados diretamente extraído do BDPF, sem realizar um método de balanceamento das classes, por exemplo.

Outro ponto que necessita ser abordado, neste mesmo aspecto, é a opção de não normalização dos dados numéricos referentes aos atributos representativos das distâncias dos aminoácidos para o grupo cromóforo. De acordo com a literatura, sabe-se que a normalização de dados tende a melhorar os modelos de classificação baseados em RNA e

SVM. Porém, a normalização dos dados numéricos acaba por descaracterizar as distâncias calculadas. Embora não seja um problema nos classificadores baseados em RNA e SVM, uma vez que são caixa-preta, a descaracterização das distâncias causa problemas de interpretação do conhecimento no classificador baseado em AD.

Para uma comparação formalmente correta, não seria viável utilizar um conjunto de dados normalizado para o treinamento dos classificadores baseados em RNA e SVM e outro não normalizado para o classificador baseado em AD. Assim, optou-se pelo uso do conjunto de dados não normalizado, que não prejudica nenhum dos modelos de classificação e mantém a hegemonia necessária na comparação dos classificadores, mesmo que no caso das RNA e SVM os melhores classificadores possíveis (potencialmente) não sejam construídos.

Neste contexto, passa-se à discussão dos resultados quantitativos e qualitativos produzidos no trabalho, com enfoque na comparação dentre os três classificadores construídos e avaliados.

Em relação aos modelos de classificação construídos, Seção 8.1 do capítulo Resultados, é possível notar claramente as diferenças entre os classificadores e suas características. Tanto o classificador baseado em AD quanto o baseado em RNA apresentam representação gráfica de seus modelos de classificação, porém de forma distintas, ao passo que o classificador baseado em SVM não apresenta nenhuma representação gráfica ou matemática do modelo construído.

Embora seja considerado um poderoso classificador pela comunidade de aprendizado de máquina devido aos bons resultados apresentados nos últimos anos, um classificador baseado em SVM caracteriza-se por não evidenciar ao meio externo os padrões e o conhecimento extraído do conjunto de dados, ou seja, é um classificador caixa-preta.

Dependendo da aplicação, no entanto, pode ser importante ou necessário ter a disposição dos especialistas o conhecimento extraído dos dados, o que pode impossibilitar a utilização de classificadores baseados em SVM nestes casos, independente da possível superioridade de seus resultados.

Confirmando sua característica, no classificador gerado pelo algoritmo *LibSVM* do software WEKA, todo o conhecimento extraído do conjunto de dados das proteínas fluorescentes encontra-se codificado em equações internas ao modelo de classificação. A forma de utilizar o conhecimento extraído é através do próprio classificador, utilizando-o para a classificação de novos exemplos.

O classificador baseado em RNA também é *caixa-preta*, apesar do modelo de classificação construído pelo algoritmo *MultilayerPerceptron* ilustrar a topologia da rede com as camadas de entrada, as intermediárias e de saída, os neurônios que formam cada camada e as conexões entre eles (Figura 18).

Afinal, mesmo de posse destas informações relativas ao modelo de classificação, o conhecimento extraído dos dados analisados está codificado em equações matemáticas.

De maneira similar ao classificador baseado em SVM, dependendo de sua aplicação, esta característica pode não ser favorável à utilização do classificador baseado em RNA.

O classificador baseado em AD, por sua vez, apresenta um modelo de classificação interpretável, onde os padrões e o conhecimento extraído do conjunto de dados estão expostos ao analista através da própria representação gráfica do modelo. A árvore de decisão construída pelo algoritmo J48 (Figura 17) demonstra em seus nós de decisão quais os atributos do conjunto de dados são os mais decisivos para a definição da classe de cor das proteínas fluorescentes.

Dentre os 20 atributos que representam os aminoácidos, observa-se que 9 deles foram utilizados pelo algoritmo J48 no modelo de classificação. Na raiz da árvore de decisão, representando o atributo mais decisivo na definição da classe de cor, encontra-se o Triptofano (*W*). De maneira geral, através da divisão binária dos valores das distâncias para a criação dos ramos da árvores de decisão, este classificador vincula a proximidade ou distanciamento dos aminoácidos com a definição das classes de cores nas proteínas fluorescentes.

É interessante observar ainda o agrupamento de classes de cores que ocorre já inicialmente, a partir da ramificação do nó-raiz. Segundo o modelo, proteínas fluorescentes que possuem em sua estrutura algum aminoácido Triptofano mais próximo do cromóforo (distância $<= 6.8$ Å) pertencem às classes que representam cores de comprimento de onda de emissão mais longo: *Orange*, *Red*, *Far Red* e *Long Stokes*. Já as proteínas fluorescentes que apresentam o aminoácido Triptofano mais distante do cromóforo (distância > 6.8 Å) pertencem às classes de cores de comprimento de onda de emissão mais curto: *Blue*, *Cyan*, *Green* e *Yellow*.

Esta observação pode ser facilmente confirmada tendo por base a Tabela 25, da divisão dos comprimentos de onda de emissão em classes de cores. É possível verificar como a presença do aminoácido Triptofano mais próximo ou mais distante do grupo cromóforo divide perfeitamente as classes de cores em dois grupos, segundo a faixa de emissão de comprimento de onda.

Classe de cor	*Comprimentos de onda*	*Presença de Triptofano*
Blue	440nm - 470nm	> 6.8 Å
Cyan	470nm - 500nm	
Green	500nm - 525nm	
Yellow	525nm - 555nm	
Orange	555nm - 580nm	$<= 6.8$ Å
Red	580nm - 630nm	
Far-Red	630nm - 700nm	
Long-Stoke	Acima de 570nm	

Tabela 25: Resumo da influência do aminoácido Triptofano, mais próximo ou mais distante do grupo cromóforo, na definição da classe de cor das proteínas fluorescentes - dados de (OLENYCH et al., 2007) e do autor

Um padrão que pode ser extraído da interpretação deste modelo de classificação, com

o auxílio da Tabela 25, é o seguinte:

- proteínas fluorescentes que têm em sua estrutura tridimensional um aminoácido Triptofano a uma distância do cromóforo maior do que 6.8 Å, vão emitir fluorescência com comprimento de onda de, no máximo, 555nm;

- proteínas fluorescentes que têm em sua estrutura tridimensional um aminoácido Triptofano a uma distância do cromóforo menor ou igual a 6.8 Å, vão emitir fluorescência com comprimento de onda a partir de 555nm.

Diversos outros padrões e conhecimentos podem ser extraídos do modelo de classificação de AD, dependendo do que se busca interpretar ou entender do conjunto de dados em questão. Para tanto, basta percorrer, a partir do nó-raiz, diferentes ramos da árvore de decisão e realizar a extração de regras de decisão e padrões de comportamento. Essa capacidade de interpretação do conhecimento é uma característica intrínseca ao classificador baseado em AD.

Após esta análise puramente qualitativa dos modelos de classificação construídos, que permitiu a identificação de características e capacidades inerentes a cada um deles, prossegue-se para uma análise simultaneamente quantitativa e qualitativa. Isso se explica pois, através de algumas métricas de desempenho dos classificadores, certos fenômenos qualitativos sobre os modelos de classificação podem ser observados e confirmados.

A construção individual dos classificadores através do ambiente *Explorer* do software WEKA permite a avaliação de desempenho utilizando-se o seu próprio conjunto de treinamento. Isso é feito pelo método de avaliação de validação cruzada em 10 partições, considerado um método eficiente pois alia rapidez na execução com uma boa estratégia de particionamento para estimar o desempenho dos classificadores através do próprio conjunto de treinamento dos mesmos.

Analisando-se as Tabelas 15, 16 e 17, que representam as matrizes de confusão dos classificadores, com suas contagens de classificações corretas e incorretas realizadas sobre o conjunto de treinamento através da validação cruzada, é possível verificar se o forte desbalanceamento entre o número de instâncias das classes produziu os fenômenos de *overfitting* para a classe majoritária *Green* ou *underfitting* para a classe minoritária *Long Stokes*.

Os fenômenos de *overfitting* para a classe majoritária e *underfitting* para a classe minoritária podem ser verificados através das métricas de precisão e revocação referentes às classes em questão. Se a classe majoritária apresentar uma alta revocação associada a uma precisão relativamente baixa, indica a ocorrência do fenômeno de *overfitting* para esta classe.

Ao apresentar uma alta revocação para a classe majoritária, o classificador demonstra que possui uma alta capacidade em reconhecer todas as instâncias desta classe. Associado

a isso, uma precisão não tão alta para esta mesma classe majoritária demonstra que o mesmo classificador não possui uma boa capacidade em reconhecer esta classe e rejeitar as demais.

Em outras palavras, este par de métricas nesta configuração citada demonstra que o classificador reconhece quase todas as instâncias da classe majoritária (revocação alta) e reconhece erroneamente diversas outras instâncias como pertencentes à classe majoritária (precisão relativamente baixa). Tal situação evidencia o fenômeno de (*overfitting*) para a classe majoritária, especialmente quando esta condição do par de métricas de precisão e revocação não ocorre nas demais classes.

De forma análoga, se a classe minoritária apresentar uma baixíssima revocação indica a presença do fenômeno de *underfitting* para esta classe, pois demonstra que o classificador possui pouca capacidade em reconhecer as instâncias desta classe.

Para auxiliar nesta análise, o cálculo das métricas precisão e revocação para as classes *Green* (majoritária) e *Long Stokes* (minoritária) foi feito conforme demonstra a Tabela 26.

Classificador	*Classe*	*Precisão*	*Revocação*
AD	Green	0.909	0.741
	Long Stokes	0.750	1.00
RNA	Green	0.625	0.556
	Long Stokes	1.00	0.667
SVM	Green	0.735	0.925
	Long Stokes	1.00	1.00

Tabela 26: Valores das métricas de desempenho Precisão e Revocação, calculadas a partir das matrizes de confusão, para as classes *Green* e *Long Stokes* de cada classificador - do autor

Observando-se as métricas relativas à classe minoritária *Long Stokes* na Tabela 26 é possível identificar que nenhum dos classificadores apresentou o fenômeno de *underfitting* para a classe minoritária. A métrica revocação maior do que zero indica que o desbalanceamento entre as classes não causou o fenômeno de *underfitting* para a classe minoritária. Ou seja, mesmo com poucas instâncias desta classe, estes classificadores conseguem reconhecer instâncias pertencentes à ela.

Similarmente, após análise das métricas referentes à classe majoritária *Green* pode-se identificar a ocorrência do fenômeno de *overfitting* para esta classe no classificador baseado em SVM, uma vez que ele apresenta revocação de 0.925 associada à uma precisão de 0.735. Em relação aos outros dois classificadores, baseados em AD e RNA, as métricas não indicam o fenômeno de *overfitting* para a classe *Green*, evidenciando que apesar do desbalanceamento entre o número de registros das classes, estes classificadores não estão ajustados demasiadamente à classe majoritária.

Referente a ocorrência dos fenômenos de *overfitting* para a classe majoritária e *underfitting* para a classe minoritária, devido ao desbalanceamento entre as classes, conclui-se que os classificadores baseados em AD e RNA não apresentam nenhum dos fenômenos.

Por outro lado, o classificador baseado em SVM apresenta *overfitting* para a classe *Green*.

Em momento oportuno, todas as análises e considerações discutidas anteriormente em relação às características, capacidades e qualidade dos classificadores são aplicadas no cumprimento do objetivo do trabalho. Porém, além destas análises, a partir daqui segue-se uma análise puramente quantitativa das métricas de desempenho dos classificadores.

As métricas de desempenho apresentadas nas Tabelas 18, 19 e 20, no capítulo 8, são o resultado da média aritmética das métricas de avaliação de desempenho medidas em 1000 execuções de cada um dos 3 algoritmos. Da mesma forma, as métricas de precisão, revocação e *F-measure* de cada classificador em cada execução são a média destas métricas para cada uma das classes deste classificador.

Antes de tratar do teste estatístico comparativo das métricas de desempenho, analisa-se os valores do índice *Kappa* (Tabela 20). Este índice expressa a confiabilidade de um classificador, medindo o nível de concordância entre as classes preditas e as classes verdadeiras nas classificações, segundo a escala de valores da Tabela 3. Por possuir esta escala de valores que indica o significado para o valor numérico e permite a comparação direta entre diferentes índices, o índice *Kappa* não foi incluído no teste estatístico.

Consultando a Tabela 3, verifica-se que os classificadores baseados em RNA, AD e SVM possuem uma confiabilidade Boa (respectivamente, 0.643, 0.744 e 0.781), pois todos apresentam valores acima de 0.6. Porém, cabe destacar que os classificadores baseados em AD e SVM possuem índice *Kappa* acima de 0.70, o que é considerado como preferencial em relação à confiabilidade, no consenso dos estatísticos.

A conclusão desta primeira análise é que, segundo o índice *Kappa*, a confiabilidade dos três classificadores é a mesma, contudo, pode-se considerar a confiabilidade dos classificadores baseados em AD e SVM superior à confiabilidade do classificador baseado em RNA, pois são valores acima de 0.7. Com este resultado em mente, prossegue-se ao *teste-t* estatístico com as métricas de acurácia, precisão, revocação e *F-measure*.

A métrica de desempenho *F-measure* é utilizada pois ela representa a média harmônica entre a precisão e a revocação. Como tais medidas podem ser enganosas quando avaliadas separadamente, o uso da métrica *F-measure* ratifica a análise sobre estas medidas, pois um valor alto de *F-measure* só é obtido com altos valores de precisão e revocação.

Diferentemente do índice *Kappa*, estas quatro métricas não possuem uma escala hierárquica de valores que indica um significado para os seus valores numéricos, porém quanto maior seus valores, há a indicação de um desempenho superior do classificador. Então, nesse contexto, foi necessário comparar estatisticamente os valores destas métricas para as 1000 iterações dos algoritmos para ser possível identificar superioridade, inferioridade ou similaridade dentre os classificadores. O valor médio destas métricas estão elencados nas Tabelas 18 e 19.

O *teste-t* pareado corrigido, que considera no cálculo do *valor-t* a dependência entre

os conjuntos de treinamento e teste, é realizado entre as médias das mesmas métricas, tomadas aos pares de classificadores. Para cada par de classificadores, as médias de determinada métrica de um classificador são comparadas, segundo uma significância de 5%, com as médias da mesma métrica de outro classificador, produzindo um resultado no qual é indicado se há diferença ou não entre as médias das métricas de desempenho dos dois classificadores.

Uma bateria de testes estatísticos nas configurações citadas foram executados, conforme mostram os resultados: Tabela 21, comparando-se RNA vs. AD; Tabela 22, comparando-se SVM vs. RNA; e Tabela 23, comparando-se AD vs. SVM.

É possível verificar nos resultados que, independente da métrica utilizada, o teste estatístico concluiu afirmações semelhantes para os mesmos pares de classificadores. Para um nível de confiança de 95%, não há diferença estatística significativa entre as médias das métricas de desempenho do classificador baseado em AD e do baseado em SVM. E, para o mesmo nível de confiança, as médias das métricas dos classificadores baseados em AD e SVM são estatisticamente diferentes às médias das métricas do classificador baseado em RNA.

Sob o ponto de vista que estas 4 métricas reunidas são bons indicadores do desempenho geral dos classificadores e levando em consideração os resultados do *teste-t*, conclui-se que tanto o classificador baseado em AD quanto o classificador baseado em SVM apresentam o mesmo nível de desempenho, enquanto o classificador baseado em RNA é inferior a ambos.

Retomando o resultado da análise do índice *Kappa*, o qual apontou confiabilidade Boa para os três classificadores, apesar de indicar valores do índice preferenciais para os classificadores baseados em AD e SVM, nota-se que a conclusão do *teste-t* é reforçada e ratificada pelos níveis de confiabilidade indicados no índice *Kappa*, pois demonstram superioridade dos classificadores baseados em AD e SVM sobre o baseado em RNA.

Além das avaliações e comparações já abordadas, também foi realizada a aplicação dos classificadores sobre um conjunto de teste contendo 4 instâncias, para avaliar a acurácia dos modelos na classificação de exemplos não utilizados no treinamento.

Para tanto, a partir de uma variante inédita de uma proteína fluorescente produzida no projeto Peixes Transgênicos Fluorescentes, gerou-se 4 potenciais modelagens para a estrutura tridimensional desta nova proteína, a qual ainda não possui uma estrutura experimentalmente resolvida. Estas 4 estruturas, que representam a mesma proteína neste caso, foram formatadas adequadamente e formaram o conjunto de teste para os classificadores.

Excepcionalmente, esta proteína fluorescente inédita emite fluorescência em duas faixas de cores, o que a caracteriza como pertencente às classes *Green* e *Red* simultaneamente. Este fato, consequentemente, orienta que a classificação das instâncias deste conjunto seja considerada como correta nas situações em que as classes *Green* ou *Red* forem preditas.

Os resultados das classificações realizadas sobre este conjunto de teste, dispostos na Tabela 24, mostra que os três classificadores classificaram corretamente as 4 instâncias, indicando sempre as classes *Green* ou *Red* como resultado na classificação.

Com este último resultado quantitativo, percebe-se que, embora exista superioridade de desempenho dos classificadores baseados em AD e SVM sobre o classificador baseado em RNA, na classificação deste exemplo inédito os três classificadores foram capazes de classificá-lo corretamente.

Considerando-se estas análises puramente quantitativas, pode-se considerar que os classificadores baseados em AD e SVM encontram-se empatados em desempenho, pois apresentam o mesmo índice de confiabilidade, suas métricas são estatisticamente as mesmas e ambos classificaram todos os exemplos do conjunto de teste corretamente, ao passo que o classificador baseado em RNA, mesmo apresentando a mesma confiabilidade e acurácia no conjunto de teste que os demais classificadores, apresenta métricas de desempenho significativamente inferiores. Porém, o termo performance, tratado no objetivo deste trabalho, vai além das métricas de desempenho quantitativas. A performance refere-se também às qualidades do classificador e sua adequação ao problema no âmbito do projeto Peixes Transgênicos Fluorescentes.

Inicialmente, é necessário relembrar as características e capacidades dos dois classificadores de interesse. O classificador baseado em AD produz um modelo gráfico passível de ser interpretado e no qual os padrões de comportamento dos dados podem ser identificados. De maneira oposta, o classificador baseado em SVM é caixa-preta, onde nenhum modelo gráfico interpretável é produzido e o conhecimento extraído dos dados encontra-se codificado em equações internas ao classificador.

Além disso, também é preciso levar em consideração a análise quanto a ocorrência ou não dos fenômenos de *overfitting* para a classe majoritária e *underfitting* para a minoritária, devido ao desbalanceamento do número de registro entre as classes, nestes classificadores. Observou-se anteriormente que, enquanto o classificador baseado em SVM apresentou *overfitting* para a classe *Green*, o classificadores baseados em AD e RNA não apresentaram a ocorrência de nenhum dos fenômenos, demonstrando uma maior qualidade neste quesito.

Soma-se a isso algumas motivações para o uso de modelos de classificação interpretáveis na predição envolvendo proteínas. A primeira motivação refere-se a melhorar a confiança dos especialistas na predição em si, uma vez que entender a predição realizada pelo modelo ajuda o especialista a ter mais confiança no classificador. Consequentemente, possuindo uma alta confiança nas predições, os especialistas tornam-se mais dispostos a investir tempo e recursos para realizar os experimentos biológicos necessários para a confirmação das previsões (FREITAS; WIESER; APWEILER, 2010).

A segunda motivação é a possibilidade do especialista explorar novas perspectivas sobre os dados e o problema associado, permitindo um entendimento mais amplo so-

bre os mecanismos ou caraterísticas das proteínas em estudo. Inclusive, um modelo de predição devidamente interpretado por um especialista, pode prover evidências para confirmar ou rejeitar hipóteses já criadas, ou até orientá-lo a formular novas hipóteses biológicas (FREITAS; WIESER; APWEILER, 2010).

Outra razão, ainda, é interpretar o modelo de classificação com o intuito de potencialmente detectar erros no classificador, possivelmente causado por erros nos dados. Afinal, as principais fontes de erros em predições são oriundas de dados de treinamento limitados em quantidade ou qualidade, ou ambos (FREITAS; WIESER; APWEILER, 2010).

Devido às motivações recém abordadas, conclui-se que classificador baseado em AD, por possuir um modelo de classificação interpretável, demonstra qualidades mais desejáveis do que o classificador baseado em SVM, no contexto de um projeto de pesquisa sobre proteínas fluorescentes conduzido por especialistas das áreas de biologia molecular e oceanologia.

Em especial, essas qualidades são desejáveis pois o projeto Peixes Transgênicos Fluorescentes tem por objetivo utilizar um classificador como ferramenta de apoio à pesquisa. Então, no momento em que o classificador baseado em AD apresenta todas as características desejáveis mencionadas nas motivações do uso de modelos de classificação interpretáveis, ele demonstra uma performance mais desejada em relação às qualidades dos classificadores.

Tal afirmação pode ser feita baseando-se na premissa de que, ao possuir qualidades mais alinhadas às necessidades e características do problema a ser aplicado, o classificador baseado em AD contribui de maneira mais efetiva com a extração de novos conhecimentos e padrões do conjunto de dados, contemplando plenamente o objetivo básico do processo de KDD.

Diante do exposto, fica evidente que o classificador que apresentou a melhor performance na predição da classe de cor de proteínas fluorescentes no âmbito do projeto Peixes Transgênicos Fluorescentes é o classificador baseado em AD. Ele é quantitativamente, em relação às métricas de desempenho, similar ao classificador baseado em SVM e superior ao classificador baseado em RNA, é qualitativamente superior ao classificador baseado em SVM, não apresentando fenômeno de *overfitting* para a classe majoritária e mostrou-se mais adequado às necessidades de uso no projeto de pesquisa.

10 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho foi desenvolvido um processo comparativo sobre classificadores para investigação de sua performance na predição da classe de cor, relacionada à estrutura tridimensional da molécula, em proteínas fluorescentes, no âmbito do projeto Peixes Transgênicos Fluorescentes.

Um sistema de informação online foi desenvolvido a fim de se armazenar e organizar os dados relativos às proteínas fluorecentes. Um algoritmo específico para preparação e formatação do conjunto de dados utilizado na construção dos classificadores também foi desenvolvido e incluído no sistema.

Através da avaliação de métricas de desempenho quantitativas e características qualitativas dos três classificadores baseados em técnicas distintas, foi realizada uma comparação tendo em mente sua utilização no projeto Peixes Transgênicos Fluorescentes.

Os resultados obtidos, já analisados e discutidos, indicaram que:

- classificadores se mostraram importantes ferramentas de apoio à decisão e obtenção de conhecimento no contexto do projeto Peixes Transgênicos Fluorescentes;
- os classificadores baseados em Árvores de Decisão e Máquinas de Vetores de Suporte apresentaram desempenho estatisticamente superior ao classificador baseado em Redes Neurais Artificiais;
- os classificadores baseados em Árvores de Decisão e Máquinas de Vetores de Suporte apresentaram desempenho estatisticamente similares entre si;
- o classificador baseado em Árvores de Decisão apresentou mais capacidades e características alinhadas às necessidades do projeto Peixes Transgênicos Fluorescentesem comparação aos classificadores baseados em Máquinas de Vetores de Suporte e Redes Neurais Artificiais;
- o classificador baseado em Árvores de Decisão se mostrou o mais adequado para o objetivo proposto, sendo o escolhido no processo de comparação realizado.

Como possíveis trabalhos futuros, pode-se apontar:

- A escolha de classificadores baseados em técnicas diferentes das utilizadas neste trabalho, capazes de lidar com dados categóricos e numéricos, para reaplicação do processo comparativo entre estes novos classificadores e o classificador baseado em Árvore de Decisão.

- A reaplicação do mesmo processo comparativo deste trabalho, com um incremento do conjunto de dados de treinamento dos classificadores, no intuito de aprimorar a forma de relacionar a classe de cor das proteínas fluorescentes com a estrutura tridimensional da molécula, acrescentando novos dados relativos às propriedades físico-químicas dos aminoácidos.

- A ampliação dos métodos de pré-processamento e transformação aplicados ao conjunto de dados com o intuito de melhorar o desempenho do classificador baseado em Árvores de Decisão.

REFERÊNCIAS

ALMEIDA, D. V. **Peixes Transgênicos Fluorescentes**: um novo campo para a Piscicultura Ornamental no Brasil. Edital Chamada Universal MCTI/CNPQ Num. 14/2014, 2014.

BATISTA, G. E. d. A. P. A. **Pré-processamento de dados em aprendizado de máquina supervisionado**. 2003. Tese de Doutorado — Universidade de São Paulo.

BENSUSAN, H. N. **Automatic bias learning**: an inquiry into the inductive basis of induction. 1999. Tese de Doutorado — University of Sussex.

BERMAN, H. M.; WESTBROOK, J.; FENG, Z.; GILLILAND, G.; BHAT, T.; WEISSIG, H.; SHINDYALOV, I. N.; BOURNE, P. E. The protein data bank. **Nucleic Acids Research**, [S.l.], v.28, n.1, p.235–242, 2000.

BISHOP, C. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2007.

BOUCKAERT, R. R.; FRANK, E.; HALL, M.; KIRKBY, R.; REUTEMANN, P.; SEEWALD, A.; SCUSE, D. **WEKA manual for version 3-7-12**. 2015.

BRAGA, L. P. V. **Introdução à Mineração de Dados**: Edição ampliada e revisada. 2.ed. [S.l.]: Editora E-papers, 2005.

BURGES, C. J. A tutorial on support vector machines for pattern recognition. **Data mining and knowledge discovery**, [S.l.], v.2, n.2, p.121–167, 1998.

BUSSAB, W. d. O.; MORETTIN, P. A. **Estatística básica**. [S.l.]: Saraiva, 2010.

CASIMIRO, A.; ASHIKAGA, F.; KURCHEVSKI, G.; ALMEIDA, F.; ORSI, M. Os impactos das introduções de espécies exóticas em sistemas aquáticos continentais. **Boletim da Sociedade Brasileira de Limnologia**, [S.l.], n.38, p.1–12, 2010.

CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. **ACM Transactions on Intelligent Systems and Technology**, [S.l.], v.2, p.27:1–27:27, 2011.

CHUDAKOV, D. M.; MATZ, M. V.; LUKYANOV, S.; LUKYANOV, K. A. Fluorescent proteins and their applications in imaging living cells and tissues. **Physiological Reviews**, [S.l.], v.90, n.3, p.1103–1163, 2010.

CHUQUIPIONDO, C. Alternativas de Producción de Peces Ornamentales en la Amazonía Peruana. **Revista de la Facultad de Medicina Veterinaria y de Zootecnia**, [S.l.], n.54, p.123–127, 2007.

DALL'OGLIO, P. **PHP - Programando com Orientação a Objetos**. [S.l.]: Novatec, 2009.

DUGGAN, I. C.; RIXON, C. A.; MACISAAC, H. J. Popularity and propagule pressure: determinants of introduction and establishment of aquarium fish. **Biological invasions**, [S.l.], v.8, n.2, p.377–382, 2006.

ELMASRI, R. E.; NAVATHE, S. B. **Fundamentals of Database Systems**. 4.ed. [S.l.]: Addison Wesley, 2003.

FARIAS, F. de. GFP: Uma ferramenta brilhante para a visualização da vida. **Revista Virtual de Química**, [S.l.], v.1, n.1, p.2–8, 2009.

FAYYAD, U. M.; IRANI, K. B. Multi-interval discretization of continuous valued attributes for classification learning. In: **Thirteenth International Joint Conference on Articial Intelligence**. [S.l.]: Morgan Kaufmann Publishers, 1993. v.2, p.1022–1027.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI Magazine**, [S.l.], v.17, n.3, p.37–54, 1996.

FIESLER, E. Neural Network Topologies. In: **The Handbook of Neural Computation**. [S.l.]: Oxford University Press, 1996. p.1–17.

FIGUEIREDO, M. A.; MARECO, E. A.; SILVA, M. D. P.; MARINS, L. F. Muscle-specific growth hormone receptor (GHR) overexpression induces hyperplasia but not hypertrophy in transgenic zebrafish. **Transgenic Research**, [S.l.], v.21, n.3, p.457–469, 2012.

FIGUEIREDO, M. d. A.; LANES, C. F. C.; ALMEIDA, D. V.; MARINS, L. F. Improving the production of transgenic fish germlines: in vivo evaluation of mosaicism in zebrafish (Danio rerio) using a green fluorescent protein (GFP) and growth hormone cDNA transgene co-injection strategy. **Genetics and Molecular Biology**, [S.l.], v.30, n.1, p.31–36, 2007.

FRANK, E.; HALL, M.; TRIGG, L.; HOLMES, G.; WITTEN, I. H. Data mining in bioinformatics using Weka. **Bioinformatics**, [S.l.], v.20, n.15, p.2479–2481, 2004.

FREITAS, A. A.; WIESER, D. C.; APWEILER, R. On the importance of comprehensible classification models for protein function prediction. **ACM Transactions on Computational Biology and Bioinformatics**, [S.l.], v.7, n.1, p.172–182, 2010.

FURTADO, J. F. R. **Piscicultura**: uma alternativa rentável. [S.l.]: Livraria e Editora Agropecuária, 1995.

GABARDO, A. C. **PHP e MVC com CodeIgniter**. [S.l.]: Novatec, 2012.

GOLDSCHMIDT, R.; PASSOS, E. **Data Mining**: um guia prático. [S.l.]: Campus, 2005.

GONG, Z.; WAN, H.; TAY, T. L.; WANG, H.; CHEN, M.; YAN, T. Development of transgenic fish for ornamental and bioreactor by strong expression of fluorescent proteins in the skeletal muscle. **Biochemical and biophysical research communications**, [S.l.], v.308, n.1, p.58–63, 2003.

GROSS, L. A.; BAIRD, G. S.; HOFFMAN, R. C.; BALDRIDGE, K. K.; TSIEN, R. Y. The structure of the chromophore within DsRed a red fluorescent protein from coral. **PNAS**, [S.l.], v.97, n.22, p.11990–11995, 2000.

HAN, J.; KAMBER, M.; PEI, J. **Data mining**: concepts and techniques. [S.l.]: Elsevier, 2011.

HANSON, R. M. Jmol–a paradigm shift in crystallographic visualization. **Journal of Applied Crystallography**, [S.l.], v.43, n.5, p.1250–1260, 2010.

HAYKIN, S. **Redes Neurais**: Princípios e Prática. [S.l.]: Bookman, 2001.

HINCHLIFFE, A. **Molecular modelling for beginners**. 2.ed. [S.l.]: Wiley, 2008.

HIXON, J.; RESHETNYAK, Y. K. Algorithm for the analysis of tryptophan fluorescence spectra and their correlation with protein structural parameters. **Algorithms**, [S.l.], v.2, n.3, p.1155–1176, 2009.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, [S.l.], v.2, n.5, p.359–366, 1989.

HUANG, L.-T.; WU, C.-C.; LAI, L.-F.; GROMIHA, M. M.; WANG, C.-S.; CHEN, Y.-R. Data mining application in biomedical informatics for probing into protein stability upon double mutation. **Applied Mathematics and Information Sciences**, [S.l.], v.8, n.1L, p.125–132, 2014.

JOACHIMS, T. Making large-scale support vector machine learning practical. In: **Advances in Kernel Methods - Support Vector Learning**. [S.l.]: MIT Press, 1998. p.41–56.

JUBRAN, A. J.; JUBRAN, L. M. P.; MAGALHÃES CIPPARRONE, F. A. de; ALMEIDA JÚNIOR, J. R. de. **Data Mining na Web**. In: I WORKCOMP-SUL,2004.

KELLEY, L. A.; MEZULIS, S.; YATES, C. M.; WASS, M. N.; STERNBERG, M. J. The Phyre2 web portal for protein modeling, prediction and analysis. **Nature Protocols**, [S.l.], v.10, n.6, p.845–858, 2015.

LABROU, N. E. Random mutagenesis methods for in vitro directed enzyme evolution. **Current Protein and Peptide Science**, [S.l.], v.11, n.1, p.91–100, 2010.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. **Biometrics**, [S.l.], v.33, n.1, p.159–174, 1977.

LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, [S.l.], v.14, n.2, p.43–67, 2007.

MAGALHÃES, A. L. d. Novos registros de peixes exóticos para o Estado de Minas Gerais, Brasil. **Revista Brasileira de Zoologia**, [S.l.], v.24, n.1, p.250–252, 2007.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, [S.l.], v.5, n.4, p.115–133, 1943.

MERSCHMANN, L. H. d. C. **Classificação probabilística baseada em análise de padrões**. 2007. Tese de Doutorado — Universidade Federal Fluminense.

MIIKKULAINEN, R. Topology of a Neural Network. In: **Encyclopedia of Machine Learning**. [S.l.]: Springer, 2010. p.988–989.

MILANI, A. **Construindo aplicações web com PHP e MySQL**. [S.l.]: Novatec, 2010.

MONARD, M.; BARANAUSKAS, J. Conceitos sobre Aprendizado de Máquinas. In: **Sistemas Inteligentes**: Fundamentos e Aplicações. 1.ed. [S.l.]: Manole, 2003. p.89–114.

NANTASENAMAT, C.; ISARANKURA-NA-AYUDHYA, C.; TANSILA, N.; NAENNA, T.; PRACHAYASITTIKUL, V. Prediction of GFP spectral properties using artificial neural network. **Journal of Computational Chemistry**, [S.l.], v.28, n.7, p.1275–1289, 2007.

NANTASENAMAT, C.; SRUNGBOONMEE, K.; JAMSAK, S.; TANSILA, N.; ISARANKURA-NA-AYUDHYA, C.; PRACHAYASITTIKUL, V. Quantitative structure–property relationship study of spectral properties of green fluorescent protein with support vector machine. **Chemometrics and Intelligent Laboratory Systems**, [S.l.], v.120, p.42–52, 2013.

OLENYCH, S. G.; CLAXTON, N. S.; OTTENBERG, G. K.; DAVIDSON, M. W. The fluorescent protein color palette. In: **Current Protocols in Cell Biology**. [S.l.]: Wiley Online Library, 2007. p.1–20.

PRATI, R. C. **Novas abordagens em aprendizado de máquina para a geraçao de regras, classes desbalanceadas e ordenaçao de casos**. 2006. Tese de Doutorado — Universidade de São Paulo.

QUINLAN, J. R. Induction of decision trees. **Machine Learning**, [S.l.], v.1, n.1, p.81–106, 1986.

QUINLAN, J. R. **C4. 5**: Programs for Machine Learning. [S.l.]: Morgan Kaufmann, 1993. v.1.

QUINLAN, J. R. Improved use of continuous attributes in C4. 5. **Journal of Artificial Intelligence Research**, [S.l.], v.4, p.77–90, 1996.

RASGUIDO, J.; ALBANEZ, J. Piscicultura em Minas Gerais. **Informativo Agropecuário**, [S.l.], v.21, p.32–37, 2000.

RESHETNYAK, Y. K.; KOSHEVNIK, Y.; BURSTEIN, E. A. Decomposition of protein tryptophan fluorescence spectra into log-normal components. III. Correlation between fluorescence and microenvironment parameters of individual tryptophan residues. **Biophysical Journal**, [S.l.], v.81, n.3, p.1735–1758, 2001.

RIBEIRO, F. d. A. S. **Policultivo de acará-bandeira e camarão marinho**. 2010. Tese de Doutorado — Universidade Federal Paulista.

RIBEIRO, F. d. A. S.; LIMA, M. T.; KOCHENBORGER, C. J. B. Panorama do mercado de organismos aquáticos ornamentais. **Boletim Associação Brasileira de Limnologia**, [S.l.], v.38, n.2, p.1–9, 2010.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological Review**, [S.l.], v.65, n.6, p.386–408, 1958.

RUMELHART, D.; HINTON, G.; WILLIAMS, R. Learning representations by back-propagating errors. **Nature**, [S.l.], v.323, p.533–536, 1986.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence**: A Modern Approach. 2.ed. [S.l.]: Prentice Hall, 2003.

SCHÖLKOPF, B.; SMOLA, A. J. **Learning with kernels**: support vector machines, regularization, optimization, and beyond. [S.l.]: MIT press, 2002.

SIMON, H. A. Why should machines learn? In: **Machine learning**. [S.l.]: Springer, 1983. p.25–37.

SOARES, J. d. A. **Pré-Processamento em Mineração de Dados**: Um Estudo Comparativo em Complementação. 2007. Tese de Doutorado — Universidade Federal do Rio de Janeiro.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introdução ao data mining**: mineração de dados. [S.l.]: Addison-Wesley, 2006.

TIMERGHAZIN, Q. K.; CARLSON, H. J.; LIANG, C.; CAMPBELL, R. E.; BROWN, A. Computational prediction of absorbance maxima for a structurally diverse series of engineered green fluorescent protein chromophores. **The Journal of Physical Chemistry B**, [S.l.], v.112, n.8, p.2533–2541, 2008.

TLUSTY, M. Small scale of production does not automatically mean small scale of impact. **OFI Journal**, [S.l.], v.46, p.6–9, 2004.

UTGOFF, P. E. Shift of bias for inductive concept learning. **Machine learning: An artificial intelligence approach**, [S.l.], v.2, p.107–148, 1986.

VAPNIK, V. N. **Statistical learning theory**. [S.l.]: Wiley, 1998.

VERLI, H. **Bioinformática da Biologia á flexibilidade molecular**. [S.l.]: Universidade Federal do Rio Grande do Sul, 2014.

VIEIRA, S. **Introdução à bioestatística**. [S.l.]: Elsevier, 2015.

WEBB, B.; SALI, A. Comparative protein structure modeling using Modeller. In: **Current Protocols in Bioinformatics**. [S.l.]: Wiley Online Library, 2014. p.1–32.

WITTEN, I. H.; FRANK, E. **Data Mining**: Practical machine learning tools and techniques. [S.l.]: Morgan Kaufmann, 2005.

WOLPERT, D. H. The lack of a priori distinctions between learning algorithms. **Neural Computation**, [S.l.], v.8, n.7, p.1341–1390, 1996.

YANG, F.; MOSS, L. G.; PHILLIPS, G. N. The molecular structure of green fluorescent protein. **Nature Biotechnology**, [S.l.], v.14, p.1246–1251, 1996.

YANG, J.; YAN, R.; ROY, A.; XU, D.; POISSON, J.; ZHANG, Y. The I-TASSER Suite: protein structure and function prediction. **Nature Methods**, [S.l.], v.12, n.1, p.7–8, 2015.

ZAKI, M.; BYSTROFF, C. **Protein structure prediction**. [S.l.]: Springer, 2008.

ZHANG, S.; ZHANG, C.; YANG, Q. Data preparation for data mining. **Applied Artificial Intelligence**, [S.l.], v.17, n.5-6, p.375–381, 2003.

ANEXO A ATRIBUTOS DO MODELO RELACIONAL

Listagem dos atributos pertencentes às relações presentes no esquema do modelo relacional ilustrado na Figura 16. PK (*Primary Key*, ou Chave Primária): é o campo identificador único de uma tupla na tabela. FK (*Foreign Key*, ou Chave Estrangeira): é o campo da tabela que identifica unicamente uma tupla em outra tabela.

Relação: *protein*		
Representa a proteína fluorescente e suas propriedades fluorescentes, além de campos informativos oriundos do arquivo PDB.		
Atributo	*Tipo*	*Definição*
protein_id *PK*	Inteiro Sequencial	Identificador da proteína no banco de dados
name	Texto	Nome completo da estrutura da proteína, quando oriunda do PDB
short_name	Texto	Nome simplificado ou apelido da proteína
pdb_code	Texto	Código de acesso da estrutura da proteína no PDB
resolution	Real	Resolução da cristalografia da estrutura da proteína no PDB
keywords	Texto	Palavras-chaves vinculadas a estrutura da proteína no PDB
classification	Texto	Classificação da estrutura da proteína no PDB
has_mutation	Booleano	Indica se a estrutura da proteína tem mutação na sequência de aminoácidos
header	Texto	Linha de cabeçalho do arquivo PDB, contendo informações para identificação e data de publicação da estrutura no PDB
excitation	Inteiro	Comprimento de onda de excitação da proteína
emission	Inteiro	Comprimento de onda de emissão da proteína
quantum_yield	Real	Rendimento quântico da proteína
extinction_coefficient	Real	Absorção de luz em um dado comprimento de onda por densidade de massa ou concentração molar
brightness	Real	Brilho da fluorescência
maturation	Real	Tempo necessário para a proteína fluorescente produzir metade do seu máximo de fluorescência
pka	Real	Valor de pH no qual o brilho da proteína fluorescente é igual a 50% do brilho máximo medido no pH ótimo
class_id *FK*	Inteiro	Identificador da classe de cor a qual pertence
organism_id *FK*	Inteiro	Identificador do organismo do qual origina-se
oligomerization_id *FK*	Inteiro	Identificador da oligomerização
annotation	Texto	Comentários do arquivo PDB
reference	Texto	Referências vinculadas a estrutura da proteína fluorescente no PDB

Relação: *chain*		
Representa as cadeias de aminoácidos que compõem as proteínas fluorescentes.		
Atributo	Tipo	Definição
chain_id *PK*	Inteiro Sequencial	Identificador da cadeia
chain	Texto	Nome da cadeia, representado pelas letras A, B, C, D, etc.
length	Inteiro	Comprimento da cadeia (número de aminoácidos)
protein_id *FK*	Inteiro	Identificador da proteína a qual pertence a cadeia

Relação: *sequence*		
Representa a sequência de aminoácidos que compõem as cadeias.		
Atributo	Tipo	Definição
sequence_id *PK*	Inteiro Sequencial	Identificador da sequência
sequence	Texto	Sequência de aminoácidos de uma cadeia em sua representação de uma letra
chain_id *FK*	Inteiro	Identificador da cadeia que é composta pela sequência de aminoácidos

Relação: *mutation*		
Representa as mutações de aminoácidos nas cadeias que compõem as proteínas fluorescentes.		
Atributo	Tipo	Definição
mutation_id *PK*	Inteiro Sequencial	Identificador da mutação
num_res	Inteiro	Número da posição na sequência de aminoácidos onde ocorre a mutação
code3_ori	Texto	Código de 3 letras do aminoácido original da posição
code3_mut	Texto	Código de 3 letras do novo aminoácido na posição
chain_id *FK*	Inteiro	Identificador da cadeia na qual existe a mutação na sequência de aminoácidos

Relação: *class*		
Representa as classes de cores das proteínas fluorescentes.		
Atributo	Tipo	Definição
class_id *PK*	Inteiro Sequencial	Identificador da classe de cor
name	Texto	Nome da classe de cor

Relação: *organism*		
Representa os organismos nos quais as proteínas fluorescentes se originam.		
Atributo	Tipo	Definição
organism_id *PK*	Inteiro Sequencial	Identificador do organismo
name	Texto	Nome do organismo
tax_id	Inteiro	Número de classificação do organismo

Relação: *oligomerization*		
Representa a classificação de acordo com o número de cadeias que compõem as proteínas fluorescentes.		
Atributo	Tipo	Definição
oligomerization_id *PK*	Inteiro Sequencial	Identificador da oligomerização
name	Texto	Nome da oligomerização
num	Inteiro	Número que representa a oligomerização

Relação: *atom*		
Representa os átomos existentes.		
Atributo	Tipo	Definição
atom_id *PK*	Inteiro Sequencial	Identificador do átomo
name	Texto	Nome do átomo

Relação: *residue*		
Representa os aminoácidos (ou resíduos de aminoácidos, por isso o nome) existentes.		
Atributo	Tipo	Definição
residue_id *PK*	Inteiro Sequencial	Identificador do aminoácido
name	Texto	Nome do aminoácido
code3	Texto	Código de 3 letras do aminoácido
code1	Texto	Código de 1 letra do aminoácido

Relação: *rel_chain_res*		
Representa a relação de quais aminoácidos estão presentes nas cadeias.		
Atributo	Tipo	Definição
rel_chain_res_id *PK*	Inteiro Sequencial	Identificador da relação
chain_id *FK*	Inteiro	Identificador da cadeia
residue_id *FK*	Inteiro	Identificador do aminoácido

Relação: *rel_res_atom*		
Representa a relação de quais átomos estão presentes nos aminoácidos das cadeias.		
Atributo	Tipo	Definição
rel_res_atom_id *PK*	Inteiro Sequencial	Identificador da relação
rel_chain_res_id *FK*	Inteiro	Identificador da relação entre cadeia e aminoácidos
atom_id *FK*	Inteiro	Identificador do átomo

Relação: *atom_chain*		
Representa a relação dos átomos e sua numeração, e dos aminoácidos e suas posições sequenciais nas cadeias.		
Atributo	Tipo	Definição
atom_chain_id *PK*	Inteiro Sequencial	Identificador da relação
chain_id *FK*	Inteiro	Identificador da cadeia
atom_id *FK*	Inteiro	Identificador do átomo
num_atom	Inteiro	Numeração do átomo na cadeia (em relação ao arquivo PDB)
residue_id *FK*	Inteiro	Identificador do aminoácido
num_res	Inteiro	Numeração do aminoácido na sequência da cadeia (em relação ao arquivo PDB)

Relação: *coord_atom_chain*		
Representa a relação dos átomos que formam os aminoácidos nas cadeias e sua posição espacial (x,y,z)		
Atributo	Tipo	Definição
coord_atom_chain_id *PK*	Inteiro Sequencial	Identificador da relação
atom_chain_id *FK*	Inteiro	Identificador da relação dos átomos, aminoácidos e cadeias
coord_x	Inteiro	Coordenada espacial do átomo no eixo X
coord_y	Inteiro	Coordenada espacial do átomo no eixo Y
coord_z	Inteiro	Coordenada espacial do átomo no eixo Z

Relação: *user*		
Representa os usuários registrados para acesso ao sistema.		
Atributo	Tipo	Definição
user_id *PK*	Inteiro Sequencial	Identificador do usuário
name	Texto	Nome do usuário
email	Texto	E-mail utilizado para login
password	Texto	Senha de acesso

ANEXO B CÓDIGOS DE ACESSO PDB

Listagem dos códigos de acesso ao *Protein Data Bank* (BERMAN et al., 2000) para as 109 estruturas tridimensionais utilizadas no processo de KDD deste trabalho:

1BFP	2FWQ	4ORN	3M24	2WSN	2WSO	2YDZ
4AR7	3ZTF	1CV7	2Q57	2YE0	2YE1	3I19
3LA1	4EN1	4RYS	4KGE	3SVN	3SVO	3SVU
3U8C	3IP2	1UIS	3BX9	3BXA	3PJ5	4EDO
4EDS	4OJ0	3NF0	4H3N	2QLG	4H3L	4H3M
1YZW	2QLH	2QLI	2Q6P	4KW8	2Y0G	4EUL
4KW4	1EMB	1Q4A	1Q4C	1Q4E	1Q73	1YHH
1YHI	2AWK	2DUG	2DUI	2G6E	2QZ0	3P28
4KW9	4P1Q	4ZF3	1YHG	2G2S	2G16	4OGS
2G6X	3ADF	3IR8	2WHU	2WHT	4Q7U	2H5O
4Q7R	4Q7T	2ZMU	3MGF	2ZMW	2H5R	2H5Q
2VAD	3NED	1G7K	3NEZ	1ZGQ	3M22	3T6H
4KF4	4JF9	3U0N	3U0M	3PJB	3CFA	3CFF
4ZIO	3V3D	1MYW	1HUY	3W1C	3W1D	1F0B
2YFP	3DQ3	3DQ4	3DQ5	3DQ6	3DQ8	3DQ9
3DQA	1YFP	4JGE	4HE4			

ANEXO C AMINOÁCIDOS

Listagem dos 20 aminoácidos naturais formadores das proteínas, contendo o nome oficial, o símbolo de 3 letras e a abreviatura utilizados (VERLI, 2014):

Aminoácido	*Símbolo*	*Abreviatura*
Alanina	ALA	A
Cisteína	CIS/CYS	C
Aspartato (Ácido aspartico)	ASP	D
Glutamato (Ácido glutâmico)	GLU	E
Fenilalanina	FEN/PHE	F
Glicina	GLI/GLY	G
Histidina	HIS	H
Isoleucina	ILE	I
Lisina	LIS/LYS	K
Leucina	LEU	L
Metionina	MET	M
Asparagina	ASN	N
Prolina	PRO	P
Glutamina (Glutamida)	GLN	Q
Arginina	ARG	R
Serina	SER	S
Treonina	TRE/THR	T
Valina	VAL	V
Triptofano (Triptofana)	TRP	W
Tirosina	TIR/TYR	Y