UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS GRADUAÇÃO EM INFORMÁTICA

MARCELO ALMEIDA SANTANA

Um Estudo Comparativo das Técnicas de Predição na identificação de Insucesso Acadêmico dos Estudantes durante Cursos de Programação Introdutória

Maceió
2015

Marcelo Almeida Santana

Um Estudo Comparativo das Técnicas de Predição na identificação de Insucesso Acadêmico dos Estudantes durante Cursos de Programação Introdutória

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal de Alagoas.

Orientador: Prof. Dr. Evandro de B. Costa

Coorientador: Prof. Dr. Baldoino Fonseca dos Santos Neto

Maceió
2015

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecário: Valter dos Santos Andrade

S232e Santana, Marcelo Almeida.
Um estudo comparativo das técnicas de predição na identificação de insucesso acadêmico dos estudantes durante cursos de programação introdutória / Marcelo Almeida Santana. – 2015.
72 f. : il.

Orientador: Evandro de B. Costa.
Coorientador: Baldoino Fonseca dos Santos Neto.
Dissertação (Mestrado em Informática) – Universidade Federal de Alagoas. Instituto de Computação. Programa de Pós-Graduação em Informática. Maceió, 2015.

Bibliografia: f. 63-68.
Apêndice: f. 69-72.

1. Programação – Disciplina. 2. Estudantes – Avaliação. 3. Informática – Estudo e ensino. 4.Técnicas de mineração de dados educacionais. 5. Técnicas de predição. I. Título.

CDU: 004:378.115

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Informática – PpgI
Instituto de Computação
Campus A. C. Simões BR 104-Norte Km 14 BL 12 Tabuleiro do Martins
Maceió/AL - Brasil CEP: 57.072-970 | Telefone: (082) 3214-1401

Membros da Comissão Julgadora da Dissertação de Mestrado de Marcelo Almeida Santana, intitulada: "*Um Estudo Comparativo das Técnicas de Predição na Identificação de Insucesso Acadêmico dos Estudantes Durante Cursos de Programação Introdutória*", apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas em 06 de novembro de 2015, às 10h00min, na Sala de Reuniões do Instituto de Computação da UFAL.

COMISSÃO JULGADORA

Prof. Dr. Evandro de Barros Costa
UFAL – Instituto de Computação
Orientador

Prof. Dr. Baldoino Fonseca dos Santos Neto
UFAL – Instituto de Computação
Orientador

Prof. Dr. Patrick Henrique da Silva Brito
UFAL – Instituto de Computação
Examinador

Prof. Dr. Ryan Shaun Joazeiro Baker
Columbia University
Examinador

Dedico este trabalho ao meu pai meu grande herói. Infelizmente não teve tempo para presenciar essa conquista mas, onde estiver com certeza estará orgulhoso.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela força nos momentos de dificuldades e, pela luz que iluminou e guiou meus passos direcionando-me no caminho da aprendizagem.

Agradeço a minha mãe Marlene de Almeida Santana, pela sua dedicação e amor e ao meu pai Olival Matias de Santana (*In Memoriam*), que sempre me mostrou por meio de atitudes o valor da humildade e do trabalho. O meu irmão que sempre esteve presente e disposto a me ajudar em qualquer momento.

Agradeço também aos meus orientadores Prof. Dr. Evandro de B. Costa e Prof. Dr. Baldoino Fonseca dos Santos Neto, não só pela valiosa orientação, mas também pela enorme paciência e compreensão. Aos membros da Banca Avaliadora (Prof. Dr. Patrick Henrique da Silva Brito e Prof. Dr. Ryan S. Baker), pelas contribuições valiosas durante a defesa do mestrado.

E por fim, um agradecimento à Universidade Federal de Alagoas (*Campus Arapiraca*), em especial a todos que fazem o Núcleo de Tecnologia da Informação (NTI), que de alguma forma, contribuíram para essa minha vitória.

RESUMO

As altas taxas de insucesso nas universidades nos cursos que contemplam a disciplina de programação introdutória na sua grade curricular têm alarmado e preocupado muitos educadores, pois o insucesso dos estudantes podem gerar prejuízos dos mais diversos tipos e interesses. Assim, há relevantes motivos para se tentar esclarecer eventuais fatores que afetam tal insucesso. Ainda neste contexto, um dos desafios importantes é o de identificar antecipadamente os estudantes propensos ao insucesso na disciplina de programação introdutória, assumindo-se um tempo hábil para permitir intervenção pedagógica eficaz. Deste modo, buscou-se neste trabalho um estudo em técnicas de mineração de dados educacionais, objetivando-se comparar a eficácia dos algoritmos de predição capazes de identificar, em tempo hábil para intervenção pedagógica, os estudantes propensos ao insucesso. Neste estudo, avaliou-se a eficácia de algoritmos de predição em duas fontes de dados diferentes e independentes, uma na modalidade de ensino presencial e a outra na modalidade de ensino a distância sobre as disciplinas de programação introdutória. Os resultados mostraram que as técnicas analisadas no estudo são eficazes na identificação dos estudantes propensos ao insucesso no início da disciplina. Além disso, mostrou-se também que após a realização das etapas de pré-processamento e ajustes nos parâmetros dos algoritmos, tais algoritmos analisados tiveram uma melhora em seus resultados. Ao fim do processo, o algoritmo máquina de vetor de suporte (SVM: Support Vector Machine) apresentou os melhores resultados, tanto na modalidade de ensino presencial quanto na modalidade a distância, alcançando uma taxa de *f-measure* de 83% e 92%, respectivamente.

Palavras-chaves: Disciplina de programação. Predição. Insucesso. Mineração de dados Educacionais.

ABSTRACT

The high failure rates of students in the introductory programming course within the universities worldwide have alarmed and worried many educators. Those rates can lead to losses of various types and interests. Thus, there are important reasons to try to clarify the main factors that possibly influency such failures. Furthermore, one of the major challenges is on how to early identify the students likely to fail in the introductory programming course, eventually allowing effective pedagogical interventions. Thus, in this study we aim to explore educational data mining techniques, in order to compare the effectiveness of prediction algorithms capable of identifyingvstudents likely to fail, in a timely manner suitable for pedagogical intervention. This study evaluated the efficacy of prediction algorithms in two different and independent data sources, one in the classroom teaching mode and the other in the distance education mode on the disciplines of introductory programming. The results showed that the techniques discussed in this study are effective in this task of prediction. In addition, it was shown also that after the completion of the pre-processing and adjustments to the parameters of the algorithms, such algorithms analyzed had an improvement in their results. At the end of the process, the Support Vector Machine (SVM) algorithm showed the best results, both in the classroom teaching mode as in the distance, reaching an *f-measure* rate of 83% and 92% respectively.

Keywords: Course programming. Prediction. Failure. Educational data mining.

LISTA DE ILUSTRAÇÕES

Figura 1 – Hierarquia do Aprendizado 17
Figura 2 – Os processos de KDD 19
Figura 3 – Conceito de Árvore de decisão 26
Figura 4 – Hiperplano com margem de separação estreita (A) e margem de separação larga (B) 27
Figura 5 – Resumo das funções de *Kernel* 28
Figura 6 – Estrutura de um neurônio artificial 29
Figura 7 – Rede Neural Artificial de múltiplas camadas 30
Figura 8 – Transformação em Kettle 42
Figura 9 – Weka (GUI Chooser) 42
Figura 10 – Etapas do pré-processamento dos dados 46
Figura 11 – Pesos e atributos do ensino a distância 49
Figura 12 – Pesos e atributos do ensino presencial 49
Figura 13 – Evolução dos resultados semanalmente ensino a distância 53
Figura 14 – Evolução dos resultados semanalmente ensino presencial 54
Figura 15 – Eficácia dos algoritmos de predição do ensino a distância (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte) 55
Figura 16 – Eficácia dos algoritmos de predição do ensino presencial (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte) 55
Figura 17 – Comparativo dos resultados da eficácia dos algoritmos de predição do ensino a distância sem pré-processamento (SP) e com pré-processamento (P) 57
Figura 18 – Comparativo dos resultados da eficácia dos algoritmos de predição do ensino presencial sem pré-processamento (SP) e com pré-processamento (P) 57
Figura 19 – Comparativo dos resultados da eficácia dos algoritmos de predição do ensino a distância sem ajustes finos (SAF) e com ajustes finos (AF) . . 58
Figura 20 – Comparativo dos resultados da eficácia dos algoritmos de predição do ensino presencial sem ajustes finos (SAF) e com ajustes finos (AF) . . 58
Figura 21 – Eficácia do algoritmo de predição KNN após pré-processamento e ajustes finos do ensino a distância (KNN - K vizinhos mais próximos) . . . 71
Figura 22 – Eficácia do algoritmo de predição KNN após pré-processamento e ajustes finos do ensino presencial (KNN - K vizinhos mais próximos) 71

Figura 23 – Eficácia dos algoritmos de predição de caixa branca após pré-processamento do ensino a distância 72
Figura 24 – Eficácia dos algoritmos de predição de caixa branca após pré-processamento do ensino presencial 72

LISTA DE TABELAS

Tabela 1 – Atributos Selecionados na modalidade de ensino a distância 48
Tabela 2 – Atributos Selecionados na modalidade de ensino presencial 48
Tabela 3 – Resultados detalhados dos algoritmos de predição do ensino a distância (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte) 69
Tabela 4 – Resultados detalhados dos algoritmos de predição do ensino a distância após o pré-processamento dos dados (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte) . 69
Tabela 5 – Resultados detalhados dos algoritmos de predição do ensino a distância após ajustes finos (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte) 69
Tabela 6 – Resultados detalhados dos algoritmos de predição do ensino presencial (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte) 70
Tabela 7 – Resultados detalhados dos algoritmos de predição do ensino presencial após pre-processamento dos dados (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte) . . . 70
Tabela 8 – Resultados detalhados dos algoritmos de predição do ensino presencial após ajustes finos (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte) 70

SUMÁRIO

1 INTRODUÇÃO 11
1.1 Motivação 11
1.2 Problemática 12
1.3 Questões de Pesquisa 14
1.4 Objetivos 15
1.4.1 Objetivo Geral 15
1.4.2 Objetivos Específicos 15
1.5 Relevância 16
1.6 Estrutura da Dissertação 16

2 FUNDAMENTAÇÃO TEÓRICA 17
2.1 Aprendizagem de Máquina 17
2.1.1 Aprendizagem Supervisionada 18
2.2 Descoberta de Conhecimento em Banco de Dados (KDD) 18
2.2.1 Seleção 19
2.2.2 Pré-processamento 19
2.2.3 Transformação 20
2.2.4 Mineração 22
2.2.5 Interpretação dos resultados 22
2.3 Mineração de dados Educacionais 22
2.3.1 Conceito de Mineração de dados 22
2.3.2 Mineração de Dados Educacionais 22
2.3.3 Tarefas de Mineração de Dados 23
2.3.4 Técnicas de Mineração de dados 24
2.3.4.1 Classificação Bayesiana 24
2.3.4.2 Árvore de Decisão 25
2.3.4.3 Máquinas de Vetores de Suporte (SVM) 26
2.3.4.4 Redes Neurais Artificiais 28

3 TRABALHOS RELACIONADOS 31
3.1 Métodos de predição de Insucesso 31
3.2 Pré-processamento dos dados 33
3.3 Ajustes finos nos algoritmos 34

4 MÉTODO 36
4.1 Seleção dos Dados 36

4.1.1 Ensino a distância . . . 37
4.1.2 Ensino presencial . . . 37
4.2 Seleção dos Algoritmos . . . 38
4.2.1 Validação . . . 39
4.2.2 Métrica . . . 39
4.3 Instrumentação . . . 40
4.3.1 Pentaho . . . 40
4.3.2 Weka . . . 41

5 REALIZAÇÃO DO EXPERIMENTO . . . 44
5.1 Pré-processamento dos dados . . . 44
5.1.1 Integração . . . 44
5.1.2 Limpeza . . . 45
5.1.3 Transformação . . . 45
5.1.4 Seleção dos atributos . . . 46
5.1.5 Balanceamento dos dados . . . 49
5.2 Execuções dos Algoritmos . . . 50
5.2.1 Ajustes finos nos Algoritmos . . . 51

6 RESULTADOS E DISCUSSÕES . . . 53
6.1 Discussões dos Resultados . . . 53
6.2 Resposta as questões de pesquisa . . . 54
6.2.1 Qual a eficácia dos algoritmos de predição para identificar estudantes propensos ao insucesso? . . . 54
6.2.2 As etapas de pré-processamento de dados são realmente importantes e capazes de aumentar a eficácia dos algoritmos de predição? . . . 56
6.2.3 A fase de ajustes finos dos algoritmos será capaz de aumentar ainda mais a eficácia dos algoritmos de predição? . . . 58
6.2.4 Depois de realizar o pré-processamento dos dados e os ajustes finos nos algoritmos de predição, quais das técnicas são mais eficazes na identificação dos estudantes propensos ao insucesso? . . . 59
6.3 Limitações e Ameaças a validade do estudo . . . 59

7 CONSIDERAÇÕES FINAIS . . . 61
7.1 Conclusão . . . 61

REFERÊNCIAS . . . 63
APÊNDICE A – RESULTADOS DETALHADOS . . . 69
A.1 Resultados . . . 69

1 INTRODUÇÃO

1.1 Motivação

A disciplina de programação introdutória faz parte de todos os cursos em nível universitário das áreas de Ciências da Computação, Engenharia da Computação, Sistemas da Informação, Análise de sistemas e áreas afins. No entanto, é uma constante já há muitos anos, o relato de dificuldades no processo de aprendizagem nas disciplinas introdutórias de programação(HANKS et al., 2004), (TAN; TING; LING, 2009) e também a queixa de que essas dificuldades aumentam os índices de insucesso nestas disciplinas (IEPSEN; BERCHT; REATEGUI, 2013), (WATSON; LI, 2014).

Diversos estudos têm mostrado altas taxas de insucesso nas universidades ao longo dos anos em cursos que tem a disciplina introdutória de programação na sua grade curricular (WATSON; LI, 2014), (BENNEDSEN; CASPERSEN, 2007). Essas altas taxas de insucesso, em cursos da área de computação e afins, têm alarmado e preocupado muitos educadores, pois o insucesso dos estudantes podem gerar prejuízos dos mais diversos tipos e interesses.

Para tentar esclarecer e abordar este problema, uma das vias importantes é a análise de dados, com a utilização de ambientes de aprendizagem baseado na web, onde são geradas grandes quantidades de dados educacionais, provenientes de interações e dos registros contínuos de dados dos professores, estudantes, gestores e demais atores dos sistemas educacionais. Esse grande volume de dados pode conter muita informação relevante que pode ser usada para compreender melhor os estudantes e educadores. Assim, uma solução interessante para extração de informações relevantes da grande quantidade de dados educacionais geradas é com a utilização técnica de Mineração de Dados Educacional (EDM)[1] (RYAN; SEIJI; CARVALHO, 2011; ROMERO; VENTURA, 2010).

Neste sentido, o uso de tais técnicas possibilita examinar grandes quantidades de dados, objetivando identificar padrões eficazes que possam apontar com precisão os estudantes que estão propensos ao insucesso em um estágio inicial do curso de programação introdutória, permitindo aos educadores realizar, em tempo hábil, as intervenções pedagógicas necessárias para garantir o sucesso dos estudantes na disciplina. Com isso, pretende-se contribuir para diminuir os altos índices de insucesso em relação aos cursos que tem a disciplina de programação introdutória na sua grade curricular.

O conceito de insucesso acadêmico, aqui utilizado, significa o não atingimento de metas (fim de um ano letivo ou de um ciclo de estudos) pelos estudantes dentro dos limites temporais estabelecidos, em que o principal indicador se traduz pelas taxas de reprovação, com consequências diretas na repetição de disciplinas ou até mesmo abandono da

[1] do inglês *Educational Data Mining*

universidade (MARTINS, 2013). É importante destacar que o insucesso trata-se de um problema multifatorial onde fatores pessoais, familiar, social, econômicos podem influenciar no insucesso dos estudantes. No entanto, aspectos externos ao ambiente não são contemplados no estudo como por exemplo: aspectos afetivos, sociais e econômicos.

O presente trabalho se enquadra neste contexto, estando circunscrito no cenário de cursos de ensino de programação introdutórios nas modalidades de ensino presencial e a distância, utilizando duas fontes de dados diferentes e independentes disponibilizadas pela Universidade Federal de Alagoas (UFAL). Deste modo, investe-se num estudo sobre insucesso de estudantes baseando-se apenas nos dados acadêmicos que são gerados nestes sistemas.

1.2 Problemática

O abandono do estudante sem a finalização dos seus estudos tem gerado prejuízos a todos os envolvidos no processo educativo (LOBO, 2013), pois perde o aluno, seus professores, a instituição de ensino, o sistema de educação e toda a sociedade. Essa perda coletiva ocorre na medida em que esses estudantes terão maiores dificuldades de atingir seus objetivos pessoais e porque, no geral, existirá um número menor de pessoas com formação completa do que se poderia, tendo mais dificuldades para que cumpram seu papel na sociedade com eficiência e competência (LOBO, 2013).

Neste contexto um problema relevante é a capacidade de conseguir identificar em tempo hábil para intervenção pedagógica os estudantes propensos ao insucesso em cursos que têm em sua grade curricular a disciplina introdutória de programação. A fim de lidar com este problema, alguns trabalhos (ROMERO et al., 2013), (BAYER et al., 2012), (ARORA; SINGHAL; BANSAL, 2014), (aES; CRUZ; aO, 2014), (MARTINHO; NUNES; MINUSSI, 2013) se propuseram a analisar o uso de técnicas de mineração de dados educacionais para prever o insucesso dos estudantes.

No ensino presencial, que não utiliza nenhum tipo de sistema de informação, cabe ao professor identificar os estudantes com possíveis chances de insucesso, seja através da utilização de instrumentos de verificação da aprendizagem ou seja pelo próprio comportamento do aluno em sala de aula. Contudo, ao fazer uso de tecnologias para suporte ao ensino, essa tarefa se torna ainda mais difícil. Isto acontece devido ao fato das informações e registros sobre as ações e comportamento dos estudantes, geralmente, ficarem dispersas, dificultando a tarefa do professor em sumarizar para compreender o status geral do aluno no curso. Tornando necessário o desenvolvimento de ferramentas capazes de identificar perfis de estudantes propensos ou não ao insucesso.

Por esta razão, muitas universidades estão levando em conta em seus planos estratégicos um objetivo primordial: reduzir a taxa de insucesso dos estudantes, pois com a redução das taxas de insucesso acarretará na redução das perdas sociais, de recursos e de tempo de todos os envolvidos no processo de ensino.

Além dos problemas expostos acima, destacamos a seguir alguns problemas técnicos inerentes a este estudo.

O primeiro problema técnico está relacionado ao entendimento dos dados. Dados do mundo real são normalmente ruidosos, enorme em termos de volume, e podem ter origem a partir de uma mistura de fontes heterogêneas (HAN; KAMBER; PEI, 2011). Um bom conhecimento sobre os dados é útil para a próxima grande tarefa do processo de mineração de dados que corresponde a etapa de pré-processamento de dados (HAN; KAMBER; PEI, 2011).

Perguntas como: Quais são os tipos de atributos ou campos que compõem os seus dados? Que tipo de valores cada atributo tem? Quais são os atributos que tem valores discretos ou contínuos? O que os dados parecem? Como são distribuídos os valores? Existem maneiras que podemos visualizar os dados para obter um melhor sentido a cerca de tudo isso? Podemos identificar os valores atípicos? Podemos medir a similaridade de alguns objetos de dados em relação aos outros? Conseguir tal visão sobre os dados pode ajudar com a análise posterior.

Uma vez que os dados foram entendidos, o próximo problema técnico que precisa ser solucionado é com relação a preparação dos dados. Reunir, limpar, transformar, organizar e deixar os dados no formato adequado para mineração é um processo não trivial (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Geralmente este processo pode ser auxiliado por técnicas de descoberta de conhecimento em base de dados KDD[2] na etapa de pré-processamento dos dados (HAN; KAMBER; PEI, 2011).

Para conseguir realizar a etapa de pré-processamento de dados e a aplicação dos algoritmos de mineração de dados de forma eficiente é preciso escolher as técnicas e ferramentas de mineração de dados necessárias. Segundo (GIBERT; MARRE; CODINA, 2010), a escolha da técnica de mineração de dados depende do problema de negócio a ser solucionado e das características dos dados disponíveis para análise. Enquanto que, na escolha da ferramenta de mineração de dados deve-se levar em consideração vários parâmetros, tais como: características gerais da ferramenta, conexão a bancos de dados, critérios de desempenho computacional, critérios de funcionalidade, usabilidade e custo.

Um outro problema importante a ser discutido é como conseguir identificar quais atributos, realmente, estão relacionados com o insucesso dos estudantes e que irão potencialmente influenciar ou contribuir significativamente para identificação precoce do perfil dos estudantes propícios ao insucesso, dentro do universo de dados gerados e disponíveis nos sistemas utilizados pela Universidade.

Conseguir distinguir qual seria o melhor algoritmo que nos responda com qualidade em termos de precisão quais estudantes de uma determinada turma estariam propícios ou não ao insucesso, considerando apenas dados obtidos no início do curso, é outro grande desafio a ser considerado. Dado que, cada algoritmo funciona e produz resultados diferentes e

[2] do inglês *Knowledge Discovery in Databases*

alguns podem produzir mais de um tipo de resultado.

Portanto, prever o insucesso dos estudantes em tempo hábil para uma intervenção pedagógica não é uma tarefa fácil, não só porque o insucesso trata-se de um problema multifatorial e os dados do mundo real geralmente tendem a ser sujos, incompletos e inconsistentes, mas também porque o sucesso do preditor dependerá da escolha mais adequada das técnicas de EDM utilizadas.

1.3 Questões de Pesquisa

Diante do cenário exporto acima, surgem algumas questões de pesquisa que pretendemos responder com este trabalho.

Questão 1. Qual a eficácia dos algoritmos de predição para identificar estudantes propensos ao insucesso?

A intenção com a *1ª Questão* é avaliar a eficácia dos algoritmos de predição que têm sido utilizados por abordagens existentes para identificar os estudantes propensos ao insucesso. Para responder esta questão, os algoritmos de predição serão aplicados nos dois conjuntos de dados distintos e, em seguida, a medida de *F-measure* será utilizada para avaliar a eficácia de tais técnicas.

Questão 2. As etapas de pré-processamento de dados são realmente capazes de aumentar a eficácia dos algoritmos de predição?

A *2ª Questão* tem como objetivo analisar se a eficácia destes algoritmos aumentam após a realização do pré-processamento de dados. Para responder a *2ª Questão*, será realizado o pré-processamento nos dois conjuntos de dados utilizados neste trabalho, em seguida, serão aplicados os algoritmos de predição sobre os conjuntos de dados, avaliando a eficácia dessas técnicas e comparando esses resultados com a eficácia obtida executando as mesmas técnicas sobre o conjunto de dados, sem o pré-processamento.

Questão 3. A fase de ajustes finos dos algoritmos será capaz de aumentar ainda mais a eficácia dos algoritmos de predição?

A *3ª Questão* tem como objetivo analisar se a eficácia dos algoritmos de predição pode aumentar ainda mais após a realização de ajustes finos nos algoritmos utilizados. Para responder a *3ª Questão*, serão realizados os ajustes finos nos algoritmo utilizados, em seguida, serão executados os algoritmos de predição no conjunto de dados pré-processados, que avaliou a eficácia dos algoritmos em seguida comparar a sua eficácia com os resultados obtido pela realização dos algoritmos de predição, sem o ajuste fino.

Questão 4. Depois de realizar o pré-processamento dos dados e os ajustes finos nos algoritmos de predição, quais das técnicas são mais eficazes na identificação dos estudantes propensos ao insucesso?

A *4º Questão* visa encontrar a técnica mais eficaz para a identificação precoce de estudantes susceptíveis ao insucesso. Para responder a essa questão, será analisada e comparada a eficácia das técnicas de algoritmos de predição após a realização das fases de pré-processamento de dados e de ajustes finos nos algoritmos.

1.4 Objetivos

Tendo em vista as motivações apresentadas anteriormente, este trabalho tem como objetivo comparar a eficácia dos algoritmos de predição capazes de identificar precocemente os estudantes propensos ao insucesso utilizando somente dados extraídos dos sistemas utilizados pela universidade, em cursos na modalidade de ensino a distância ou presencial, especificamente por meio de técnicas de mineração de dados educacionais. Tal análise prioriza fatores relacionados ao sucesso ou insucesso dos estudantes nas disciplinas iniciais do curso de programação introdutória, medidos através das interações, participações e notas obtidos na disciplina. Neste sentido, questões afetivas, de interface e aspectos externos ao ambiente não são contemplados no estudo.

1.4.1 Objetivo Geral

Este trabalho tem como objetivo geral investigar quais são as técnicas de predição mais eficazes na identificação dos estudantes propensos ao insucesso, o mais breve possível, a fim de aumentar as chances de uma intervenção pedagógica, nas disciplinas iniciais de programação.

1.4.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

1. Avaliar a eficácia das técnicas de predição que têm sido utilizadas por abordagens existentes para identificar os estudantes propensos ao insucesso;

2. Identificar se as atividades de pré-processamento dos dados, realmente contribuem e influenciam de forma positiva nos resultados das técnicas de predição analisadas;

3. Analisar se o procedimento de ajustes finos é capaz de aumentar ainda mais a eficácia das técnicas de predição analisadas;

4. Identificar qual técnica de predição, é mais apropriada para ser aplicada na identificação, em tempo hábil, dos estudantes propensos ao insucesso nas disciplinas de

programação introdutórias, estando restrito apenas aos dados acadêmicos disponíveis nos sistemas utilizados pela universidade.

1.5 Relevância

Considerando que a qualidade da educação não está somente relacionada ao aumento do número de estudantes matriculados, mas também ao acompanhamento adequado do aprendizado desses estudantes (ABED, 2014) e aos processos decisórios adotados pelos educadores. É interessante a intervenção da tecnologia através da utilização de técnicas de mineração de dados educacionais no sentido de apoiar os educadores no processo de tomada de decisão.

Este estudo pretende dar uma contribuição ao problema do insucesso fazendo um comparativo entre os algoritmos de predição que têm sido utilizados por abordagens existentes, considerando os ambientes nos quais eles interagem, tais como Ambientes Virtuais de Aprendizagem, Sistemas Tutores Inteligentes, entre outros. Atingindo o objetivo da proposta, será possível ter subsídios para identificar a técnica mais apropriada para ser aplicada na identificação, em tempo hábil, dos estudantes propensos ao insucesso nas disciplinas de programação introdutórias estando restrito apenas aos dados acadêmicos, contribuindo, assim, para que educadores possam usar essas informações para auxiliar nas suas decisões pedagógicas.

Além de identificar os estudantes que estão propensos ao insucesso, este estudo também contribuirá no que se refere às etapas de pré-processamento de dados e ajustes finos nos parâmetros dos algoritmos de predição, pois com o desenvolvimento destas etapas através de ferramenta específica será possível tornar esse processo menos complexo e mais transparente para pesquisadores que trabalham na área de EDM.

1.6 Estrutura da Dissertação

Este estudo foi dividido em 7 capítulos. No Capítulo 1 é apresentada uma visão geral da dissertação que se segue, descrevendo a motivação, contribuição, objetivos gerais e específicos do presente documento. O Capítulo 2 apresenta os conceitos relevantes para o trabalho como Aprendizagem de máquina, Descoberta de conhecimento em banco de dados e Mineração de dados Educacionais. O Capítulo 3 apresenta os trabalhos relacionados para cada uma das áreas do trabalho. No Capítulo 4 é apresentada a metodologia para o desenvolvimento dos experimentos. No Capítulo 5 é descrito com detalhes como foi realizado o estudo comparativo. No Capítulo 6 são apresentados e comparados os resultados obtidos através dos algoritmos de predição. E finalmente no Capítulo 7 são apresentas as considerações finais e direcionamentos para trabalhos futuros.

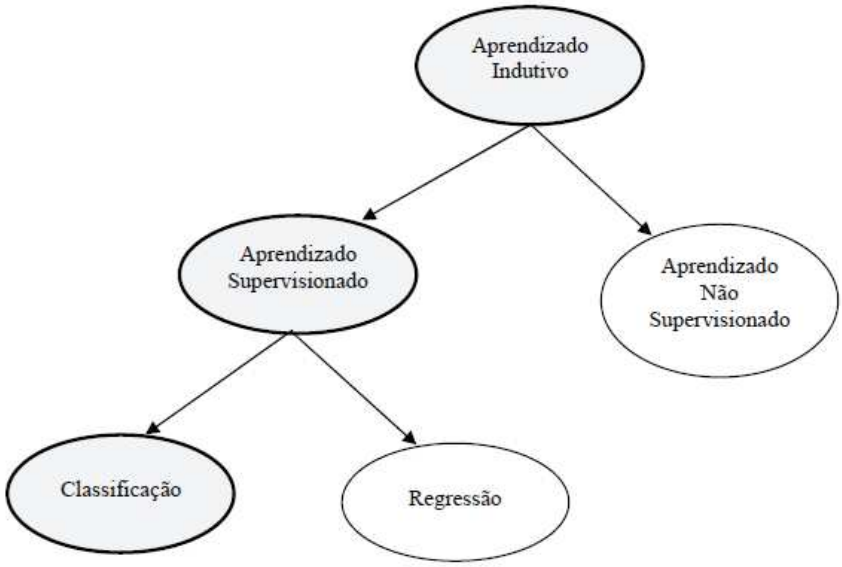
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos que servem como base para o entendimento dos elementos conceituais relevantes para a compreensão da proposta de trabalho. Abordando assuntos como aprendizagem de máquina, descoberta de conhecimento em banco de dados e mineração de dados educacionais.

2.1 Aprendizagem de Máquina

Aprendizagem de máquina é uma subárea da Inteligência Artificial, cujo objetivo consiste em obter resultados genéricos a partir de um conjunto particular de dados (MITCHELL, 2006). Para tal, as técnicas de aprendizagem de máquina empregam um princípio de inferência denominado indução, no qual se obtém conclusões genéricas a partir de um conjunto particular de exemplos. O aprendizado indutivo é realizado a partir de raciocínios sobre exemplos providos por um procedimento externo ao sistema de aprendizado. Portanto, quanto maior o número de exemplos relevantes, melhores serão os resultados para um novo conjunto de dados. O aprendizado indutivo pode ser dividido em dois tipos fundamentais: supervisionado e não-supervisionado (RUSSELL; NORVIG, 2002). Na Figura 1 é exibida a hierarquia associada ao aprendizado indutivo. Nesse trabalho devido aos tipos dos dados, a técnicas mais apropriada é a de aprendizagem supervisionada.

Figura 1 – Hierarquia do Aprendizado



Fonte: (MONARD; BARANAUSKAS, 2003)

2.1.1 Aprendizagem Supervisionada

No aprendizado supervisionado há a figura do professor externo, o qual apresenta o conhecimento do ambiente por conjuntos de exemplos na forma: entrada, saída desejada (HAYKIN, 2000). O objetivo é induzir conceitos a partir de exemplos que estão pré-classificados, seu funcionamento consiste na inserção de um "Supervisor" no ciclo de aprendizado, responsável por dizer ao modelo se suas previsões estão corretas ou não. Uma das formas de se implementar tal decisor se baseia na construção de um conjunto de dados denominados conjunto de treinamento, no qual, amostras são retiradas do sistema e previamente classificadas pelo supervisor. Assim, durante seu processo de aprendizado, a máquina pode verificar se suas respostas estão coerentes ou não com o esperado, ajustando-se de forma a minimizar o erro para tal conjunto. Tal abordagem pode ser bastante útil para problemas de classificação e regressão, se as classes possuírem valores discretos, o problema é categorizado como classificação. Caso as classes possuam valores contínuos, o problema é categorizado como regressão. Portanto, devido ao tipo de dados essa foi a técnicas de aprendizagem utilizada no trabalho.

2.2 Descoberta de Conhecimento em Banco de Dados (KDD)

O termo Descoberta de Conhecimento em Banco de dados (do inglês, *Knowledge Discovery in Databases* (KDD)) foi formalizado por Fayyad, em 1989 (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996), como um processo, não trivial, de extração de informações implícitas, previamente desconhecidas e potencialmente úteis, a partir dos dados armazenados em grandes conjuntos de dados. Não trivial por que, normalmente, encontra-se complexidade no decorrer da execução do processo. Interativo, significa que o usuário poderá optar pela retomada em qualquer uma das etapas do processo. Iterativo significa que o processo pode repetir diversas vezes para se chegar a um resultado e, a cada vez, gera um resultado parcial, o qual será usado na vez seguinte.

Entretanto, os termos KDD e Mineração de dados (do inglês, *Data Mining* (DM)) têm sido muitas vezes utilizados como sinônimos e muitas outras vezes como conceitos distintos, não existindo um consenso entre os autores. (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) entendem que Mineração de dados é uma etapa do processo de KDD e consiste de algoritmos que produzem, dada um eficiência computacional aceitável, uma enumeração de padrões em um conjunto de dados, consideram a mineração de dados, como um componente do processo de KDD, trata principalmente dos meios pelos quais padrões são extraídos de um conjunto de dados.

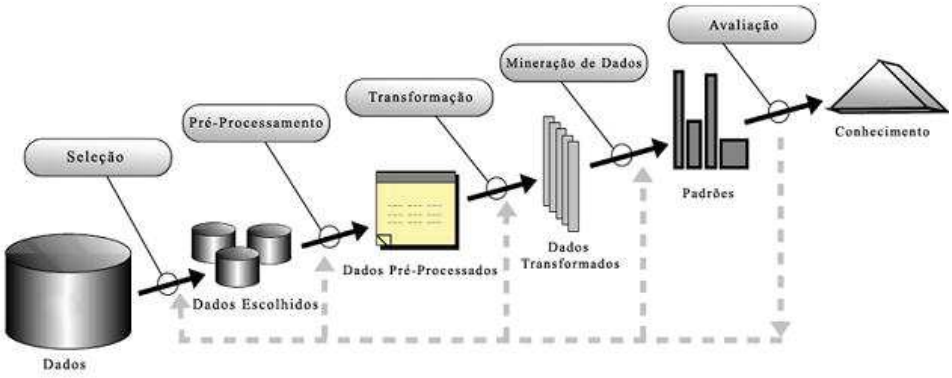
O entendimento adotado por (ADRIAANS; ZANTINGE, 1997), é de que o KDD e a mineração de dados são conceitos distintos. Segundo os autores, durante a Primeira Conferência Internacional de KDD, realizada em Montreal em 1995, foi proposto que o termo KDD deveria ser empregado para descrever inteiramente o processo de extração de

conhecimento a partir de dados. Ainda, o termo mineração de dados passou a ser utilizado exclusivamente para a etapa de descoberta durante um processo de KDD. (BIGUS, 1996) traz que a mineração de dados, também conhecida como descoberta de conhecimento, é a descoberta eficiente de informação valiosa e não óbvia a partir de um conjunto grande de dados. Ou seja, até o momento não se tem um consenso da definição dos termos Mineração de dados e KDD. No entanto, todos concordam que o processo de mineração deve ser iterativo, interativo e divido em fases.

Portanto, não existindo um consenso entre os autores pesquisados, o presente trabalho adotará o que parece ser o entendimento mais comum, isto é, a mineração de dados como sendo uma etapa do processo de KDD, a qual são aplicados algoritmos para reconhecimento de padrões nos dados.

As etapas definidas por (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) para o processo de descoberta de conhecimento em banco de dados são apresentadas na Figura 2.

Figura 2 – Os processos de KDD



Fonte: (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996)

2.2.1 Seleção

A etapa de seleção dos dados é a primeira no processo de descobrimento do conhecimento, possui impacto significativo sobre a qualidade do resultado final, uma vez que nesta fase é escolhido o conjunto de dados contendo todas as possíveis variáveis (atributos) e registros (casos) que farão parte da análise. O processo de seleção é bastante complexo, uma vez que os dados podem vir de uma série de fontes diferentes e podem possuir os mais diversos formatos.

2.2.2 Pré-processamento

O Pré-processamento é uma parte crucial no processo de KDD, pois a qualidade dos dados vai determinar a eficiência dos algoritmos de mineração. Nesta etapa, deverão ser realizadas tarefas que eliminem dados inválidos, redundantes e inconsistentes, recu-

perem dados incompletos e avaliem possíveis dados discrepantes ao conjunto (*outliers*). Segundo (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996), consistem principalmente em:

Limpeza dos dados: Frequentemente, os dados são encontrados com diversas inconsistências: registros incompletos, valores errados e dados inconsistentes. A etapa de limpeza dos dados visa eliminar estes problemas de modo que eles não influam no resultado dos algoritmos usados. As técnicas usadas, nesta etapa, vão desde a remoção do registro com problemas, passando pela atribuição de valores padrões, até a aplicação de técnicas de agrupamento para auxiliar na descoberta dos melhores valores. Devido ao grande esforço exigido nesta etapa, (HAN; KAMBER; PEI, 2011) propõem o uso de um processo específico para a limpeza dos dados.

Integração dos dados: É comum se obter os dados a serem minerados de diversas fontes: banco de dados, arquivos textos, planilhas, *data warehouses*, vídeos, imagens, entre outras. Surge então, a necessidade da integração destes dados de forma a termos um repositório único e consistente. Para isto, é necessária uma análise aprofundada dos dados observando redundâncias, dependências entre as variáveis e valores conflitantes (categorias diferentes para os mesmos valores, chaves divergentes, regras diferentes para os mesmos dados, entre outros).

2.2.3 Transformação

Não existe um critério único para transformação dos dados e diversas técnicas podem ser usadas de acordo com os objetivos pretendidos (HAN; KAMBER; PEI, 2011). Algumas das técnicas empregadas nesta etapa são: suavização (remove valores errados dos dados), agrupamento (agrupa valores em faixas sumarizadas), generalização (converte valores muito específicos para valores mais genéricos), normalização (colocar as variáveis em uma mesma escala), criação de novos atributos (gerados a partir de outros já existentes), Redução e balanceamento dos dados.

Seleção de atributos: O volume de dados usado na mineração costuma ser alto. Em alguns casos, este volume é tão grande que torna o processo de análise dos dados e da própria mineração impraticável. Nestes casos, as técnicas de seleção de atributos podem ser aplicadas, cujo objetivo é selecionar um conjunto de atributos de forma a aumentar a precisão da classificação em relação ao conjunto de todos os atributos ou, então, reduzir o conjunto de atributos inicial sem diminuir significativamente a taxa de precisão de um método de aprendizado em um determinado domínio (KOLLER; SAHAMI, 1996). Em linhas gerais, os métodos de seleção de atributos podem ser categorizados em três grandes tipos.

Método *Wrapper* avalia a qualidade dos subconjuntos dos atributos utilizando o próprio algoritmo de classificação adotado. Normalmente, possuem boa capacidade preditiva

pois avaliam cada conjunto de atributos usando o mesmo algoritmo de classificação que será usado no processo de classificação. No entanto, requer várias execuções do algoritmos de classificação, o que eleva o custo computacional em relação aos outros métodos.

Método *Filter* é um método independente do algoritmo de classificação que será aplicado. Utiliza medidas específicas para avaliar a qualidade dos atributos disponíveis. Esse método pode avaliar cada atributo independente dos outros, determinando o grau de correlação de cada atributo e a classe, ou pode avaliar subconjuntos de atributos, buscando através de estratégias e heurística, aqueles que, em conjunto, melhor identificam as classes (HALL, 2000)

Método *Embedded* é incorporado ao algoritmo de classificação. É aplicado internamente e de forma integrada ao algoritmo de classificação. Algoritmos de indução de árvore de decisão são exemplos típicos pois internamente selecionam os atributos que formarão os rótulos da árvore gerada.

Dados desbalanceados: Uma base de dados é dita desbalanceada, no domínio de classificação, quando o número de casos em uma classe é muito menor do que o número de casos em outra classe (GU et al., 2008). Estas classes com pouca representação são chamadas de classes raras (WEISS, 2004). Algoritmos de classificação são muitos sensíveis a este tipo de desbalanceamento e tendem a valorizar as classes predominantes e a ignorar as classes de menor representação (PHUA; ALAHAKOON; LEE, 2004). Os classificadores resultantes de dados com classes raras apresentam altas taxas de falsos negativos para as classes raras, o que é problemático quando a classe de interesse é justamente a classe rara. Quando existem casos raros, esses casos não são aprendidos, o que é indesejável quando eles pertencem a classe de interesse, afetando negativamente os resultados obtidos.

A técnica de *oversampling* (HAN; KAMBER; PEI, 2011) poderia ser utilizada para resolver o problema das classes desbalanceadas. No entanto, a técnica de *oversampling* é amplamente atacada pela comunidade científica, pois apenas replicam casos positivos existentes. Replicar casos já existentes da classe minoritária, realmente melhora os resultados classificados para essa classe. Entretanto, ocorre o efeito indesejado de *overfitted*, ou seja, modelos muito específicos para estes casos replicados, prejudicando, dessa forma, seu poder de generalização da classe de interesse.

Diante da problemática do *oversampling*, (CHAWLA et al., 2002) desenvolveram um método diferente de fazer *oversampling* da classe minoritária, que consiste na geração de casos sintéticos (artificiais) para a classe de interesse a partir dos casos já existentes.

Estes novos casos serão gerados na vizinhança (k vizinhos mais próximos gerados aleatoriamente) de cada caso da classe minoritária, de forma a fazer crescer a região de decisão e, assim, aumentar o poder da generalização dos classificadores gerados para estes dados. Esses autores chamam este novo método de SMOTE[1]. Portanto, a técnica de

[1] do inglês, *Synthetic Minority Oversampling Technique*

SMOTE foi utiliza nesse trabalho, pois essa técnica reduz consideravelmente a chances de ocorrer o efeito de *overfitted* que normalmente ocorre utilizado a técnica de *oversampling*.

2.2.4 Mineração

Segundo (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) mineração de dados é um componente do processo do KDD que consiste na exploração e análise, em grande conjuntos de dados, em busca por padrões e regras através da aplicação de algoritmos e técnicas computacionais específicas. Devido a sua abrangência falaremos um pouco mais sobre mineração de dados na Seção 2.3.1.

2.2.5 Interpretação dos resultados

Nessa fase de interpretação dos resultados, o conhecimento adquirido através da técnica de mineração de dados deve ser interpretado e avaliado para que o objetivo final seja alcançado. Caso o resultado não seja satisfatório o processo pode retornar a qualquer um dos estágios anteriores ou até mesmo ser recomeçado, conforme pode ser observado na Figura 2, até que o objetivo seja alcançado.

2.3 Mineração de dados Educacionais

2.3.1 Conceito de Mineração de dados

No decorrer da história, o homem sempre aprendeu observando os padrões que ocorrem em seu meio, formulando hipóteses e testando-as para descobrir regras, e durante séculos estes padrões foram extraídos manualmente. Com a informatização de vários setores, os conjuntos de dados têm crescido em tamanho e complexidade, responder uma questão tornou-se crucial (LAROSE, 2004): O que fazer com essa grande quantidade de dados armazenados? As técnicas tradicionais de exploração de dados não são mais adequadas para tratar a grande maioria dos repositórios. Com a finalidade de responder a esta questão, foi proposta, no final da década de 80, a Mineração de Dados (em inglês, *Data Mining* (DM)).

Definida como o processo de descoberta de padrões nos dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996), a mineração de dados tem com o objetivo de transformar dados em informação na forma de fatos, padrões ou realizar predições a partir da mesma (WITTEN; FRANK; HALL, 2011). Estes padrões podem ser expressos de maneira que sejam úteis e nos permitam realizar predições não triviais gerando novos dados.

2.3.2 Mineração de Dados Educacionais

Mineração de dados Educacionais (do inglês, *Educational Data Mining* (EDM)) é uma área que procura desenvolver ou adaptar métodos e algoritmos de mineração existentes,

de tal modo que se prestem a compreender melhor os dados em contextos educacionais, produzidos principalmente por estudantes e professores, considerando os ambientes nos quais eles interagem, tais como Ambientes Virtuais de Aprendizagem, Sistemas Tutores Inteligentes, entre outros (COSTA et al., 2012).

Muitos métodos de EDM surgiram ao longo dos últimos anos, alguns são semelhantes aos de mineração de dados usados em outros domínios, enquanto outros são únicos para a mineração de dados educacional (RYAN; SEIJI; CARVALHO, 2011). Porém, os principais métodos que são usados frequentemente pela comunidade EDM são: Predição (*Prediction*), Mineração de relações (*Relationship Mining*), Descobertas com modelos (*Discovery with Models*) e Descoberta de estrutura(*Structure Discovery*) (BAKER; INVENTADO, 2014).

Não existe somente um objetivo na aplicação de técnicas de EDM, existem os objetivos de investigação aplicada, como a melhoria do processo de aprendizagem e orientação da aprendizagem dos estudantes, bem como existem os objetivos de investigação pura, como a obtenção de uma compreensão mais profunda dos fenômenos educacionais. Essas metas são às vezes difíceis de quantificar e exigem seu próprio conjunto especial de técnicas de medição (ROMERO; VENTURA, 2010).

2.3.3 Tarefas de Mineração de Dados

Diferentes estratégias podem ser utilizadas para minerar as bases de dados na busca por indícios que possam relacionar dados ou fatos. As principais estratégias empregadas, nesta tarefa, incluem a classificação, agrupamento, a associação e a regressão. Em todas estratégias, o objetivo maior é o de poder generalizar o conhecimento adquirido para novas ocorrências do fenômeno ou para outros contextos ou situações parecidas com a utilizada na construção do modelo computacional. Em cada uma destas estratégias diferentes técnicas e algoritmos podem ser aplicados.

- **Classificação** é a estratégia que consiste na busca por um modelo de algum tipo que possa ser aplicado a dados não classificados visando categorizá-los em classes. Um objeto é examinado e classificado de acordo com uma classe definida (HARRISON, 1998). A construção do modelo segundo esta estratégia, pressupõe o conhecimento prévio das possíveis classes e a correta classificação dos exemplos usados na modelagem.

- **Agrupamento (ou *clustering*)** é um processo de partição de uma população heterogênea em vários subgrupos ou clusters mais homogêneos (HARRISON, 1998). A principal diferença entre esta abordagem e a classificação é que no agrupamento não se tem conhecimento prévio sobre as classes predefinidas, os registros são agrupados de acordo com a semelhança.

- **Associação** se baseia em identificar fatos que possam ser direta ou indiretamente associados. O exemplo clássico é determinar quais produtos costumam ser colocados juntos em um carrinho de supermercado, daí o termo "análise de *market basket*". As cadeias de varejo usam associação para planejar a disposição dos produtos nas prateleiras das lojas ou em um catálogo, de modo que os itens geralmente adquiridos na mesma compra sejam vistos próximos entre si (HARRISON, 1998).

- **Regressão** consiste na busca por uma função que represente, de forma aproximada, o comportamento apresentado pelo fenômeno em estudo. Ou seja, é aprender uma função que mapeia um item de dado para uma variável de predição real estimada (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

2.3.4 Técnicas de Mineração de dados

De acordo com Harrison(HARRISON, 1998) não há uma técnica que resolva todos os problemas de mineração de dados. Diferentes métodos servem para diferentes propósitos, cada método oferece suas vantagens e suas desvantagens. Tradicionalmente, os métodos de mineração de dados são divididos em aprendizado supervisionado e não-supervisionado. A diferença entre os métodos de aprendizado supervisionados e não-supervisionados reside no fato de que nos métodos não-supervisionados o conjunto de dados não precisam estar rotulados (HAN; KAMBER; PEI, 2011), ou seja, não é necessário um atributo alvo. Tais métodos, geralmente, usam alguma medida de similaridade entre os atributos (HAN; KAMBER; PEI, 2011). A familiaridade com as técnicas é importante para facilitar a escolha de uma delas de acordo com os problemas apresentados. A seguir, são descritas as técnicas de mineração de dados utilizadas neste estudo.

2.3.4.1 Classificação Bayesiana

É um modelo probabilístico (probabilidade condicional), utilizado na tarefa de classificação em aprendizado supervisionado, baseado no teorema de Thomas Bayes (HAN; KAMBER; PEI, 2011). Segundo o teorema de Bayes, é possível encontrar a probabilidade de um certo evento ocorrer, dada a probabilidade de um outro evento que já ocorreu. O Naive Bayes parte do principio que dado um conjunto contendo grupos divididos por valores e atributos é possível predizer em qual grupo uma nova instância pertence. A fórmula abaixo exemplifica esse conceito. Ela é definida como "a probabilidade de A dado B", ou seja, dado um conjunto de evidências B, qual é a probabilidade da hipótese A estar em B.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

Seguindo o paradigma estatístico, o algoritmo faz uso de fórmulas estatísticas e cálculos de probabilidades para realizar a classificação. O classificador é denominado ingênuo

(*naive*) por assumir que atributos são condicionalmente independentes, ou seja, a informação de um evento não é informativa sobre nenhum outro (que não ocorre na maioria dos problemas práticos). Apesar dessa premissa "ingênua" e simplista, o classificador se reporta bem em várias tarefas de classificação (MITCHELL, 1997). Outro fator que torna o Naive Bayes eficiente é que ele realiza a leitura dos dados do conjunto de treinamento apenas uma vez, para com isso estimar todas as probabilidades requeridas na classificação. O modelo pode ser usado de forma incremental, além do fato de poder ser alterado facilmente com a inclusão de novos dados uma vez que a probabilidade pode ser convenientemente revisada.

Existem dois tipos de modelos estatísticos para o classificador Naive Bayes, o modelo binário e o modelo multinomial. O modelo binário representa um documento através de um vetor binário, indicando a não-ocorrência de um atributo com o valor 0 (zero), enquanto o valor 1 (um) representa no mínimo uma ocorrência. Já o modelo multinomial assume que o documento é representado por um vetor de valores inteiros, caracterizando o número de vezes que cada termo ocorre no documento (MITCHELL, 1997). Neste trabalho foi utilizado o modelo binário (disponível no WEKA)

2.3.4.2 Árvore de Decisão

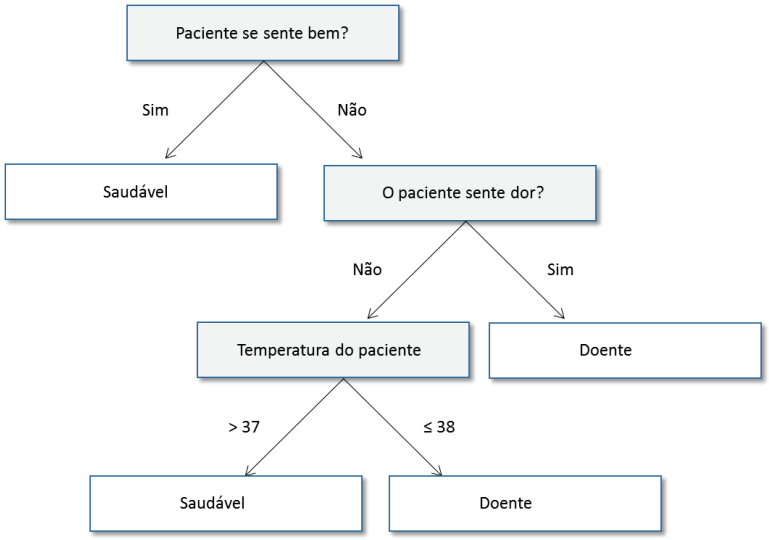
Uma Árvore de Decisão (AD) consiste em uma hierarquia de nós internos e externos que são conectados por ramos. Uma das principais características de uma árvore de decisão é o seu tipo de representação: consiste em uma estrutura hierárquica que traduz uma árvore invertida a qual se desenvolve da raiz para as folhas. A estrutura hierárquica traduz uma progressão da análise de dados no sentido de desempenhar uma tarefa de previsão/classificação. Em cada nível da árvore tomam-se decisões acerca da estrutura do nível seguinte até atingir os nós terminais (nós folha) (BARANAUSKAS; MONARD, 2000).

A aprendizagem por árvore de decisão é um dos métodos mais usados e práticos para a inferência indutiva. A indução mediante a árvore de decisão é uma das formas mais simples de algoritmos de aprendizagem e de muito sucesso. Recebe como entrada um objeto ou uma situação descrita por um conjunto de propriedades ou atributos e retorna como saída uma decisão. Em termos de árvore de decisão, um exemplo é descrito pelos valores dos atributos e pelo predicado meta. O valor do predicado meta é chamado classificação do exemplo. Para cada um dos possíveis valores de atributos, tem-se ramo para outra árvore de decisão (sub-árvore). Cada sub-árvore contém a mesma estrutura de uma árvore.

Uma árvore de decisão é formada por um conjunto de regras de classificação. Cada caminho da raiz até uma folha representa uma destas regras. Cada percurso da árvore de decisão, desde um nó raiz até um nó folha é convertido em uma regra, onde a classe do nó folha corresponde à classe prevista pelo consequente (parte "Então" da regra) e

as condições ao longo do caminho correspondem às condições do antecedente (parte “Se” da regra). A Figura 3 mostra um exemplo simplificado de uma árvore de decisão para diagnóstico de pacientes (BARANAUSKAS; MONARD, 2000).

Figura 3 – Conceito de Árvore de decisão



Fonte: (BARANAUSKAS; MONARD, 2000)

De acordo com (BARANAUSKAS; MONARD, 2000), as regras de classificação que resultam da transformação da árvore de decisão podem ser as seguintes vantagens:

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

2.3.4.3 Máquinas de Vetores de Suporte (SVM)

A Máquina de Vetores de Suporte (do inglês, *Support Vector Machines* (SVM)) é um método baseado na teoria de aprendizagem estatística e otimização matemática (VAPNIK, 1995).

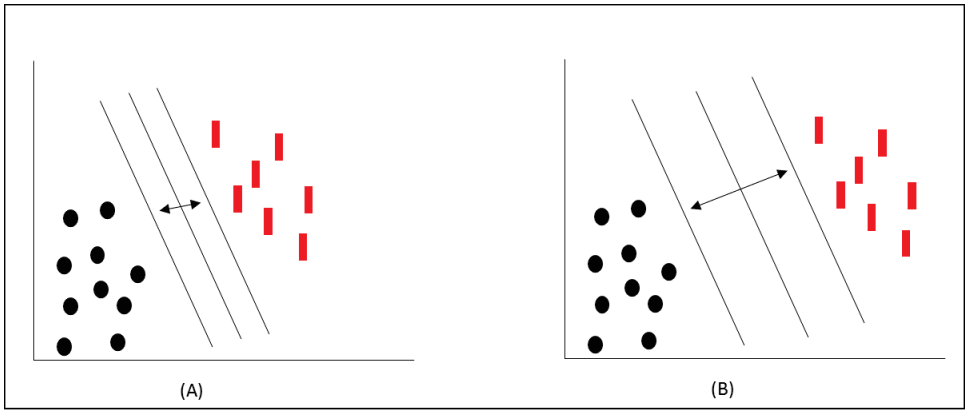
O SVM é um algoritmo que usa um mapeamento não linear para transformar os dados de treino inicial para uma dimensão de alta dimensionalidade. Dentro desta nova dimensão, ele procura o hiperplano de separação ótimo linear (ou seja, uma "fronteira de decisão"que separa as tuplas de uma classe de outra). Com um mapeamento não linear apropriado para uma dimensão suficientemente elevada, os dados de duas classes podem ser sempre separados por um hiperplano. O SVM encontra este hiperplano usando

vetores de suporte ("essenciais"tuplas de formação) e as margens (definidos pelos vetores de suporte) (HAN; KAMBER; PEI, 2011).

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

No método SVM, uma variável de predição é denominada atributo, este atributo quando empregado na construção de hiperplanos é chamado de característica. O conjunto de características selecionadas para descrever um documento na classificação é chamado de vetor e os vetores que se encontram próximos dos hiperplanos construídos para separar as categorias são chamados de vetores de suporte (HAN; KAMBER; PEI, 2011). Na Figura 4 (A) pode ser visto o hiperplano com uma margem de separação estreita, já na Figura 4 (B) o hiperplano como uma margem de separação mais larga, que deverá ocasionar uma maior generalização para os padrões dados como entrada no processo de classificação.

Figura 4 – Hiperplano com margem de separação estreita (A) e margem de separação larga (B)



Fonte: (HAN; KAMBER; PEI, 2011)

O desempenho do algoritmo de Máquina de Vetores de Suporte, assim como outros algoritmos de aprendizado de máquina, é sensível ao ajuste de parâmetros, principalmente em problemas do mundo real (VIANA et al., 2007). Nesse contexto, o SVM possui um pequeno número de parâmetros que proporciona um efeito de regularização sobre o método, como por exemplo, o parâmetro "C" com função de regularização do classificador e os parâmetros de largura do Kernel gaussiano (RBF) chamado *gamma* (γ) ou a dimensão de *kernel* polinomial, dentre outros parâmetros existentes. Uma vez que, os valores desses parâmetros não foram escolhidos adequadamente, podem ocasionar a redução no desempenho geral do classificador.

- **Função de kernel**: é um dos principais elementos usados pelo método SVM para

conseguir separar dados linearmente ou não-linearmente separáveis. Através dela, o SVM constrói uma superfície de decisão que é não-linear no espaço de entrada, mas é linear no espaço de atributos (HAYKIN, 2000). A escolha do *Kernel* é determinada a priori de acordo com a natureza do problema e realizada empiricamente, Algumas funções *Kernel* mais utilizadas são: as polinomiais, função de base radial (do inglês, *Radial-Basis Function* (RBF) conhecida também como Gaussiana e sigmoidais. Um resumo das funções *Kernel* podem ser vista na Figura 5.

Figura 5 – Resumo das funções de *Kernel*

Tipo de *Kernel*	Função $k(x, x')$	Comentários
Polinomial	$(x'^T x + 1)^p$	Dimensão p é definida a priori pelo usuário
RBF	$e^{-\gamma\|x-x'\|^2}$	A largura γ é definida a priori pelo usuário
Sigmoidal	$\tanh(\beta_0 x^T x_i + \beta_1)$	O teorema de Mercer é satisfeito apenas para alguns valores de β_0, β_1.

Fonte: Adaptado. (HAYKIN, 2000)

- **Parâmetro** γ: representa a largura do *Kernel* gaussiano (RBF) é especificado pelo usuário.
- **Parâmetro** $\boldsymbol{P}$: É um parâmetro utilizado para o *Kernel* polinomial, o parâmetro corresponde ao grau do polinômio e é especificado a priori pelo usuário, de acordo com (BOSWELL, 2002), na prática o valor de $\boldsymbol{p}$ varia de 1 a 10.
- **Parâmetro C**: Chamado de coeficiente de custo, esse parâmetro controla a relação entre a complexidade da máquina e o número de amostras de treinamento classificados de maneira incorreta, o qual pode ser visto como um parâmetro de regularização e seu valor precisa ser determinado pelo usuário. Normalmente, este processo é realizado empiricamente através do uso padrão de um conjunto de treinamento e teste (validação). Valores muito altos propiciam a geração de modelos mais complexos, aumentando o risco de *overfitting* enquanto que valores muito baixos podem aumentar o risco de *unerfitting.*

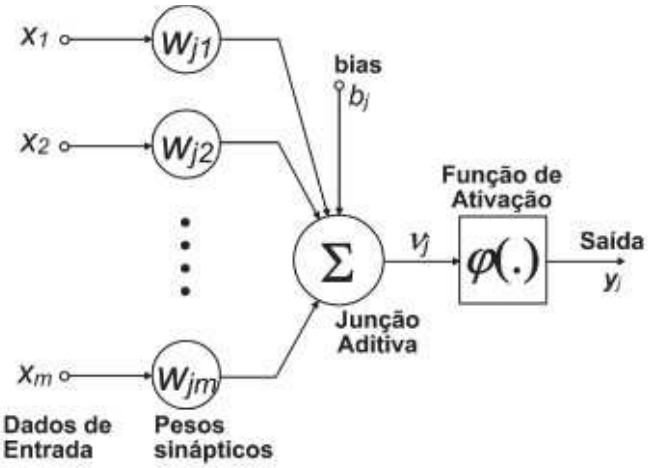
2.3.4.4 Redes Neurais Artificiais

Redes neurais ou redes neurais artificiais (RNA) representam uma tecnologia que tem raízes em muitas disciplinas: neurociência, matemática, estatística, física, ciência da computação e engenharia (HAYKIN, 2000). Consiste em um método para solucionar problemas da área de inteligência artificial, através da construção de um sistema que tenha

circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. São técnicas computacionais que apresentam um modelo inspirado na estrutura neural dos organismos inteligentes, que adquirem conhecimento através da experiência (GOEBEL; GRUENWALD, 1999).

Assim como, o sistema nervoso é composto de bilhões de células nervosas, a rede neural artificial também é formada por pequenos módulos (nós) que simulam o funcionamento de um neurônio. O neurônio artificial é uma estrutura lógico-matemática que procura simular a forma, comportamento e as funções de um neurônio biológico (GOEBEL; GRUENWALD, 1999), é um elemento fundamental no processamento de uma rede neural (HAYKIN, 2000). O processo de aprendizado é o conceito que norteia a rede neural e é realizado através dos neurônios. A Figura 6 ilustra a estrutura básica de um neurônio artificial (*perceptron*).

Figura 6 – Estrutura de um neurônio artificial



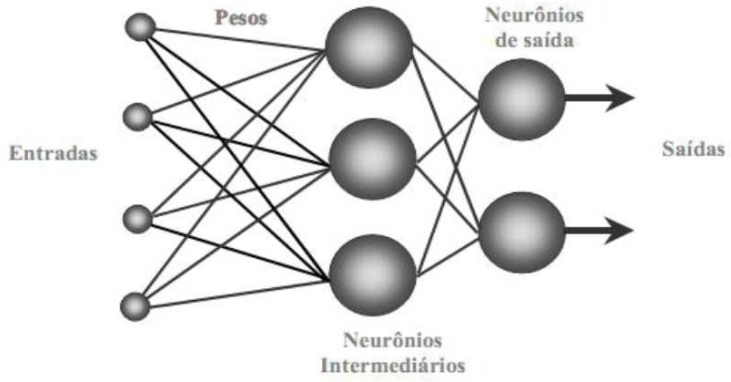
Fonte: Adaptado. (HAN; KAMBER; PEI, 2011)

- Um conjunto de sinapses ou conexões de entrada ponderadas com um peso sináptico. Desse modo, um sinal X_j na entrada da sinapse j conectada ao neurônio k é multiplicado pelo peso w_{kj};
- Uma junção de soma responsável pela adição dos sinais de entrada ponderados pelos respectivos pesos do neurônio;
- Uma função de ativação geralmente não linear representando a ativação de saída do y_k neurônio.
- O processo que busca a melhor calibração dos pesos w_{kj} é conhecido como processo de aprendizado ou treinamento da rede

Um neurônio (*perceptron*) corresponde a um hiperplano no espaço de atributos, podendo separar duas classes linearmente separáveis, mas não podendo separar mais que

duas classes ou fazer separações não-lineares. Não é possível separar duas classes linearmente não separáveis com apenas um neurônio. Um número maior de neurônios é necessário para classes não separáveis linearmente. A solução é combinar vários neurônios em que cada um deles corresponde a um hiperplano, para criar hiper superfícies. Esta solução é o esquema de uma *multilayerperceptron* (MLP) - múltiplas camadas de neurônios, mostrado na Figura 7.

Figura 7 – Rede Neural Artificial de múltiplas camadas



Fonte: Adaptado. (HAN; KAMBER; PEI, 2011)

Uma das principais vantagens das redes neurais é sua variedade de aplicação, mas os seus dados de entrada são difíceis de serem formados e os modelos produzidos por elas são difíceis de entender (HARRISON, 1998). A técnica de redes neurais é apropriada às seguintes tarefas: classificação, regressão e agrupamento. Exemplos de redes neurais: Perceptron, Rede MLP, Redes de Kohonen, Rede Hopfield, Rede BAM, Redes ART, Rede IAC, Rede LVQ, Rede Counterpropagation, Rede RBF, Rede PNN, Rede Time Delay, Neocognitron, Rede BSB (AZEVEDO, 2000) e (HAYKIN, 2000). Para este estudo utilizaremos a rede *multilayerperceptron* (MLP), disponível na ferramenta weka.

3 TRABALHOS RELACIONADOS

Vários trabalhos têm sido relatados na literatura relacionada com abordagens preditivas para identificar automaticamente os estudantes que irão ter sucesso ou não em cursos introdutórios de programação. Na presente análise sobre os trabalhos existentes com efeitos semelhantes aos da pesquisa relatada neste trabalho, organizamos nossa discussão no geral, para abordagens de previsão mais específicas, focando apenas no domínio da introdução a programação, utilizando apenas dados acadêmicos. Além dos métodos de previsão, também discutiremos brevemente aspectos relacionados com o pré-processamento de dados e ajustes finos dos algoritmos.

3.1 Métodos de predição de Insucesso

Existem várias tentativas de construir modelos teóricos que explicam o fenômeno do insucesso na educação em diversas modalidades de ensino. A maioria deles revelam uma série de características comuns como a utilização de técnicas de EDM com algoritmos de predição, a fim de prever o sucesso ou insucesso dos estudantes.

A abordagem proposta em (MARQUEZ-VERA; MORALES; SOTO, 2013), propõe, por meio de questionários aplicados ao estudantes, desenvolver um modelo preditivo, a fim de identificar o mais cedo possível os alunos em risco de insucesso escolar no ensino médio ou secundário, através de técnicas de EDM e a utilização das técnicas de pré-processamento dos dados, com dez algoritmos de classificação "caixa-branca", tais como regras de indução e árvores de decisão. Por fim, os autores mostraram que os algoritmos de classificação podem ser utilizados com sucesso, a fim de prever o desempenho acadêmico do aluno e, em particular, para modelar a diferença entre estudantes propensos ao insucesso e ao sucesso. No entanto, seu estudo difere do nosso por meio da coleta dos dados que não é totalmente automática e não ficamos limitados a apenas um método de classificação. No artigo citado, os autores ficaram restritos a utilizar apenas algoritmos de árvores de decisão e regras de indução.

O trabalho de (BAYER et al., 2012), utiliza algoritmos de previsão para classificar os estudantes em risco ao insucesso durante o curso. Eles usam dados pessoais dos estudantes enriquecidos com dados relacionados com comportamentos sociais. Os autores fazem uso de técnicas de pré-processamento de dados e sete algoritmos de classificação *(ZeroR, Naive Bayes, SMO, IB1, PART, OneR, J48)*, a fim de descobrir qual dos algorítimos fornece os melhores resultados. Os autores conseguiram chegar a taxas de acurácia de até 93.51%. No entanto, as melhores taxas são apresentados no final do curso. Portanto, considerando que o objetivo do estudo é o de identificar e prever o insucesso com maior antecedência possível, o estudo seria mais relevante se os resultados satisfatórios (93.51%)

fossem obtidos no início do curso. Nosso trabalho difere desta abordagem uma vez que somos capazes de prever o insucesso mais cedo, dando melhores chances para educadores e professores de tomar uma decisão adequada. Além disso, os autores mencionam apenas a etapa de seleção de atributos na tarefa de pré-processamento de dados, de acordo com nosso trabalho essas taxas de precisão poderiam ser aumentadas se os autores usassem outras etapas de pré-processamento como balanceamento dos dados e ajustes finos nos parâmetros dos algoritmos.

O trabalho de (ARORA; SINGHAL; BANSAL, 2014), utiliza apenas o algoritmo de predição baseado na Função de Base Radia (RBF) para prever as notas obtidas pelos estudantes em uma determinada disciplina baseadas nas notas das disciplinas anteriores. De acordo com os resultados obtidos, os estudantes são classificados em grupos susceptíveis ao insucesso ou não. Diferentemente do nosso trabalho, neste trabalho os autores usam apenas um tipo de algoritmo de predição para criar o modelo e não usam os recursos de pré-processamento de dados existentes. A abordagem proposta por nós, além de utilizar vários passos de dados de pré-processamento também usa quatro algoritmos de previsão que utilizam diferentes métodos de aprendizagem.

O trabalho de (aES; CRUZ; aO, 2014) apresenta uma arquitetura que utiliza técnicas de EDM para prever e identificar os estudantes que estão em risco de abandono. Para fazer isso, eles usam dados dos cursos de engenharia civil, mecânica e de produção da Universidade Federal do Rio de Janeiro (UFRJ). De acordo com o experimento, o classificador Naive Bayes apresenta a maior taxa de verdadeiro positivo para todo o conjunto de dados usado no estudo. Mas os autores não fazem uso da etapa de pré-processamento dos dados nem ajustes finos nos parâmetros dos algoritmos. Este aspecto difere do nosso trabalho pois usamos esta questão para obter melhores resultados.

O trabalho de (WATSON; LI; GODWIN, 2013) apresenta um novo algoritmo chamado *Watwin*, um algoritmo dinâmico projetado para prever o desempenho dos estudantes em cursos de programação. Esta abordagem é baseada na análise de dados, resumidamente a abordagem é realizada da seguinte maneira: quando um estudante compila o seu programa em um computador da universidade, a abordagem grava o código fonte do programa e coleta informações sobre sucesso ou o fracasso do programa, o tempo de execução, mensagens de erro e número da linha de código. Esta informação é utilizada pelo algoritmo *Watwin* para prever a falha dos estudantes. Os resultados indicam que o método é capaz de alcançar uma eficácia de até 75%.

Um modelo para prever os níveis de desempenho dos estudantes é proposto por (ER, 2012), utilizando três algoritmos de aprendizado de máquina (Classificador de aprendizagem baseado na instância (k-Star), Árvore de Decisão (C4.5) e Naive Bayes). O objetivo geral do estudo é propor um método para a previsão dos estudantes em risco de insucesso na modalidade de ensino a distância, somente com os registros do ambiente virtual de aprendizagem. O experimento foi realizado em três etapas que correspondem a diferentes

estágios de um semestre. Em cada etapa, o conjunto dos dados era aumentado, a cada etapa os resultados foram melhorando. Obtendo nas últimas etapas uma taxa de 85.33% de acurácia.

(LYKOURENTZOU et al., 2009), através da utilização dos algoritmos de predição Máquina de Vetor de Suporte, Rede Neural e Fuzzy ARTMAP tem como objetivo prever a evasão de estudantes em cursos online. Obtendo a maioria dos dados através de questionários aplicados aos estudantes durante o curso. Os autores usam três métricas para avaliar a precisão global dos resultados (acurácia, sensibilidade e precisão). O método proposto obteve uma taxa de classificação de estudantes no geral de 75-85% no início do curso.

(MARTINHO; NUNES; R., 2013), propõe desenvolver um sistema que tem como objetivo identificar os grupos de alunos em risco de evasão em cursos de tecnologia do Instituto Federal de Mato Grosso com base nos registros acadêmicos e socioeconômicos dos alunos, utilizando a técnica de Rede Neural Fuzzy ARTMAP. Esta técnica se mostrou bastante interessante na aplicação do problema da identificação dos alunos propensos a evasão, a análise dos resultados mostraram que o sistema proposto é satisfatório, com exatidão global entre 85 e 94%. No entanto, os autores não informam em qual período do curso são obtidos esses resultados, não utilizam técnicas de pré processamento de dados e alguns dados foram obtidos através da aplicação de um questionário aos alunos no momento da inscrição do curso.

Como visto anteriormente, existem muitas abordagens que estudam o fenômeno do insucesso dos estudantes através de técnicas de EDM. Embora essas abordagens apresentem formas promissoras para identificar estudantes que possam ter insucesso, eles são um pouco limitados em termos de predição do insucesso com precisão e com antecedência suficiente para permitir uma intervenção pedagógica em tempo hábil. Além disso, estas abordagens não investigam corretamente a influência do pré-processamento de dados e ajuste fino nos algoritmos na eficácia das técnicas analisadas. A fim de mitigar tais limitações, foi realizado um estudo comparativo que avaliou a eficácia das técnicas mais promissoras de EDM utilizadas pelas obras existentes para identificar os estudantes propensos ao insucesso. Em nosso experimento, foi avaliada a eficácia dessas técnicas para prever o mais cedo possível o insucesso acadêmico dos estudantes em cursos introdutórios de programação, bem como a influência das técnicas de pré-processamento de dados e ajustes finos nos algoritmos de predição sobre as previsões.

3.2 Pré-processamento dos dados

O sucesso das técnicas de aprendizagem de máquina é significativamente influenciada pela qualidade dos dados. Quaisquer dados defeituosos, ruidosos, supérfluos e inadequados podem levar a resultados defeituosos (MITCHELL, 2006). No entanto, geralmente dados do mundo real tendem a ser sujos, incompletos e inconsistentes. Portanto, o pré-

processamento de dados, que estão sendo contemplados nas duas primeiras camadas de nosso modelo, é um passo importante no processo de mineração de dados tendo um impacto sobre o sucesso de um projeto de mineração de dados (HU, 2003) e (ZAKI; JR, 2014).

Partindo desse princípio (CRONE; LESSMANN; STAHLBOCK, 2006) pretendem investigar o impacto de diferentes técnicas de pré-processamento sobre os resultados dos algoritmos de árvores de decisão, redes neural e máquinas de vetores de suporte. Através da utilização de uma base de dados real de *marketing*. Os autores concluem que os algoritmos de redes neurais e máquina de vetores de suporte foram as que tiveram os melhores resultados. Analisando, apenas as etapas de detecção de *outliers*, falta de valores, discretização e normalização dos dados (KOTSIANTIS; KANELLOPOULOS; PINTELAS, 2006), propõem examinar se com apenas essas etapas é possível que algoritmos de classificação tenham algum tipo de ganho no seu poder de generalização, após esses procedimentos. Os autores chegaram a conclusão que após essas etapas os algoritmos tiveram uma melhora considerável nos resultados da generalização.

(ROMERO; VENTURA, 2010) em seus trabalhos, têm fornecido referências valiosas em relação à descoberta de conhecimento em ambientes educacionais. Em um dos seus estudos eles desenvolvem um tutorial para aplicação de técnica de mineração de dados no ambiente virtual de aprendizagem (AVA) MOODLE[1], com o objetivo de apresentar teoria e prática a todos os interessados nesta nova área de pesquisa e, em especial para professores *online* e os administradores de *e-learning*. Mostrando todo o processo passo a passo da mineração de dados no AVA MOODLE, bem como algumas etapas de pré-processamento dos dados e a forma de aplicar as principais técnicas de mineração de dados utilizadas.

Portanto, de acordo com os documentos mencionados acima fica evidente a importância das tarefas de pré-processamento dos dados. Desta maneira, neste estudo utilizamos e apresentamos de forma transparente todas as etapas de pré-processamento dos dados realizadas. Através dos experimentos concluímos que o pré-processamento dos dados teve um impacto significativo nos resultados.

3.3 Ajustes finos nos algoritmos

Grande parte dos algoritmos de predição possuem parâmetros cujos valores podem ser especificados pelo usuário. Esses parâmetros, em geral, influenciam diretamente no desempenho destes algoritmos, principalmente em problemas do mundo real (VIANA et al., 2007). Os métodos de escolha destes parâmetros variam amplamente e são conhecidos como *tuning*. A abordagem mais comum para definir os valores dos parâmetros é por tentativa e erro (KOHAVI; JOHN, 1995).

[1] do inglês *Modular Object-Oriented Dynamic Learning Environment*

Para Rede Neural, por exemplo, (BASHEER; HAJMEER, 2000), afirmam que bons valores para os parâmetros são encontrados, geralmente, por meio de tentativa e erro. Segundo os autores, a escolha de valores para os parâmetros do algoritmo de aprendizado *backpropagation* para Rede Neural influencia na convergência do aprendizado e no desempenho geral da rede.

De acordo com (CHAPELLE et al., 2002) o desempenho do algoritmo de Máquina de Vetor de Suporte é diretamente influenciado pela escolha da função de *kernel* e os valores de seus parâmetros. No entanto, escolha manual dos parâmetros é indesejável, pois é imprecisa e não garante resultados de qualidade (IMBAULT; LEBART, 2004). Portanto, técnicas de otimização dos parâmetros como por exemplo *Grid-Search* é utilizada em busca dos melhores parâmetros.

Portanto, os ajustes dos parâmetros de um algoritmo possui um papel importante por definirem seu comportamento e consequentemente, interferindo na qualidade das soluções encontradas. Logo, serão realizados os ajustes nos parâmetros dos algoritmos utilizados neste estudo.

4 MÉTODO

Está sendo proposto neste trabalho uma comparação entre algoritmos de predição, cujo o objetivo é encontrar a técnica mais eficaz com o intuito de identificar, em tempo hábil para intervenção pedagógica, os estudantes propensos ao insucesso nas disciplinas iniciais de programação. Foi analisada a eficácia de quatro algoritmos de previsão com diferentes métodos de predição em dois conjuntos de dados distintos usando a abordagem de avaliação *Goal*, *Question*, *Metrics* (GQM) (BASILI; CALDIERA; ROMBACH, 1994).

A seção está organizada como segue: na Seção 4.1 apresentamos os dados selecionados para o experimento. A Seção 4.2 apresenta quais algoritmos foram escolhidos para a tarefa de predição, e finalmente, a Seção 4.3 mostra as ferramentas utilizadas para a execução do experimento.

4.1 Seleção dos Dados

Existem maneiras distintas de se obter os dados para o desenvolvimento da pesquisa. A maneira mais comum é através da aplicação de questionários aos estudantes ao longo do curso. O problema desta alternativa é que se torna um pouco intrusiva e custosa, pois exige a participação dos estudantes no processo de coleta dos dados (ARAQUE; ROLDáN; SALGUERO, 2009), embora com esse método seria possível conseguir dados mais precisos e focados no desenvolvimento de um modelo de predição mais eficaz.

A outra opção para conseguir os dados é através da utilização de técnicas de KDD aplicadas aos sistemas existentes na universidade. Apesar de que, os dados obtidos sejam menos precisos e informativos do que os dados adquiridos através de questionários, com essa alternativa temos uma maior quantidade de dados e o processo se torna transparente para os estudantes. Por esse motivo optamos pela segunda alternativa, utilizando apenas os dados acadêmicos fornecidos pelos sistemas utilizados pela Universidade Federal de Alagoas (UFAL).

O departamento de tecnologia da informação da universidade nos forneceu todos os dados que precisávamos sobre os estudantes. A fim de evitar problemas legais, dados pessoais como nome e documentos pessoais foram removidos, tornando os dados totalmente anônimos.

A UFAL oferece cursos de graduação e pós-graduação nas modalidades presencial e a distância. Nos cursos presenciais são cerca de 28 mil alunos matriculados nos 84 cursos de graduação e 39 cursos de pós-graduação. Nos cursos a distância, há mais de quatro mil estudantes matriculados (UFAL, 2014). Os cursos, geralmente, são divididos em semestres, onde diferentes disciplinas são ensinadas ao longo destes semestres. Cada semestre tem geralmente cinco disciplinas, o tempo de cada disciplina varia de acordo

com a modalidade de ensino.

Os dados para esse trabalho foram obtidos a partir de duas fontes de dados diferentes e independentes dos cursos de Sistemas de Informação, Ciência da Computação e Engenharia de Computação, nas modalidades de ensino presencial e a distância.

A escolha das disciplinas iniciais de programação nas diferentes modalidades de ensino pesquisadas ocorreu por que, de acordo com os autores (ROMERO et al., 2013), (ARAQUE; ROLDáN; SALGUERO, 2009) e (PAURA; ARHIPOVA, 2013) a maioria dos casos de insucesso, ocorrem nas disciplinas iniciais do curso. Outra razão para a escolha destas disciplinas foi por serem disciplinas fundamentais para os cursos de Sistemas de Informação, Ciências da Computação e Engenharia de Computação.

4.1.1 Ensino a distância

Os dados utilizados da modalidade de ensino à distância foram referentes ao curso de Sistemas de Informação (SI) EAD do ano de 2013, com cerca de 262 estudantes matriculados entre os polos de Maceió, Maragogi, Santana do Ipanema, Olho d'água das Flores e Arapiraca. Estes estudantes cursaram a disciplina de "Algoritmo e Estrutura de Dados I" que utiliza a linguagem de programação Python (PYTHON, 2015) para o ensino introdutório de programação. Essa disciplina é dividida em oito semanas, em cada semana os estudantes recebem notas de acordo com as atividades desenvolvidas e no fim da quinta semana é realizada a primeira avaliação.

Os dados dos estudantes matriculados, nessa modalidade de ensino, estão dispostos em duas bases de dados separadas: a primeira base de dados pertence ao Sistema de Informação para o Ensino (SIE) (UFAL, 2014), que é o sistema de controle acadêmico usado pela universidade. A segunda base de dados pertence ao Ambiente Virtual de Aprendizagem Moodle (MOODLE, 2014). A junção destas duas fontes de dados resultaram nos seguintes atributos: Id, idade, sexo, renda, cidade, estado civil, matricula, semestre, turma, período, ano de ingresso no curso dos estudantes, frequência de acesso ao AVA, participação no fórum de discussões, quantidade de arquivos enviados e baixados do AVA, uso das ferramentas educacionais fornecidas pelo AVA como: *blog, glossary, quiz, wiki, message*, notas relacionados com as atividades desenvolvidas dentro do AVA, notas dos estudantes na disciplina e a situação do estudante (aprovado ou reprovado).

4.1.2 Ensino presencial

Os dados utilizados da modalidade de ensino presencial pertencem aos cursos de Ciência da Computação e Engenharia de Computação dos anos de 2013/2014 com cerca de 161 alunos matriculados. Os dados foram extraídos da disciplina de "Programação 1" que utiliza a linguagem de programação *C* (KERNIGHAN, 1988) para o ensino introdutório de programação. Essa disciplina está dividida em 16 semanas, separadas em duas avali-

ações, caso o estudante não alcance a média mínima será realizada outra avaliação. No entanto, antes das avaliações os estudantes são submetidos a alguns exercícios.

Os dados dos estudantes, desta modalidade de ensino, também estão dispostos em duas bases de dados distintas, embora sejam modalidades de ensino diferentes a primeira base continua sendo a base do sistema de controle acadêmico utilizado pela Universidade SIE. No entanto, a segunda base de dados pertence ao sistema Huxley. O *The Huxley* é uma ferramenta de apoio ao ensino de disciplinas de programação, onde estudantes fazem os exercícios de programação propostos pelos professores e um juiz online realiza as correções (HUXLEY, 2015). Os dados como exercícios realizados, submetidos e corretos, nota atribuída a cada exercício foram extraídos dessa ferramenta. A junção destas duas fontes de dados resultaram nos seguintes atributos: Id, idade, sexo, renda, cidade, estado civil, matricula, semestre, turma, período, ano de ingresso no curso dos estudantes, quantidade de exercícios que o estudante realizou, quantidade de exercício submetidos, quantidade de exercícios corretos, notas dos estudantes na disciplina e a situação do estudante (aprovado ou reprovado).

4.2 Seleção dos Algoritmos

Muitos métodos de EDM surgiram ao longo dos últimos anos, alguns são semelhantes aos de mineração de dados usados em outros domínios, enquanto outros são únicos para a mineração de dados educacionais (RYAN; SEIJI; CARVALHO, 2011). Segundo, (GIBERT; MARRE; CODINA, 2010), a escolha da técnica de mineração de dados depende do problema de negócio a ser solucionado e das características dos dados disponíveis para análise. Neste trabalho utilizamos o método de predição. No entanto, existem três tipos de predição: classificação, regressão e estimação de densidade. A estimação de densidade é raramente utilizada na EDM devido a falta de independência estatística (RYAN; SEIJI; CARVALHO, 2011), regressão e classificação são os tipos mais comuns de predição. Em regressão a variável preditiva pode ser numérica ou contínua, na classificação a variável preditiva pode ser categórica ou discreta (ROMERO; VENTURA, 2010).

De acordo com o conjunto de dados e o objetivo do estudo, o tipo de predição mais apropriado para o estudo é o de classificação.

Os algoritmos de predição utilizados foram: classificador probabilístico simples com base na aplicação do teorema de Bayes (ver Seção 2.3.4.1), Árvore de Decisão (ver Seção 2.3.4.2), Máquina de Vetor de Suporte (ver Seção 2.3.4.3)e Rede Neural (ver Seção 2.3.4.4). Esses métodos foram selecionados, uma vez que apresentam uma boa eficácia em diferentes domínios (CARUANA; NICULESCU-MIZIL, 2006). Além dos seus respectivos sucessos em outros domínios, segundo (WU et al., 2008) tem sido bastante utilizados em abordagem existentes para a tarefa de predição.

4.2.1 Validação

A tarefa de validação, no domínio de mineração de dados, é um processo que determina o grau de confiabilidade do modelo construído em relação aos dados apresentados. A metodologia de testes adotada no trabalho foi a validação cruzada com K-folds.

No método da validação cruzada K-folds, o conjunto de treinamento original é dividido em K subconjuntos. Destes K subconjuntos, um subconjunto é retirado para ser utilizado na validação do modelo e os K-1 subconjuntos restantes são utilizados no treinamento. O processo de validação cruzada é, então, repetido K vezes, de modo que cada um dos K subconjuntos sejam utilizados exatamente uma vez como dado de teste para a validação do modelo (HAN; KAMBER; PEI, 2011).

O resultado final desse processo é o desempenho médio do classificador nos K-1 testes. O objetivo de repetir os testes múltiplas vezes é aumentar a confiabilidade da estimativa da precisão do classificador.

A validação cruzada é recomendada quando se trabalha com um pequeno número de amostras. Quando tem-se um número maior de amostras, é recomendado outro método de validação, devido ao alto custo computacional.

4.2.2 Métrica

Dado que os modelos de predição são desenvolvidos para serem aplicados a dados diferentes daqueles que foram usados para os construir, torna-se essencial avaliá-los para medir o seu poder de generalização. Para melhor caracterizar a eficácia dos métodos EDM analisados neste trabalho, decidimos adotar a métrica *F-Measure* (HAN; KAMBER; PEI, 2011), que é amplamente utilizado em domínios como a recuperação de informação, aprendizado de máquina e outros domínios que envolvem classificação binária (OLSON; DELEN, 2008). Em suma, *F-Measure* é a média harmônica entre *Precision* e *Recall* (HAN; KAMBER; PEI, 2011), conforme descrito abaixo:

Para o cálculo destas medidas é necessário categorizar as entidades da seguinte forma:

- Verdadeiro Positivo (TP): é o número de casos positivos corretamente classificados como positivo.
- Verdadeiro Negativo (TN): é o número de casos negativos corretamente classificados como negativo.
- Falsos Positivos (FP): é o número de casos negativos incorretamente classificados como positivo.
- False Negativo (FN): é o número de casos positivos incorretamente classificados como negativo.

Recall e *Precision* são duas medidas de avaliação amplamente utilizadas. *Recall* pode ser considerada como uma medida de integridade, enquanto, *Precision* pode ser considerada como uma medida de precisão ou de fidelidade. *Recall* mede a capacidade de um sistema de encontrar o que se quer e *Precision* mede a capacidade do mesmo sistema em rejeitar o que não se quer (HAN; KAMBER; PEI, 2011). Podemos obter estas medidas a partir das fórmulas:

$$Precision : \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall : \frac{TP}{TP + FN} \tag{4.2}$$

F-Measure mede a eficiência do sistema levando em consideração o erro nas duas classes. É necessário que o acerto nas duas classes aumente para que a métrica aumente. É obtida da seguinte forma:

$$F - measure : \frac{2.Precison.Recall}{Precison + Recall} \tag{4.3}$$

4.3 Instrumentação

Para escolha das ferramentas foram levadas em consideração alguns parâmetros, tais como: se existia uma conectividade entre as ferramentas, se suas funcionalidades atendiam as necessidades exigidas no trabalho, usabilidade, seu desempenho computacional e o custo.

Não sendo o objetivo desta dissertação constituir um manual de utilização das ferramentas escolhidas é, no entanto, oportuno efetuar uma breve descrição das ferramentas utilizadas. Deste modo, em seguida descreveremos as funcionalidades essenciais apresentadas pelos sistemas Pentaho e Weka, uma vez que se tratou da ferramenta que recorremos para a realização de todas as experiências inerentes a este trabalho.

4.3.1 Pentaho

De acordo com (PENTAHO, 2015), esta plataforma é a ferramenta livre de *Business intelligence*(BI) bastante popular, sendo disponibilizada pela licença de uso comercial em sua versão *Enterprise* e pela licença GNU GPL em sua versão *Community*. O pacote *Pentaho Open BI Suite* é diferente dos fornecedores tradicionais de plataformas de BI, pois sua plataforma é baseada em processos e é orientada a soluções por componentes integrados.

O pacote *Pentaho Open BI*, consiste dos seguintes componentes macros: Funcionalidades de BI para usuários finais, Pentaho Design Studio e plataforma de BI.

As funcionalidades de BI para usuários finais incluem as capacidades de: relatórios, análises, workflow, painel de instrumentos e mineração de dados.

O Pentaho Design Studio é um conjunto de ferramentas de projeto e administração que integradas, permitem aos desenvolvedores a criação de relatórios, painel de instrumentos, modelos de análise, regras de negócios e processos de BI.

A plataforma de BI da Pentaho é o núcleo de sua arquitetura, sendo baseada em processos, pois se consubstancia no controlador central que utiliza das definições dos processos como insumo para a definição dos processos de BI na plataforma. Ainda, é considerada orientada a solução, devido ao fato de que as operações na plataforma são especificadas pelos processos e operações de forma coletiva para a definição das soluções de um problema de BI.

A plataforma de BI do Pentaho incorpora diversos módulos, que executados como parte de um processo habilitam à solução, procedimentos de BI. Os principais módulos são:

- Pentaho BI Platform and Server (Pentaho User Console e Pentaho Administrator Console);
- Pentaho Report (Pentaho Reporter Designer);
- Pentaho Data Integration Community Edition – PDI CE (Kettle);
- Pentaho Analysis Services Community Edition – PAS CE (Mondrian);
- Pentaho Data Mining (Weka).

Neste estudo será utilizado somente o componente Pentaho Design Studio e plataforma de BI com o módulo de Pentaho Data Integration Community Edition – PDI CE (Kettle).

O Pentaho Data Integration, executa os processos de extração, transformação e carga, por meio de uma interface gráfica intuitiva, como podemos verificar na Figura 8. Suporta um elevado número de formatos de entrada e saída. Cada processo é criado com uma ferramenta gráfica onde as ações são especificadas sem escrever código.

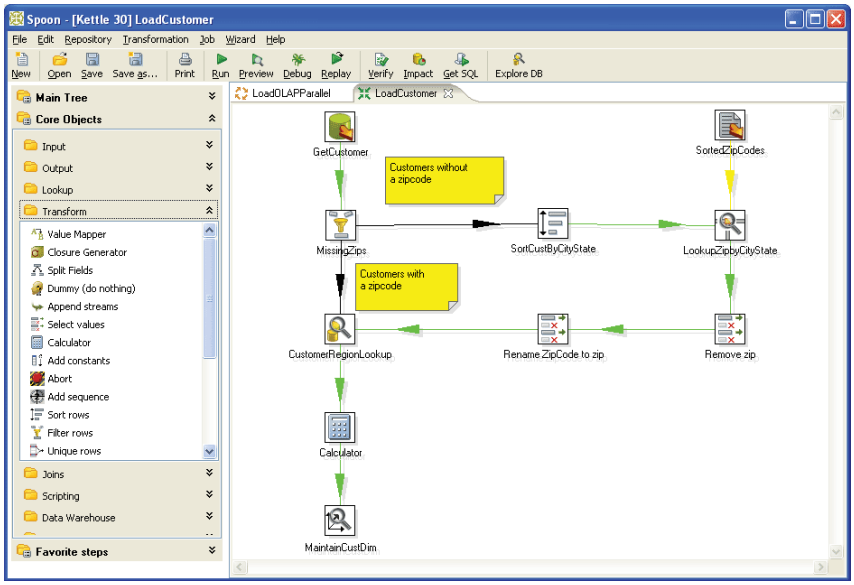
O motivo da escolha do Pentaho Data Integration Community Edition se deu devido às suas características de ser open source, possuir uma ampla comunidade de usuários e ser extensível através de uma API[1] Java.

4.3.2 Weka

Para a execução de algoritmos de mineração de dados selecionados (ver Seção 2.3.4), a ferramenta escolhida foi a Weka. WEKA (*Waikato Environment for Analysis Knowledge*) é uma ferramenta desenvolvida por pesquisadores da Universidade de *Waikato* na Nova Zelândia, sendo uma das mais populares entre a comunidade de mineração de dados. Ela possui código aberto, interface amigável, agrega um conjunto de algoritmos

[1] do inglês *Application Programming Interface*

Figura 8 – Transformação em Kettle



Fonte: (PENTAHO, 2015)

de classificação, regras de associação, regressão, pré-processamento e agrupamento, todos implementados em JAVA (WEKA, 2014). Para inserir os dados no WEKA, é necessário colocá-los em um formato que seja entendido por ele. O seu método preferido para carregar os dados é o formato de arquivo de atributo relação (ARFF).

Grande parte dos recursos do software WEKA encontra-se acessível através da sua interface gráfica, que passamos a descrever a seguir.

A interface gráfica da ferramenta WEKA GUI[2], possui uma janela – "WEKA GUI Chooser" conforme a Figura 9, que permite aos utilizadores escolherem quais as aplicações que pretendem utilizar de modo a extraírem informação dos seus dados.

Figura 9 – Weka (GUI Chooser)

Fonte: (WEKA, 2014)

[2] do inglês *Graphical User Interface*

- **Explorer**: Proporciona um ambiente gráfico intuitivo de manipulação de dados pela utilização de diversos algoritmos. Trata-se da interface mais fácil de usar, conduzindo o utilizador através de menus e formulários, impedindo-o de fazer escolhas não aplicáveis e simultaneamente apresentando pop-ups de informação relativos ao preenchimento de vários campos.

- **Experimenter**: Permite testar técnicas diferentes em classificação ou regressão, de modo a compará-las. Apesar destas operações serem igualmente possíveis no Explorer como no KnowledgeFlow, no Experimenter, no entanto, é possível escolher desde diversos conjuntos de dados a serem utilizados num só experimento, como várias técnicas a serem experimentadas, e até o número de repetições (runs) do teste em questão, entre outras escolhas.

- **KnowledgeFlow**: Permite o desenvolvimento de projetos de mineração de dados num ambiente gráfico com fluxos de informação. Por outro lado, dentre várias vantagens que possui, destaca-se o layout intuitivo. Assim como o fato de permitir o processamento de dados em batch[3] ou de modo incremental, que por sua vez permitem a sua aplicação a conjuntos de dados de elevada dimensão. Além do mais, possibilita o processamento paralelo, em que cada fluxo de dados distinto é processado na respectiva thread[4].

- **Simple CLI**: Proporciona uma interface que permite a execução direta de comandos do WEKA. Embora, disponibilize todas as funcionalidades, requer um elevado grau de conhecimento dos comandos que poderão ser utilizados.

Algumas características foram levadas em consideração para escolha da ferramenta Weka, tais como: facilidade de aquisição, instalação e disponibilidade para fazer o download diretamente da página de desenvolvedor com nenhum custo de operação; Presença de várias versões de algoritmos de mineração, disponibilidade de recursos estatísticos para comparar os resultados entre os algoritmos, possui uma ampla comunidade de usuários e ser extensível através de uma API Java.

Alguns algoritmos implementados no Weka tem nomenclatura própria como J48, SMO, MultilayerPerceptron corresponde aos algoritmos de Arvore de Decisão C4.5, Máquina de Vetor de Suporte e Rede Neural respectivamente.

É importante destacar que ambas ferramentas são de código aberto e desenvolvidas na linguagem de programação Java e podem ser extensível através de uma API Java.

---

[3] Termo referente a um processamento de dados que ocorre através de um conjunto de tarefas que se encontram enfileiradas, sendo que o sistema operativo apenas processa a próxima tarefa após o término completo da tarefa anterior.

[4] Forma de um determinado processo se dividir em duas ou mais tarefas que possam ser executadas simultaneamente.

5 REALIZAÇÃO DO EXPERIMENTO

Neste capítulo, descrevemos com detalhes as etapas de pré-processamento realizadas sobre os conjuntos de dados e os ajustes finos realizados nos algoritmos de predição utilizados no trabalho. A Seção 5.1 descreve em detalhes o pré-processamento é na Seção 5.2 descreve os ajustes finos realizados.

O experimento foi executado em um Computador HP ProBook 2.6 GHz Core I5 com 8GB de memória, com o sistema operacional Windows 8.1.

5.1 Pré-processamento dos dados

Em muitos campos da ciência da computação, tais como Reconhecimento de Padrões, Recuperação de Informação, Aprendizagem de Máquina, Mineração de Dados e Big Data, é preciso reorganizar os dados brutos (ZHANG; ZHANG; YANG, 2003). Pois, geralmente dados do mundo real tendem a ser sujos, incompletos e inconsistentes. Portanto, antes de aplicar técnicas de Mineração de dados educacional é necessário realizar algumas tarefas de pré-processamento para reorganizar e melhorar a qualidade dos dados, contribuindo assim para melhorar a precisão e a eficiência do processo de mineração subsequente (HAN; KAMBER; PEI, 2011).

5.1.1 Integração

A primeira tarefa de pré-processamento realizada foi a integração das bases de dados. A integração dos dados consiste em combinar dados de diferentes fontes para prover uma única fonte de dados. Para a modalidade de ensino a distância a integração foi entre o Sistema SIE e o AVA Moodle. Para o ensino presencial a integração ocorreu entre os sistemas SIE e Huxley ver Seção 4.1. A integração cuidadosa pode ajudar a reduzir e evitar redundâncias e inconsistências no conjunto de dados resultante. Isso pode ajudar a melhorar a precisão e a velocidade do processo de mineração de dados subsequente (HAN; KAMBER; PEI, 2011).

Para manter a integridade e confiabilidade entre os dados, um atributo obrigatório, com valor único e presente entre as bases de dados, deve ser identificado. Após um estudo entre as bases de dados, o atributo "CPF"[1] foi escolhido para fazer a unificação entre as bases de dados existentes para o ensino a distância e para o ensino presencial, uma vez que esse atributo permite a identificação único entre os estudantes selecionados.

[1] Cadastro de Pessoas Físicas

5.1.2 Limpeza

Visando ter dados mais consistentes e melhorar a eficácia dos algoritmos de predição, após a integração dos dados, foi realizada a etapa de limpeza dos dados (ver Seção 2.2). Esta etapa consiste em retirar os dados que possam distorcer a análise. Nesta etapa foi identificado e removido estudantes com dados incompletos, errados ou inconsistentes como por exemplos: estudante somente com o número da matricula sem qualquer outro tipo de informação, estudantes sem notas ou com nota negativa e dados nulos.

5.1.3 Transformação

Com objetivo de conseguir melhores resultados com os algoritmos de predição, diversos atributos sofreram alterações, sem sacrificar sua integridade. Essas transformações podem ser vistas abaixo:

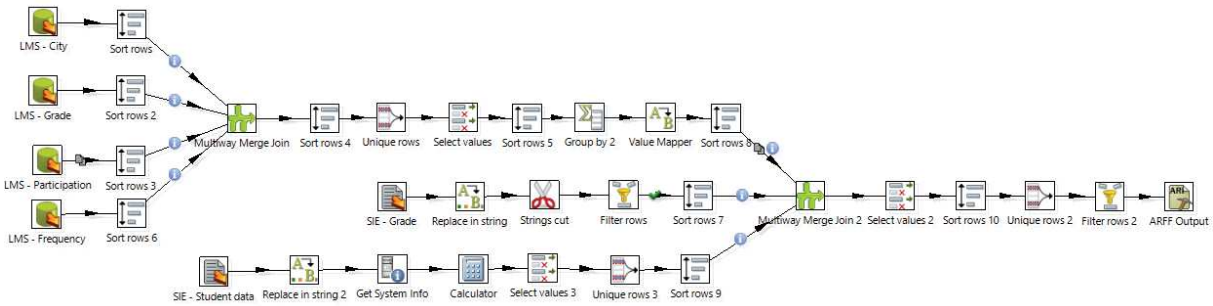
- Para fornecer uma visão mais concisa dos dados e evitar que alguns atributos influenciem negativamente nos resultados dos algoritmos diversos atributos foram transformados de numérico para nominal. Por exemplo, os valores numéricos das notas obtidas pelos estudantes em cada disciplina foram alterados para valores categóricos da seguinte forma: os estudantes que obtiveram notas nas disciplinas entre 9,0 e 10 receberam a letra "A"; ficaram com a nota "B" os estudantes que tiveram notas entre 7,0 e 8,99; Nota "C" entre 6,99 e 5,0; com nota "D" entre 4,99 e 3,0; e com nota "E" os estudantes que tiveram notas inferior a 3,0 ou nota nula.

- No atributo correspondente a cidade onde os estudantes residem foram encontradas algumas inconsistências. Por exemplo: existiam nomenclaturas diferentes para representar a mesma cidade, as instâncias de "Ouro Branco" e "Ouro Branco/AL" estão relacionadas com a mesma cidade. Este problema foi resolvido, com aplicação de técnicas de agrupamento de atributos.

- Também foram gerados novos atributos através de cálculos matemáticos. Por exemplo, o atributo "idade", que contém a idade dos estudantes, foi gerado através do calculo de um outro atributo que continha dia, mês e ano de nascimento dos estudantes.

Na Figura 10 são mostradas todas as etapas executadas no software Pentaho, durante a fase de pré-processamento dos dados, cada quadrado corresponde a uma tarefa de pré-processamento de dados realizada, começando pela extração dos dados até finalizar com a geração do arquivo compatível com o software de mineração Weka.

Figura 10 – Etapas do pré-processamento dos dados



Fonte: Elaborada pelo autor

5.1.4 Seleção dos atributos

Geralmente, os conjuntos de dados que passarão pelos processos de mineração de dados, apresentam dois problemas típicos (ROMERO et al., 2013). O primeiro diz respeito à elevada dimensionalidade dos dados, isto é, o número de atributos ou características torna-se extenso. No entanto, normalmente alguns desses atributos não irão ser significativos para a classificação e é provável que alguns desses atributos sejam correlacionados. O segundo problema, frequentemente encontrado nas bases de dados, são os dados desbalanceados, ou seja, quando o número de casos em uma classe é muito menor do que o número de casos em outra classe (ver Seção 2.2.2). Os métodos utilizados para resolver este segundo problema serão vistos na próxima Seção (5.1.5).

Para resolver o primeiro problema, torna-se recomendável a utilização de técnicas de seleção de atributos (ver Seção 2.2.2).

Quando se realiza a seleção de atributos, se espera que atributos irrelevantes e redundantes sejam removidos, o que normalmente leva um aumento da precisão do método de aprendizado. O *software* Weka dispõe de vários algoritmos de seleção de atributos, que utilizam diferentes métodos como *Filter*, *Wrapper* e *Embedded* (ver Seção 2.2.3). Durante a realização do trabalho foram testados vários algoritmos como: CfsSubsetEval,ChiSquaredAttributeEval, Consistency-SubsetEval, FilteredAttributeEval, OneRAttributeEval, FilteredSubsetEval, GainRatioAttributeEval, InfoGainAttributeEval, ReliefFAttributeEval, SymmetricalUncert-AttributeEval (WITTEN; FRANK; HALL, 2011). No entanto, o que apresentou o melhor resultado em termos de precisão foi o algoritmo *InfoGainAttributeEval* baseado no método *Filter* (ver Seção 2.2.2).

A maneira de se medir a qualidade de um atributo para a classificação é avaliar o seu grau de associação com a classe. Para determinar esse tipo de associação, foram propostas diversas métricas. Essas métricas são empregadas em uma base de dados $D(A_1, A_2, ...A_n, C), n \geq 1$, com n+1 atributos, onde C é o atributo classe e o seu domínio é $\{c_1, c_2, ...c_m\}$, $m \geq 2$. Assumindo que os valores dos atributos e da classe são

discretos.

A métrica utilizada no algoritmo *InfoGainAttributeEval* é o ganho de informação baseado no conceito de entropia (QUINLAN, 1986), que tem origem na área de teoria da informação. Dado um atributo A, cujo domínio é $\{a_1, a_2, ..., a_k\}, K \geq 1$, define-se a probabilidade $p_i, 1 \leq i \leq k$, de cada valor a_i do atributo como a razão entre o número de instância da base em que ocorre o valor a_i para o atributo A e o número total de instâncias. A entropia desse atributo, representada por H(A) é dada por:

$$H(A) = -\sum_{i=1}^{k}[p_i * log_2(p_i)], \tag{5.1}$$

A entropia da classe, representada por H(C), pode ser calculada da mesma forma, considerando p_j a razão entre o número de instâncias em que o valor c_j da classe, $1 \leq j \leq m$, ocorre na base e o número total de instâncias.

Seja a probabilidade $p_{j|i}$ a razão entre o número de instâncias da base que pertencem à classe c_j em que ocorre o valor a_i do atributo A, e o número total de instâncias da base. A entropia condicional da classe C, dado o atributo A, é calculada usando a fórmula:

$$H(C|A) = -\sum_{i=1}^{k}\sum_{j=1}^{m}[p_{j|i} * log(\frac{p_{j|i}}{p_i})] \tag{5.2}$$

Quanto mais informativo um atributo A for em relação à classe C, menor será a sua entropia condicional H(C|A). Pelas Fórmulas 5.1 e 5.2, verifica-se que a entropia de C dado A nunca será maior que a entropia de C apenas, pois o conhecimento do atributo A pode ser usado para determinar o valor de C. Logo, o ganho de informação de um atributo A em relação à classe pode ser calculado como a diferença entre a entropia da classe e a entropia condicional da classe dado o atributo A, indicado pela Fórmula 5.3:

$$GANHO(C|A) = H(C) - H(C|A) \tag{5.3}$$

Após o procedimento descrito acima, o conjunto de atributos foi reduzido de 29 para 16 atributos. Os atributos com seus respectivos pesos, ou seja, quantidade de informação que o atributo contribui para a classificação, ordenados em ordem decrescente do que contribui mais para a classificação até o que menos contribuiu para a classificação, pode ser visualizada no Gráfico 11. E o significado de cada atributo pode ser visto na Tabela 1.

Os mesmos procedimentos descritos acima foram realizados para a base de dados do ensino presencial. Os melhores atributos selecionados e ordenados em ordem decrescente, podem ser visto no Gráfico 12. O significado de cada atributo pode ser visto na Tabela 2.

Os atributos pessoais (Estado Civil, Sexo, Idade) foram usados com a intenção de verificar o quanto cada atributo contribui para a predição dos estudantes. No entanto, verificamos que esses atributos pouco contribuíam na classificação dos estudantes propensos ao insucesso. Porém, o custo para obter esses atributos foi pequeno, tornando viável para efeito de testes a inclusão destes atributos no modelo.

Tabela 1 – Atributos Selecionados na modalidade de ensino a distância

Atributos	Descrição
1ª Avaliação	Nota da primeira Avaliação
5ª Semana	Nota da quinta semana
2ª Semana	Nota da segunda semana
4ª Semana	Nota da quarta semana
3ª Semana	Nota da terceira semana
Blog	Quantidade de postagem e visualização no blog
1ª Semana	Nota da primeira semana
Forum	Quantidade de postagem e visualização no fórum
Acessos	Quantidade de acessos ao AVA
Assign	Quantidade de arquivos enviados e baixados
Cidade	Cidade
Message	Quantidade de mensagens enviadas
Wiki	Quantidade de mensagens enviadas
Sexo	Sexo do estudante
Estado	Estado Civil do estudante
Idade	Idade do estudante
Status	*Status da disciplina (Aprovado/Reprovado)*

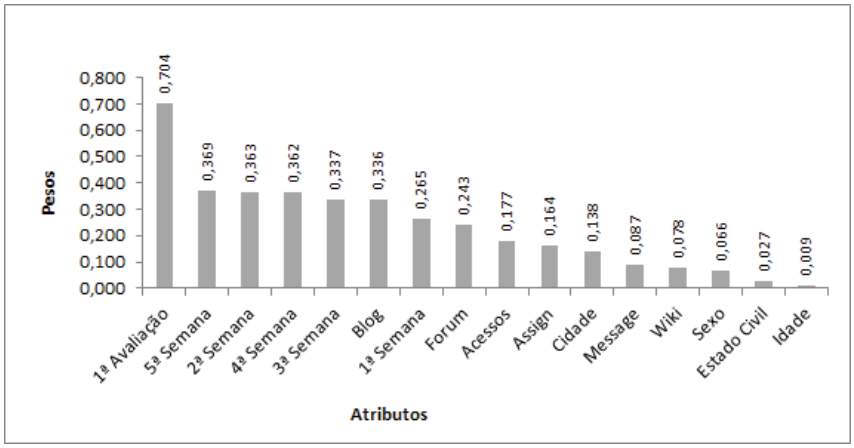
Fonte: Elaborada pelo autor

Tabela 2 – Atributos Selecionados na modalidade de ensino presencial

Atributos	Descrição
Problemas	Quantidade de Exercícios
1ª Avaliação	Nota da primeira Avaliação
Corretos	Total de Exercícios corretos
Submissoes	Total de submissões
2ª Semana	Nota da segunda semana
1ª Semana	Nota da primeira semana
3ª Semana	Nota da terceira semana
Estado Civil	Estado Civil
Sexo	Sexo
Idade	Idade
Status	*Status (Aprovado/Reprovado)*

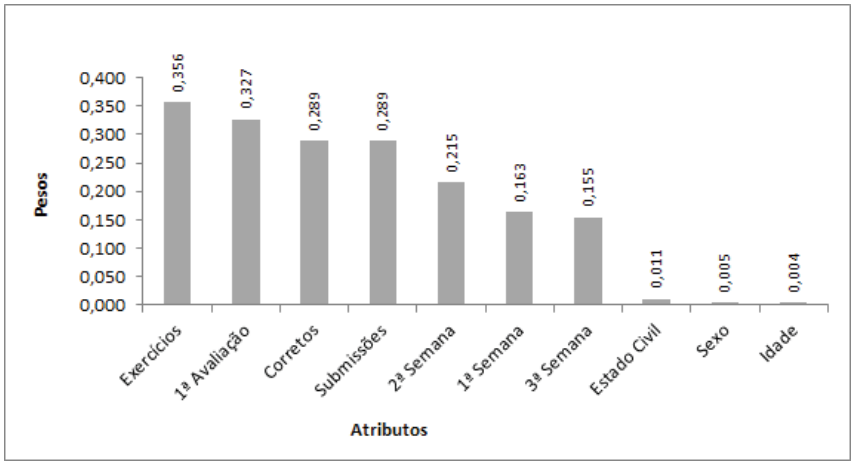
Fonte: Elaborada pelo autor

Figura 11 – Pesos e atributos do ensino a distância



Fonte: Elaborada pelo autor

Figura 12 – Pesos e atributos do ensino presencial



Fonte: Elaborada pelo autor

5.1.5 Balanceamento dos dados

Uma base de dados é dita desbalanceada, no domínio de classificação, quando o número de casos em uma classe é muito menor do que o número de casos em outra classe (GU et al., 2008). Estas classes com pouca representação são chamadas de classes raras (WEISS, 2004). Algoritmos de classificação são muitos sensíveis a este tipo de desbalanceamento e tendem a valorizar as classes predominantes e a ignorar as classes de menor representação (PHUA; ALAHAKOON; LEE, 2004). Os dois conjuntos de dados utilizados neste trabalho, apresentavam este problema, ou seja, à quantidade de estudantes reprovados eram bem menor que a quantidade de estudantes aprovados. Para realização desse procedimento foi utilizado o software Weka. Weka disponibiliza diversos algoritmos de balanceamento dos dados, no entanto, conforme explanado na Seção 2.2.2 a técnica SMOTE foi a mais apropriada para realização desta tarefa.

O balanceamento dos dados foi aplicado sobre o conjunto de dados da modalidade de

ensino presencial e a distância. Sendo utilizado posteriormente no conjunto de teste e de treinamento.

5.2 Execuções dos Algoritmos

Nesta seção, é descrito de que forma os algoritmos foram executados a fim de conseguir identificar os estudantes propensos ao insucesso.

Levando em consideração que, o objetivo é prever situação final do estudante com a maior antecedência possível, dentro da disciplina dada, executamos os algoritmos separadamente sobre cada modalidade de ensino enriquecendo os dados semanalmente. Ou seja, os algoritmos foram executados em etapas; na primeira etapa foram incluídos dados somente até o último dia da primeira semana de aula, na segunda etapa, foram acrescentados dados até o último dia da segunda semana de aula, e assim consequentemente até chegar à semana da primeira avaliação.

Além do enriquecimento dos dados, os algoritmos foram executados em três fases distintas:

1. Na primeira fase, todos os algoritmos foram executados sobre os conjuntos de dados com somente a integração de dados realizada sem a utilização de qualquer outra etapa de pré-processamento dos dados;

2. Na segunda fase, os algoritmos foram executados após as etapas de limpeza dos dados, transformação dos dados, seleção dos atributos e balanceamento de dados (ver Seção 5.1);

3. Finalmente, na última etapa os algoritmos foram executados após a realização dos ajustes finos.

Para realizar a avaliação da capacidade de generalização do modelo, todas as etapas foram avaliadas utilizando o método de validação cruzada k-fold, com k=10 (ver Seção 4.2.1). Para melhor caracterizar a eficácia dos algoritmos analisados neste trabalho, decidimos adotar a métrica *F-Measure* (ver Seção 4.2.2).

Para comparação dos resultados, foi utilizado a ferramenta *Weka Experiment Environment* (WEE). Este ambiente é apropriado para realizar comparações entre o desempenho de vários algoritmos de mineração de dados. O WEE permite ao usuário criar, executar e modificar o experimento de uma maneira mais conveniente, também permite selecionar um ou mais algoritmos disponíveis na ferramenta e analisar os resultados de modo a identificar se um classificador é, estatisticamente, melhor do que os demais (WEKA, 2014).

O WEE oferece três opções de estratificação da base de dados: i) *Crossvalidation* (*default*), ii) *Train/Test Percentage* Split (*data randomized*) e iii) *Train/Test Percentage*

Split (*order preserved*). Para obter significância estatística nos desempenhos dos algoritmos, o ambiente foi configurado com um número padrão de execuções. Por exemplo, cada algoritmo é executado 10 vezes e seu desempenho final é obtido a partir da média das execuções. No caso do 10-fold cross-validation significa que um classificador foi executado 100 vezes para os conjuntos de treinamento e teste.

5.2.1 Ajustes finos nos Algoritmos

O desempenho dos algoritmos de classificação, são sensíveis aos ajustes de parâmetros, principalmente em problemas do mundo real (VIANA et al., 2007). Os métodos de escolha destes parâmetros variam amplamente e são conhecidos como *tuning*. Com o objetivo de aumentar a eficácia dos algoritmos decidimos ajustar os parâmetros dos algoritmos utilizados neste trabalho.

(Naive Bayes) O ajuste fino do algoritmo Naive Bayes foi realizada seguindo a abordagem descrita em (JOHN; LANGLEY, 1995), que usa um método baseado na estimativa do *kernel* para realizar o ajuste fino.

(Árvore de decisão) De acordo com (WITTEN; FRANK; HALL, 2011), a eficácia do algoritmo J48 pode ser melhorada através da realização de um ajuste fino de dois parâmetros: (i) a quantidade de nós de folha; e (ii) a poda de árvore de decisão. Nós realizamos alguns experimentos comparativos, a fim de encontrar os melhores valores para esses parâmetros.

(Rede Neural) Realizamos os ajustes finos de três parâmetros do algoritmo Rede Neural: (i) a taxa de aprendizagem (*learning rate*); (ii) o impulso aplicado aos pesos durante a sua atualização (*momentum*); (iii) o número de camadas escondidas existentes na rede (*hidden layers*). De acordo com (WITTEN; FRANK; HALL, 2011), o ajuste fino destes parâmetros podem melhorar a eficácia do algoritmo de rede neural.

(Máquina de Vetor de Suporte) A primeira mudança nos parâmetros do algoritmo SVM foi em relação ao *Kernel*(ver Seção 2.3.4.3). Foram testadas 3 opções de Kernel: polinomial, RBF e Sigmóide, sendo que quem apresentou o melhor resultado foi o *Kernel* gaussiano (RBF). Posteriormente, foram alterados os parâmetros γ que pertencem ao kernel RBF e o coeficiente de custo "C".

No entanto, escolha manual dos parâmetros "C" e γ é indesejável, pois é imprecisa e não garante resultados de qualidade (IMBAULT; LEBART, 2004). Além de existir um número muito significativo de combinações entre os parâmetros "C" e γ que podem ser usadas para o SVM, de forma que, tentar encontrar a melhor configuração através de um processo exaustivo é bem difícil. Por isso, foi utilizado um método conhecido como *Grid-Search*. A técnica de *Grid-Search* é utilizada para buscar pelos

melhores parâmetros através da análise dos resultados obtidos com a execução do algoritmo para um intervalo de parâmetros. Pois, não se sabe previamente qual ou quais os melhores parâmetros para o problema em questão. Após o uso do método *Grid-Search* ajustamos os parâmetros manualmente com valores aproximados ao informado pelo método. Ou seja, o método *Grid-Search* serviu como heurística para os ajustes nos parâmetros do algoritmo SVM.

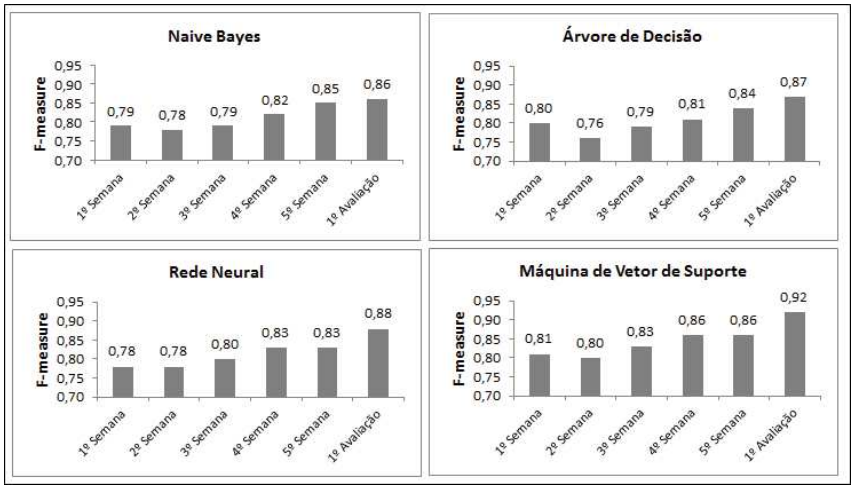
6 RESULTADOS E DISCUSSÕES

Nesta capítulo, apresentamos os principais resultados do experimento, descrito na Seção 5. Na Seção 6.2 respondemos às questões de pesquisa listados na Seção 1.3. E finalmente na Seção 6.3 discutimos algumas ameaças à validade do estudo. Mais detalhes dos resultados podem ser visto no apêndice A

6.1 Discussões dos Resultados

Como mencionado, anteriormente, as disciplinas dos cursos na modalidade a distância são divididas em oito semanas. Cada semana os estudantes recebem notas de acordo com as suas atividades realizadas e no fim da quinta semana é realizada a primeira avaliação. Os algoritmos foram executados em etapas; na primeira etapa foram incluídos dados somente até o último dia da primeira semana de aula, na segunda etapa, foram acrescentados dados até o último dia da segunda semana de aula, e assim consequentemente até chegar à semana da primeira avaliação. Após, a execução dos algoritmos de predição, foi constatado que os resultados foram melhorando a cada semana, como mostrado na Figura 13.

Figura 13 – Evolução dos resultados semanalmente ensino a distância

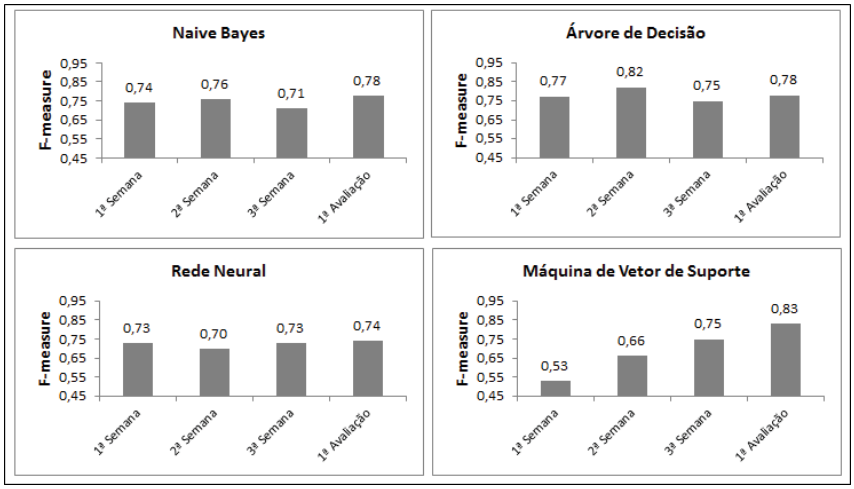


Fonte: Elaborada pelo autor

Na modalidade de ensino presencial com exceção do algoritmo de árvore de decisão, todos os outros algoritmos também obtiveram melhoras resultados com o enriquecimento dos dados como mostrado na Figura 14.

A fim de verificar a influencia dos dados sobre os resultados, incluímos dados após a primeira semana da avaliação. Com esses dados foi possível alcançar taxas de precisão de até 0,98% entre os algoritmos de predição na modalidades de ensino presencial e a

Figura 14 – Evolução dos resultados semanalmente ensino presencial



Fonte: Elaborada pelo autor

distância. Embora as taxas foram consideravelmente altas o objetivo de nosso estudo é detectar o insucesso o mais breve possível possibilitando uma intervenção pedagógica tempo hábil, portanto, esses dados foram descartados.

Foi aplicado sobre as fontes de dados o algoritmo de predição KNN (K vizinhos mais próximos)[1] conhecido como classificadores "preguiçosos" (AHA, 1997). Mesmo após as etapas de pré-processamento e ajustes finos, os resultados foram abaixo da média em comparação com os outros quatro algoritmos utilizados neste estudo, portanto, este algoritmo foi descartado. Mais detalhes dos resultados podem ser visto no apêndice A nas Figuras 21 e 22.

Também foram aplicados oito algoritmos de classificação "caixa-branca" (MARQUEZ-VERA; MORALES; SOTO, 2013), tais como regras de indução e árvores de decisão. Estes algoritmos foram aplicados após as as etapas de pré-processamento, a fim de identificar a eficácia dos algoritmos proposto por (MARQUEZ-VERA; MORALES; SOTO, 2013). Os resultados podem ser visualizados no apêndice A nas Figuras 23 e 24

6.2 Resposta as questões de pesquisa

Em seguida, vamos responder e discutir as seguintes questões de pesquisa:

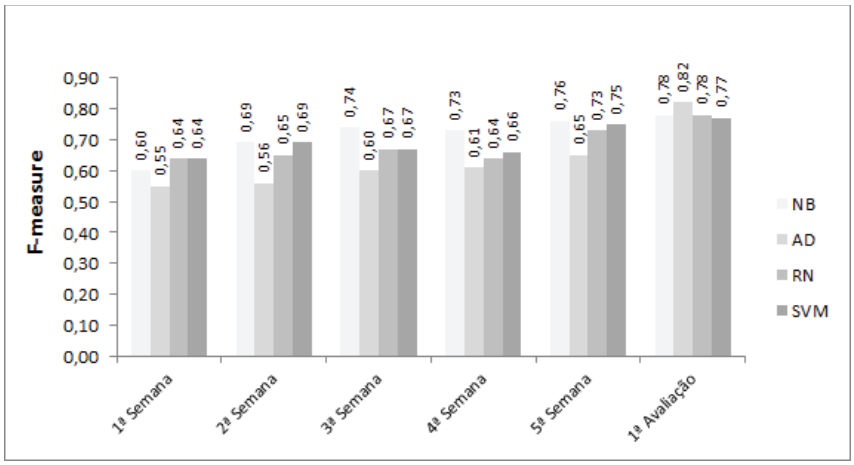
6.2.1 Qual a eficácia dos algoritmos de predição para identificar estudantes propensos ao insucesso?

A fim de responder a esta questão, executamos os algoritmos de predição, analisados neste estudo, sobre os conjuntos de dados: a distância e presencial. As Figuras 15 e 16 apresentam a eficácia (representada pela métrica *F-measure*) dos algoritmos para iden-

[1] do inglês *K Nearest Neighbors*

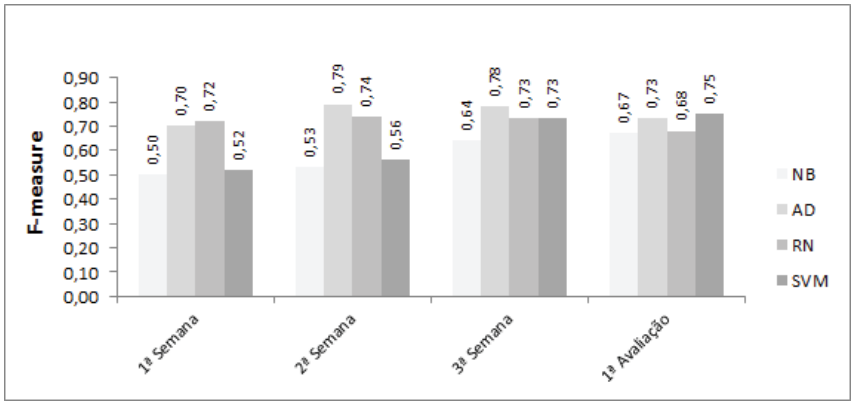
tificar precocemente os estudantes susceptíveis ao insucesso sobre os cursos a distância e presencial, respectivamente. Os algoritmos de previsão foram executados para cada semana de curso e após a aplicação da primeira avaliação.

Figura 15 – Eficácia dos algoritmos de predição do ensino a distância (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte)



Fonte: Elaborada pelo autor

Figura 16 – Eficácia dos algoritmos de predição do ensino presencial (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte)



Fonte: Elaborada pelo autor

Observamos que os algoritmos apresentaram uma eficácia que varia de 0,55 até 0,82 no curso a distância, e de 0,50 até 0,79 no curso presencial. Estes resultados indicam que, depois da primeira semana de curso os algoritmos são capazes de identificar pelo menos 0,50% de eficácia os estudantes propensos ao insucesso.

Observamos também que o algoritmo de árvore de decisão (*J48*), apresenta a melhor eficácia em ambas fontes de dados. Atingindo um *f-measure* com um valor igual a 0,82 após a aplicação da primeira avaliação no curso a distância, e 0,79 na segunda semana no curso presencial. Dado que os cursos a distância e presencial tem duração de 8 e 16 semanas, respectivamente. Podemos afirmar que o algoritmo de árvore de decisão é capaz de atingir uma eficácia igual a 82% quando os estudantes têm realizado pelo menos 60%

do curso online, e uma eficácia igual a 79% quanto os estudantes têm realizado pelo menos 25% do curso presencial.

Portanto, os resultados apresentam evidências de que os algoritmos de predição analisados neste estudo são eficazes para identificação precoce dos estudantes propensos ao insucesso. Apesar de não ser essencial para esse trabalho, é importante mencionar que o tempo de execução de cada algoritmo, nesta fase foi longo, chegando a ter algoritmos que demoraram mais de 45 minutos para apresentar os resultados. Na próxima seção, vamos mostrar que a eficácia desses algoritmos de classificação podem ser melhorados executando as etapas de pré-processamento de dados e ajustes finos nos algoritmos.

6.2.2 As etapas de pré-processamento de dados são realmente importantes e capazes de aumentar a eficácia dos algoritmos de predição?

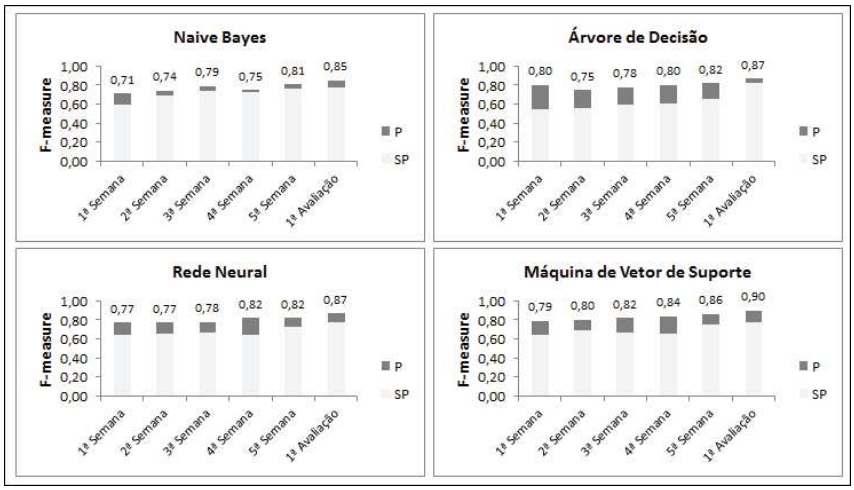
A fim de responder à segunda questão, foi realizado o pré-processamento nos dois conjuntos de dados utilizados neste trabalho, conforme descrito na Seção 5.1, em seguida, foram aplicados os algoritmos de predição sobre estes conjunto de dados pré-processados. Avaliamos a eficácia destes algoritmos sobre os conjuntos de dados pré-processados, e comparamos esses resultados com os obtidos pelos algoritmos quando aplicou-se sobre o conjunto de dados sem pré-processamento.

A Figura 17 apresenta uma comparação entre os resultados da eficácia dos quatro algoritmos (Naives Bayes, Árvore de decisão, Rede Neural e Máquina de Suporte de Vetor) aplicados sobre os dados do ensino a distância. Os resultados apresentados na Figura 17 indica que a eficácia de todos os algoritmos foram melhoradas quando aplicou-se o pré-processamento sobre o conjunto de dados do ensino a distância.

No entanto, foi necessário a utilização de um teste estatístico para verificar se essa melhoria é estatisticamente significante. Para esta verificação utilizamos o *t-test* (HAN; KAMBER; PEI, 2011). Ao aplicar o *t-test* sobre os resultados mostrados na Figura 17, obtivemos os seguintes *p-values*: (Naive Bayes) *p-values=0,1158*; (Árvore de decisão) *p-values=0,006349*; (Rede Neural) *p-values=0,002343*; e (SVM) *p-values=0,0005339*. De acordo com (HAN; KAMBER; PEI, 2011), para representar uma diferença estatisticamente significativa, normalmente, o *p-valor* deve ser inferior a *0,05*. Assim, podemos concluir que o *Naive Bayes* é o único algoritmo que não apresenta uma melhoria estatisticamente significativa quando aplicamos o pré-processamento sobre os dados da modalidade de ensino a distância.

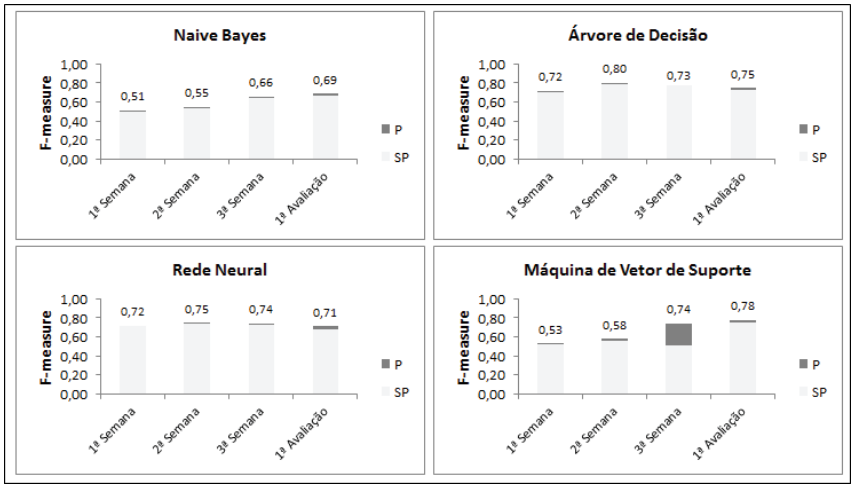
A Figura 18 apresenta uma comparação entre os resultados da eficácia dos algoritmos de predição aplicados sobre os dados da modalidade de ensino presencial. Primeiro foi aplicado os algoritmos sobre os dados sem o pré-processamento, em seguida, sobre os dados pré-processados. Com os resultados apresentados na Figura 18, não fica claro se a eficácia dos algoritmos foram melhoradas. Portanto aplicamos o teste estatístico *t-test* com o objetivo de verificar se a eficácia dos algoritmos tiveram melhoras.

Figura 17 – Comparativo dos resultados da eficácia dos algoritmos de predição do ensino a distância sem pré-processamento (SP) e com pré-processamento (P)



Fonte: Elaborada pelo autor

Figura 18 – Comparativo dos resultados da eficácia dos algoritmos de predição do ensino presencial sem pré-processamento (SP) e com pré-processamento (P)



Fonte: Elaborada pelo autor

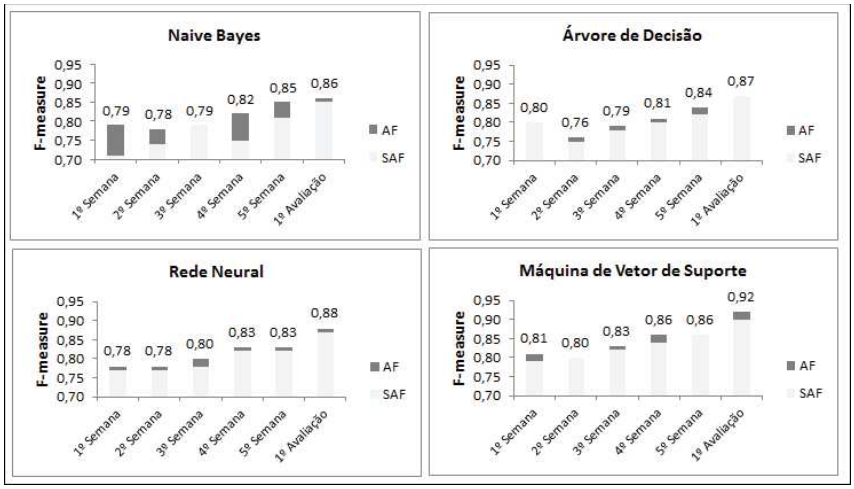
Ao aplicar o teste estatístico (*t-test*) sobre os resultados mostrados na Figura 18 obtivemos os seguintes *p-values*: (Naive Bayes) *p-values=0,7793*; (Árvore de decisão) *p-values=1*; (Rede Neural) *p-values=0,468*; e (SVM) *p-values=0,8422*. Notamos que os algoritmos não apresentam uma melhora estatisticamente significativa quando aplicou-se o pré-processamento no conjunto de dados presencial.

Com base nessa discussão, podemos concluir que o pré-processamento dos dados na modalidade de ensino a distância foi capaz de aumentar a eficácia da maioria dos algoritmos, mas o pré-processamento dos dados na modalidade de ensino presencial apesar de ter melhorado os resultados não impactou significativamente na eficácia dos algoritmos. Porém, observou-se também que, após a etapa de pré-processamento o tempo de execução dos algoritmos diminuíram consideravelmente nos dois conjuntos de dados.

6.2.3 A fase de ajustes finos dos algoritmos será capaz de aumentar ainda mais a eficácia dos algoritmos de predição?

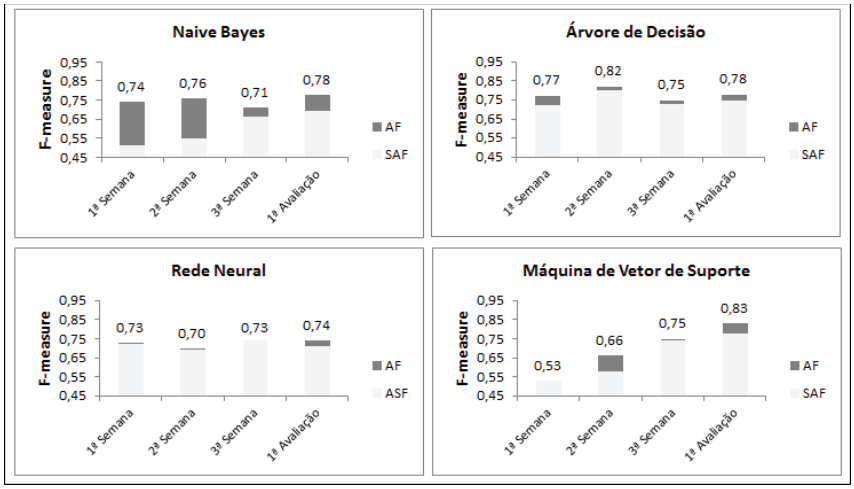
Estudos (GUNAWAN; LAU; LINDAWATI, 2011), (HUTTER et al., 2009) indicam que a eficácia de alguns algoritmos de predição pode ser melhorada após alguns ajustes em seus parâmetros, especialmente em problemas do mundo real. A fim de investigar tais evidências foi realizado os ajustes finos dos algoritmos, conforme descrito na Seção 5.2.1, em seguida, foram utilizados os conjuntos de dados pré-processados para comparar a efetividade dos algoritmos sem o ajuste fino e, em seguida, com os ajustes finos.

Figura 19 – Comparativo dos resultados da eficácia dos algoritmos de predição do ensino a distância sem ajustes finos (SAF) e com ajustes finos (AF)



Fonte: Elaborada pelo autor

Figura 20 – Comparativo dos resultados da eficácia dos algoritmos de predição do ensino presencial sem ajustes finos (SAF) e com ajustes finos (AF)



Fonte: Elaborada pelo autor

A Figura 19 apresenta os resultados da comparação da eficácia dos algoritmos quando aplicados sobre o conjunto de dados da modalidade de ensino a distância pré-processados

sem a realização dos ajustes finos e, em seguida, realizando os ajustes finos. Os resultados mostrados na Figura 19 indica que a eficácia dos algoritmos foram melhorados após a realização dos ajustes finos.

Ao aplicar o teste estatístico *t-test* sobre os resultados mostrados na Figura 19 obtivemos os seguintes *valores p*: (Naive Bayes) *p-value* = 0,01182; (Árvore de decisão) *p-value* = 0,005095; (Rede Neural) *p-value* = 0,001285; e (SVM) *p-value* = 0,0003295. Assim, podemos concluir que os ajustes finos melhorou a eficácia de todos os algoritmos, aplicando-as sobre o conjunto de dados da modalidade de ensino a distância pré-processados.

A Figura 20 apresenta os resultados da comparação da eficácia dos algoritmos quando aplicados sobre o conjunto de dados presencial. Os resultados apresentados na Figura 20 não esclarece se a eficácia de todos os algoritmos foi melhorada após os ajustes finos.

Ao aplicar o teste estatístico *t-test* sobre os resultados mostrados na Figura 20 obtivemos os seguintes *valores p*: (Naive Bayes) *p-value* = 0,02317; (Árvore de decisão) *p-value* = 0,2945; (Rede Neural) *p-value* = 0,6532; e (SVM) *p-value* = 0,03911. De acordo com estes *p-values*, só a eficácia dos algoritmos *Naive Bayes* e *SVM* tiveram uma melhora estatisticamente significativa.

Com base nos dados apresentados acima, podemos concluir que os ajustes finos dos algoritmos foi capaz de aumentar a eficácia de todos os algoritmos quando aplicado no conjunto de dados da modalidade de ensino a distância pré-processados, mas apenas dois algoritmos ajustados (Naive Bayes e SVM) tiveram um aumento na eficácia quando aplicado no conjunto de dados presencial pré-processados.

6.2.4 Depois de realizar o pré-processamento dos dados e os ajustes finos nos algoritmos de predição, quais das técnicas são mais eficazes na identificação dos estudantes propensos ao insucesso?

De acordo com os resultados apresentados na seção anterior, depois do pré-processamento dos dados e os ajustes finos nos algoritmos, foi possível identificar que uma variante do algoritmo de Máquina de Vetor de Suporte obteve os melhores resultados, em termos de eficácia, em ambas as fontes de dados chegando a um valor de *F-measure* igual a 0,92 e 0,83 após a primeira avaliação do curso na modalidade de ensino a distância e presencial, respectivamente.

Em outras palavras, uma variante do algoritmo de Máquina de Vetor de Suporte foi capaz de identificar com pelo menos 83% de eficácia os estudantes propensos ao insucesso na modalidade de ensino a distância ou presencial.

6.3 Limitações e Ameaças a validade do estudo

Apesar das considerações presentes no estudo apresentarem resultados positivos na predição de estudantes propensos ao insucesso, é importante observar algumas ameaças.

Propomos no nosso estudo a previsão de ações decorrentes de resoluções e decisões dos seres humanos. Assim, reconhecemos as limitações da metodologia e as possíveis falhas, dado que poderá ocorrer situações ou comportamentos que mesmo os estudantes estando suprindo todas as características definidas no estudo o estudante tome uma decisão inesperada que vá de encontro aos padrões definidos no estudo.

É importante salientar também que, no estudo, foram usados apenas os dados da disciplina introdutória a programação e de apenas uma única instituição de ensino. Portanto, as evidências obtidas a partir dos resultados do estudo não são gerais.

A maneira que foram realizados os ajustes finos nos algoritmos, com exceção do algoritmo máquina de vetor de suporte, também pode ser considera uma ameaça, pois, em todos os outros algoritmos os ajustes finos foram feitos de forma manual. Esta ameça é importante pois os ajustes finos realizados de forma manual não assegura os melhores resultados em termo de eficácia.

Foi adotada a medida *f-measure* como métrica para avaliar a eficácia dos algoritmos de predição. Embora esta métrica tenha sido amplamente utilizada pelos trabalhos de EDM existentes, outras métricas, como *Accuracy* e *Kappa* poderiam ser usadas.

No entanto, considerando os resultados, é evidente que o algoritmo SVM é um método poderoso e eficaz para predizer os grupos de estudantes susceptíveis ao insucesso nas disciplinas de introdutórias de programação.

7 CONSIDERAÇÕES FINAIS

7.1 Conclusão

Este trabalho se propôs a investigar quais seriam as técnicas de predição mais eficazes na identificação dos estudantes propensos ao insucesso, através de um estudo comparativo sobre o potencial e eficácia das quatro técnicas (árvore de decisão, máquina de vetor de suporte, rede neural e naive bayes) analisadas. Avaliou-se a eficácia das técnicas de predição em duas fontes de dados diferentes e independentes, uma na modalidade de ensino presencial e a outra na modalidade de ensino a distância sobre as disciplinas de programação introdutória.

Embora outros trabalhos apresentem formas promissoras para identificar estudantes que possam ter insucesso na disciplina introdutória de programação, eles são um pouco limitado em termos de predição do insucesso com precisão e com antecedência suficiente para permitir uma intervenção pedagógica em tempo hábil. Além disso, estas abordagens não investigam de forma clara e detalhada a influência do pré-processamento de dados e ajuste fino no algoritmos na eficácia das técnicas de EDM analisadas.

Diante do estudo realizado, as técnicas analisadas mostraram-se eficazes na identificação dos estudantes propensos ao insucesso no início da disciplina. E que após a realização das etapas de pré-processamento e ajustes finos todos algoritmos tiveram uma melhora em seus resultados. Ao fim do processo, o algoritmo máquina de vetor de suporte obteve os melhores resultados tanto na modalidade de ensino presencial quanto na modalidade a distância. Na modalidade a distância, o algoritmo alcançou uma taxa de *f-measure* de 92% com pelo menos 60% da disciplina realizada. Na modalidade presencial, o algoritmo alcançou uma taxa de *f-measure* de 83% com pelo menos 25% da disciplina realizada.

A relevância de um trabalho pode ser avaliada também pelas oportunidades de trabalhos futuros que o mesmo provê. Então, buscando obter tal importância, são propostos alguns direcionamentos para novas pesquisas que podem ser identificados a partir desse estudo.

Avaliar outras técnicas que não foram abordadas neste trabalho e verificar sua viabilidade para o problema. Para isso, poderiam ser utilizados, por exemplo, métodos de agrupamento ou mineração de relações. Utilizar outras métricas além da *f-measure* a fim de comparar a eficácia dos algoritmos de predição. As técnicas utilizadas neste trabalho poderiam ser aplicadas em diferentes disciplinas, cursos, instituições e modalidades de ensino, tornando assim, os resultados mais gerais. Outra questão que poderá ser investigada é o potencial de atingir melhores resultados usando novos fatores como: aspectos afetivos, profissionais, familiares, sociais e econômicos dos estudantes.

Finalmente, considerando os resultados, acredita-se que o algoritmo SVM é um mé-

todo eficaz para predizer grupos de estudantes susceptíveis ao insucesso na fase inicial das disciplinas de programação introdutórias. E que estas previsões podem ser muito úteis para os educadores, no sentido de realizar o mais breve possível, as intervenções pedagógicas necessárias para garantir o sucesso dos estudantes na disciplina.

REFERÊNCIAS

ABED. *Associação Brasileira de Educação a Distância*. 2014. <http://www.abed.org.br/site/pt/>. Acesso em novembro 2014.

ADRIAANS, P.; ZANTINGE, D. *Data Mining*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997. ISBN 0-201-40380-3.

aES, L. M. B. M.; CRUZ, S. M. S. da; aO, G. Z. Wave: An architecture for predicting dropout in undergraduate courses using edm. In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2014. (SAC 14), p. 243–247. ISBN 978-1-4503-2469-4. Disponível em: <http://doi.acm.org/10.1145/2554850.2555135>.

AHA, D. W. (Ed.). *Lazy Learning*. Norwell, MA, USA: Kluwer Academic Publishers, 1997. ISBN 0-7923-4584-3.

ARAQUE, F.; ROLDáN, C.; SALGUERO, A. Factors influencing university drop out rates. *Comput. Educ.*, Elsevier Science Ltd., Oxford, UK, UK, v. 53, n. 3, p. 563–574, nov. 2009. ISSN 0360-1315. Disponível em: <http://dx.doi.org/10.1016/j.compedu.2009.03.013>.

ARORA, Y.; SINGHAL, A.; BANSAL, A. Prediction & warning: A method to improve student's performance. *SIGSOFT Softw. Eng. Notes*, ACM, New York, NY, USA, v. 39, n. 1, p. 1–5, fev. 2014. ISSN 0163-5948. Disponível em: <http://doi.acm.org/10.1145/2557833.2557842>.

AZEVEDO, F. M. *Redes neurais com aplicações em controle e em sistemas especialistas*. 1st. ed. [S.l.]: Visual Books, 2000. ISBN 8575020056.

BAKER, R. S.; INVENTADO, P. S. Educational Data Mining and Learning Analytics. In: LARUSSON, J. A.; WHITE, B. (Ed.). *Learning Analytics*. Springer New York, 2014. p. 61–75. Disponível em: <http://dx.doi.org/10.1007/978-1-4614-3305-7_4>.

BARANAUSKAS, J. A.; MONARD, M. C. *Reviewing Some Machine Learning Concepts and Methods*. São Carlos - SP, 2000. Disponível em: <ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_102.pdf>.

BASHEER, I.; HAJMEER, M. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, v. 43, n. 1, p. 3 – 31, 2000. ISSN 0167-7012. Neural Computting in Micrbiology. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167701200002013>.

BASILI, V.; CALDIERA, G.; ROMBACH, D. H. *The goal question metric approach*. [S.l.]: Wiley, 1994.

BAYER, J. et al. Predicting drop-out from social behaviour of students. In: *Proceedings of the 5th International Conference on Educational Data Mining - EDM 2012*. Greece: [s.n.], 2012. p. 103–109. ISBN 978-1-74210-276-4.

BENNEDSEN, J.; CASPERSEN, M. E. Failure rates in introductory programming. *SIGCSE Bull.*, ACM, New York, NY, USA, v. 39, n. 2, p. 32–36, jun. 2007. ISSN 0097-8418. Disponível em: <http://doi.acm.org/10.1145/1272848.1272879>.

BIGUS, J. P. *Data Mining with Neural Networks: Solving Business Problems from Application Development to Decision Support*. Hightstown, NJ, USA: McGraw-Hill, Inc., 1996. ISBN 0-07-005779-6.

BOSWELL, D. Introduction to support vector machines. 2002.

CARUANA, R.; NICULESCU-MIZIL, A. An empirical comparison of supervised learning algorithms. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: ACM, 2006. (ICML '06), p. 161–168. ISBN 1-59593-383-2. Disponível em: <http://doi.acm.org/10.1145/1143844.1143865>.

CHAPELLE, O. et al. Choosing multiple parameters for support vector machines. *Machine Learning*, Kluwer Academic Publishers, v. 46, n. 1-3, p. 131–159, 2002. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A%3A1012450327387>.

CHAWLA, N. V. et al. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, AI Access Foundation, USA, v. 16, n. 1, p. 321–357, jun. 2002. ISSN 1076-9757. Disponível em: <http://dl.acm.org/citation.cfm?id=1622407.1622416>.

COSTA, E. et al. Mineração de dados educacionais: Conceitos, técnicas,ferramentas e aplicações. *Jornada de Atualização em Informática na Educação - JAIE*, v. 02, n. 02, p. 03, 2012. ISSN 23167734. Disponível em: <http://www.br-ie.org/pub/index.php/pie/article/view/2341/2096>.

CRONE, S. F.; LESSMANN, S.; STAHLBOCK, R. The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. *European Journal of Operational Research*, v. 173, n. 3, p. 781 – 800, 2006. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221705006739>.

ER, E. Identifying at-risk students using machine learning techniques: A case study with is 100. In: *International Journal of Machine Learning and Computing*. Singapore: IACSIT Press, 2012. p. 476–481. ISBN 978-1-4503-2469-4. Disponível em: <http://doi.acm.org/10.1145/2554850.2555135>.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Advances in knowledge discovery and data mining. In: FAYYAD, U. M. et al. (Ed.). Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996. cap. From Data Mining to Knowledge Discovery: An Overview, p. 1–34. ISBN 0-262-56097-6. Disponível em: <http://dl.acm.org/citation.cfm/id=257938.257942>.

GIBERT, K.; MARRE, M. S.; CODINA, V. Choosing the right data mining technique: Classification of methods and intelligent recommendation. In: *2010 International Congress on Environmental Modelling and Software*. [S.l.: s.n.], 2010.

GOEBEL, M.; GRUENWALD, L. A survey of data mining and knowledge discovery software tools. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 1, n. 1, p. 20–33, jun. 1999. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/846170.846172>.

GU, Q. et al. Data mining on imbalanced data sets. In: *Advanced Computer Theory and Engineering, 2008. ICACTE '08. International Conference on*. [S.l.: s.n.], 2008. p. 1020–1024.

GUNAWAN, A.; LAU, H.; LINDAWATI. Fine tuning algorithm parameters using the design of experiments approach. In: COELLO, C. (Ed.). *Learning and Intelligent Optimization*. Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6683). p. 278–292. ISBN 978-3-642-25565-6. Disponível em: <http://dx.doi.org/10.1007/978-3-642-25566-3_21>.

HALL, M. A. Correlation-based feature selection for discrete and numeric class machine learning. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000. (ICML '00), p. 359–366. ISBN 1-55860-707-2. Disponível em: <http://dl.acm.org/citation.cfm?id=645529.657793>.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123814790, 9780123814791.

HANKS, B. et al. Program quality with pair programming in cs1. *SIGCSE Bull.*, ACM, New York, NY, USA, v. 36, n. 3, p. 176–180, jun. 2004. ISSN 0097-8418. Disponível em: <http://doi.acm.org/10.1145/1026487.1008043>.

HARRISON, T. H. *Intranet Data Warehouse*. [S.l.]: Berkeley, 1998.

HAYKIN, S. *Redes Neurais, Princípios e prática*. 2. ed. [S.l.]: Bookman, 2000.

HU, X. Db-hreduction: A data preprocessing algorithm for data mining applications. *Applied Mathematics Letters*, v. 16, n. 6, p. 889 – 895, 2003. ISSN 0893-9659. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0893965903900139>.

HUTTER, F. et al. Paramils: An automatic algorithm configuration framework. *J. Artif. Int. Res.*, AI Access Foundation, USA, v. 36, n. 1, p. 267–306, set. 2009. ISSN 1076-9757. Disponível em: <http://dl.acm.org/citation.cfm?id=1734953.1734959>.

HUXLEY. *Huxley - THEHUXLEY*. 2015. <http://www.thehuxley.com/huxley//>. Accessed March 2015.

IEPSEN, E.; BERCHT, M.; REATEGUI, E. Detection and assistance to students who show frustration in learning of algorithms. In: *Frontiers in Education Conference, 2013 IEEE*. [S.l.: s.n.], 2013. p. 1183–1189. ISSN 0190-5848.

IMBAULT, F.; LEBART, K. A stochastic optimization approach for parameter tuning of support vector machines. In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. [S.l.: s.n.], 2004. v. 4, p. 597–600 Vol.4. ISSN 1051-4651.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (UAI'95), p. 338–345. ISBN 1-55860-385-9. Disponível em: <http://dl.acm.org/citation.cfm?id=2074158.2074196>.

KERNIGHAN, B. W. *The C Programming Language*. 2nd. ed. [S.l.]: Prentice Hall Professional Technical Reference, 1988. ISBN 0131103709.

KOHAVI, R.; JOHN, G. H. Automatic parameter selection by minimizing estimated error. In: *In Proceedings of the Twelfth International Conference on Machine Learning*. [S.l.]: Morgan Kaufmann, 1995. p. 304–312.

KOLLER, D.; SAHAMI, M. Toward optimal feature selection. In: *In 13th International Conference on Machine Learning*. [S.l.: s.n.], 1996. p. 284–292.

KOTSIANTIS, S. B.; KANELLOPOULOS, D.; PINTELAS, P. E. *Data preprocessing for supervised learning*. 2006.

LAROSE, D. T. *Discovering Knowledge in Data: An Introduction to Data Mining*. [S.l.]: Wiley-Interscience, 2004. ISBN 0471666572.

LOBO, M. B. C. M. Panorama da evasão no ensino superior brasileiro: Aspectos gerais das causas e soluções. In: *Instituto Lobo para Desenvolvimento da Educação, da Ciência e da Tecnologia*. [S.l.: s.n.], 2013.

LYKOURENTZOU, I. et al. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Comput. Educ.*, Elsevier Science Ltd., Oxford, UK, UK, v. 53, n. 3, p. 950–965, nov. 2009. ISSN 0360-1315. Disponível em: <http://dx.doi.org/10.1016/j.compedu.2009.05.010>.

MARQUEZ-VERA, C.; MORALES, C.; SOTO, S. Predicting school failure and dropout by using data mining techniques. *Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de*, v. 8, n. 1, p. 7–14, Feb 2013. ISSN 1932-8540.

MARTINHO, C. R. V.; NUNES, C.; R., M. C. Prediction of school dropout risk group using neural network. *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, 2013.

MARTINHO, V.; NUNES, C.; MINUSSI, C. Prediction of school dropout risk group using neural network. In: *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*. [S.l.: s.n.], 2013. p. 111–114.

MARTINS, A. Insucesso académico na universidade: Causas e estratégias para a sua minimização. *Docencia Universitaria*, v. 5, n. 1 y 2, 2013. Disponível em: <http://saber.ucv.ve/ojs/index.php/rev_docu/article/view/4665>.

MITCHELL, T. *The discipline of machine learning*. Carnegie Mellon University, Pittsburgh: [s.n.], 2006.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: _____. *Sistemas Inteligentes - Fundamentos e Aplicações*. [S.l.]: Manole, 2003. cap. 4, p. 89–114.

MOODLE. *Moodle - Modular Object-Oriented Dynamic Learning Environment*. 2014. <https://moodle.org/>. Acesso em maio 2014.

OLSON, D. L.; DELEN, D. *Advanced Data Mining Techniques*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2008. ISBN 3540769161, 9783540769163.

PAURA, L.; ARHIPOVA, I. Cause analysis of students' dropout rate in higher education study program. *2nd World Conference On Business, Economics And Management-WCBEM 2013*, p. 1282–1286, 2013.

PENTAHO. *Pentaho - Pentaho Data Integration*. 2015. <http://www.pentaho.com/>. Acesso em Janeiro 2015.

PHUA, C.; ALAHAKOON, D.; LEE, V. Minority report in fraud detection: Classification of skewed data. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 6, n. 1, p. 50–59, jun. 2004. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1007730.1007738>.

PYTHON. *Python*. 2015. <https://www.python.org/>. Accessed May 2014.

QUINLAN, J. R. Induction of decision trees. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 1, n. 1, p. 81–106, mar. 1986. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1022643204877>.

ROMERO, C. et al. Predicting students' final performance from participation in on-line discussion forums. *Comput. Educ.*, Elsevier Science Ltd., Oxford, UK, UK, v. 68, p. 458–472, out. 2013. ISSN 0360-1315. Disponível em: <http://dx.doi.org/10.1016/j.compedu.2013.06.009>.

ROMERO, C.; VENTURA, S. Educational data mining: A review of the state of the art. *Trans. Sys. Man Cyber Part C*, IEEE Press, Piscataway, NJ, USA, v. 40, n. 6, p. 601–618, nov. 2010. ISSN 1094-6977. Disponível em: <http://dx.doi.org/10.1109/TSMCC.2010.2053532>.

RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, 2002. Hardcover. ISBN 0137903952. Disponível em: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0137903952>.

RYAN, B.; SEIJI, I.; CARVALHO, A. Mineração de dados educacionais: Oportunidades para o brasil. *Revista Brasileira de Informática na Educação*, v. 19, n. 02, p. 03, 2011. ISSN 1414-5685. Disponível em: <http://www.br-ie.org/pub/index.php/rbie/article/view/1301>.

TAN, P.-H.; TING, C.-Y.; LING, S.-W. Learning difficulties in programming courses: Undergraduates' perspective and perception. In: *Computer Technology and Development, 2009. ICCTD '09. International Conference on*. [S.l.: s.n.], 2009. v. 1, p. 42–46.

UFAL. *Universidade Federal de Alagoas*. 2014. <http://www.ufal.edu.br/cied/nucleo-de-tutoria/estrutura-do-nucleo>. Acesso em novembro 2014.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.

VIANA, R. et al. Svm with stochastic parameter selection for bovine leather defect classification. In: MERY, D.; RUEDA, L. (Ed.). *Advances in Image and Video Technology*. [S.l.]: Springer Berlin Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4872). p. 600–612. ISBN 978-3-540-77128-9.

WATSON, C.; LI, F.; GODWIN, J. Predicting performance in an introductory programming course by logging and analyzing student programming behavior. In: *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*. [S.l.: s.n.], 2013. p. 319–323.

WATSON, C.; LI, F. W. Failure rates in introductory programming revisited. In: *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*. New York, NY, USA: ACM, 2014. (ITiCSE '14), p. 39–44. ISBN 978-1-4503-2833-3. Disponível em: <http://doi.acm.org/10.1145/2591708.2591749>.

WEISS, G. M. Mining with rarity: A unifying framework. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 6, n. 1, p. 7–19, jun. 2004. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1007730.1007734>.

WEKA. *Weka - The University of Waikato*. 2014. <http://www.cs.waikato.ac.nz/ml/weka/>. Acesso em Novembro 2014.

WITTEN, I. H.; FRANK, E.; HALL, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123748569, 9780123748560.

WU, X. et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, Springer-Verlag, v. 14, n. 1, p. 1–37, 2008. ISSN 0219-1377. Disponível em: <http://dx.doi.org/10.1007/s10115-007-0114-2>.

ZAKI, M. J.; JR, W. M. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. New York, NY, USA: Cambridge University Press, 2014. ISBN 0521766338, 9780521766333.

ZHANG, S.; ZHANG, C.; YANG, Q. Data preparation for data mining. *Applied Artificial Intelligence*, v. 17, n. 5-6, p. 375–381, 2003. Disponível em: <http://dx.doi.org/10.1080/713827180>.

APÊNDICE A – RESULTADOS DETALHADOS

A.1 Resultados

Tabela 3 – Resultados detalhados dos algoritmos de predição do ensino a distância (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte)

	1ª semana				2ª semana				3ª semana			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM	NB	AD	RN	SVM
Acurácia	77.15	72.03	77.55	78.79	80.59	72.62	77.95	79.99	82.57	76.37	78.65	78.43
Precision	0.70	0.61	0.70	0.72	0.73	0.61	0.69	0.72	0.75	0.69	0.70	0.69
Recall	0.56	0.55	0.63	0.63	0.69	0.57	0.65	0.70	0.77	0.57	0.68	0.69
F-measure	0.60	0.55	0.64	0.64	0.69	0.56	0.65	0.69	0.74	0.60	0.67	0.67
	4ª semana				5ª semana				1ª avaliação			
	NB	AD	RN	SVM	NB	AD	RN	SVM	NB	AD	RN	SVM
Acurácia	81.64	75.67	76.92	77.86	83.53	80.98	82.73	83.99	84.40	87.41	86.86	86.64
Precision	0.72	0.67	0.66	0.68	0.75	0.79	0.77	0.79	0.75	0.80	0.85	0.87
Recall	0.78	0.62	0.66	0.67	0.79	0.59 *	0.73	0.75	0.83	0.88	0.77	0.74
F-measure	0.73	0.61	0.64	0.66	0.76	0.65	0.73	0.75	0.78	0.82	0.78	0.77

Fonte: Elaborada pelo autor

Tabela 4 – Resultados detalhados dos algoritmos de predição do ensino a distância após o pré-processamento dos dados (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte)

	1ª semana				2ª semana				3ª semana			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	74.28	80.12	77.11	77.81	75.50	74.81	77.58	79.20	78.83	76.45	78.01	81.13
Precision	0.81	0.80	0.78	0.76	0.78	0.74	0.79	0.77	0.79	0.74	0.79	0.80
Recall	0.66	0.82	0.78	0.83	0.72	0.79	0.77	0.84	0.80	0.83	0.79	0.86
F-measure	0.71	0.80	0.77	0.79	0.74	0.75	0.77	0.80	0.79	0.78	0.78	0.82
	4ª semana				5ª semana				1ª Avaliação			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	77.70	79.65	81.71	83.74	82.25	81.51	81.84	85.38	86.09	87.51	87.36	89.51
Precision	0.83	0.79	0.81	0.82	0.87	0.81	0.81	0.85	0.90	0.88	0.87	0.88
Recall	0.71	0.83	0.84	0.88	0.77	0.84	0.85	0.88	0.82	0.89	0.89	0.93
F-measure	0.75	0.80	0.82	0.84	0.81	0.82	0.82	0.86	0.85	0.87	0.87	0.90

Fonte: Elaborada pelo autor

Tabela 5 – Resultados detalhados dos algoritmos de predição do ensino a distância após ajustes finos (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte)

	1ª semana				2ª semana				3ª semana			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	79.18	80.12	77.96	80.76	77.74	75.50	77.71	79.20	77.68	76.49	79.53	82.16
Precision	0.80	0.80	0.78	0.80	0.77	0.75	0.79	0.77	0.76	0.75	0.80	0.80
Recall	0.79	0.82	0.80	0.84	0.80	0.79	0.78	0.84	0.83	0.81	0.81	0.87
F-measure	0.79	0.80	0.78	0.81	0.78	0.76	0.78	0.80	0.79	0.79	0.80	0.83
	4ª semana				5ª semana				1ª Avaliação			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	81.83	80.75	83.02	85.60	84.97	83.62	82.56	86.52	86.12	87.45	87.82	91.78
Precision	0.82	0.80	0.83	0.84	0.86	0.84	0.82	0.87	0.88	0.88	0.88	0.89
Recall	0.82	0.85	0.84	0.90	0.84	0.85	0.85	0.87	0.85	0.88	0.89	0.96
F-measure	0.82	0.81	0.83	0.86	0.85	0.84	0.83	0.86	0.86	0.87	0.88	0.92

Fonte: Elaborada pelo autor

Tabela 6 – Resultados detalhados dos algoritmos de predição do ensino presencial (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte)

	1º Exercício				2º Exercício			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	57.17	74.5	68.33	71	59.67	73	70.17	65.5
Precision	0.5	0.75	0.73	0.62	0.53	0.76	0.78	0.6
Recall	0.55	0.7	0.76	0.49	0.58	0.87	0.75	0.53
F-measure	0.5	0.7	0.72	0.52	0.53	0.79	0.74	0.56

	3º Exercício				1ª Avaliação			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	61	73.17	71	77	70.33	73.5	70.5	79.83
Precision	0.69	0.76	0.77	0.76	0.76	0.78	0.82	0.81
Recall	0.64	0.73	0.74 v	0.71	0.65	0.74	0.64	0.71
F-measure	0.64	0.78	0.73	0.73	0.67	0.73	0.68	0.75

Fonte: Elaborada pelo autor

Tabela 7 – Resultados detalhados dos algoritmos de predição do ensino presencial após pre-processamento dos dados (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte)

	1º Exercício				2º Exercício			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	57.42	76.75	78.00	72.56	60.51	84.04	76.15	74.72
Precision	0.51	0.78	0.79	0.82	0.55	0.88	0.79	0.85
Recall	0.56	0.72	0.72	0.42	0.58	0.78	0.75	0.48
F-measure	0.51	0.72	0.72	0.53	0.55	0.80	0.75	0.58

	3º Exercício				1ª Avaliação			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	61.68	79.56	79.00	82.18	76.76	81.15	78.74	83.67
Precision	0.7	0.82	0.79	0.93	0.78	0.83	0.82	0.88
Recall	0.74	0.72	0.75	0.62	0.68	0.74	0.67	0.73
F-measure	0.66	0.73	0.74	0.74	0.69	0.75	0.71	0.78

Fonte: Elaborada pelo autor

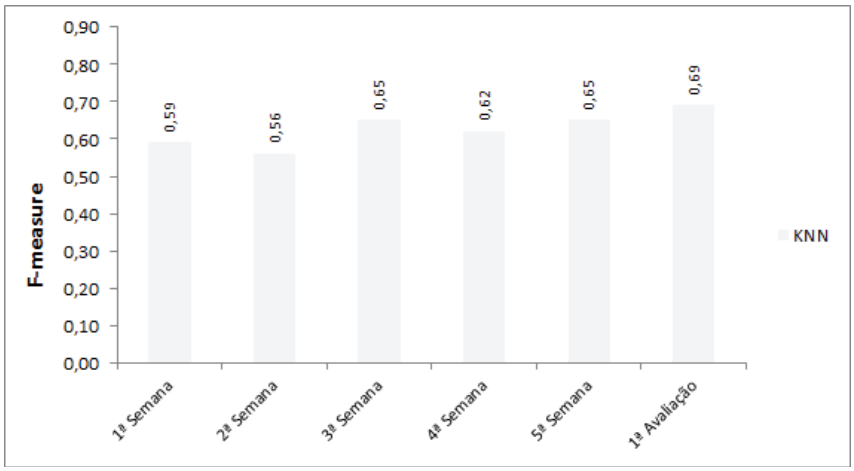
Tabela 8 – Resultados detalhados dos algoritmos de predição do ensino presencial após ajustes finos (NB - Naive Bayes; AD - Árvore de Decisão; RN - Rede Neural; SVM - Máquina de Vetor de Suporte)

	1º Exercício				2º Exercício			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	72.85	79.82	78.24	72.68	79.17	86.51	0.76	77,49
Precision	0.67	0.79	0.78	0,83	0.77	0.91	0.77	0.88
Recall	0.89	0.80	0.74	0.42	0.81	0.78	0.69	0.56
F-measure	0.74	0.77	0.73	0.53	0.76	0.82	0.70	0.66

	3º Exercício				1ª Avaliação			
Algoritmos	NB	AD	RN	SVM	NB	AD	RN	SVM
Accuracy	78.64	81.57	79.13	84.69	84.17	83.19	81.38	88.04
Precision	0.82	0.85	0.80	0.96	0.89	0.87	0.88	0.94
Recall	0.69	0.72	0.74	0.62	0.74	0.75	0.68	0.78
F-measure	0.71	0.75	0.73	0.75	0.78	0.78	0.74	0.83

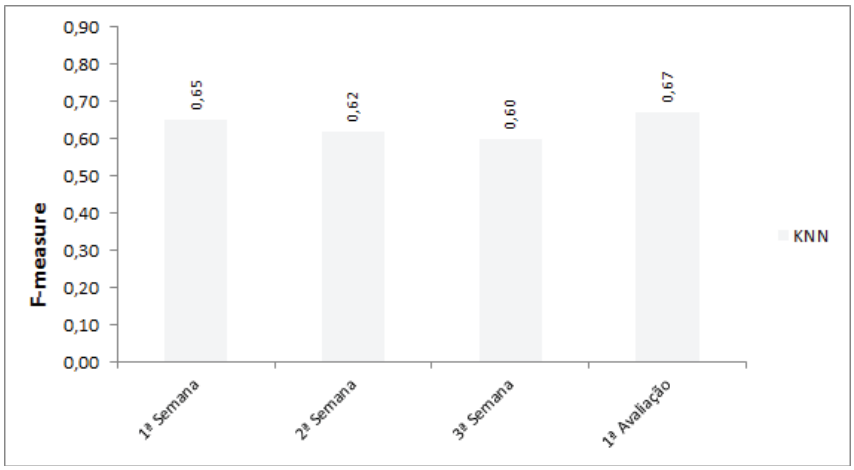
Fonte: Elaborada pelo autor

Figura 21 – Eficácia do algoritmo de predição KNN após pré-processamento e ajustes finos do ensino a distância (KNN - K vizinhos mais próximos)



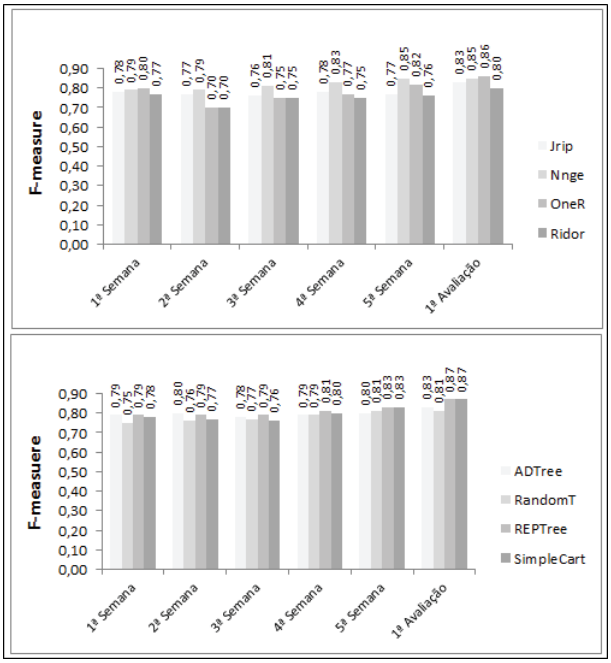
Fonte: Elaborada pelo autor

Figura 22 – Eficácia do algoritmo de predição KNN após pré-processamento e ajustes finos do ensino presencial (KNN - K vizinhos mais próximos)



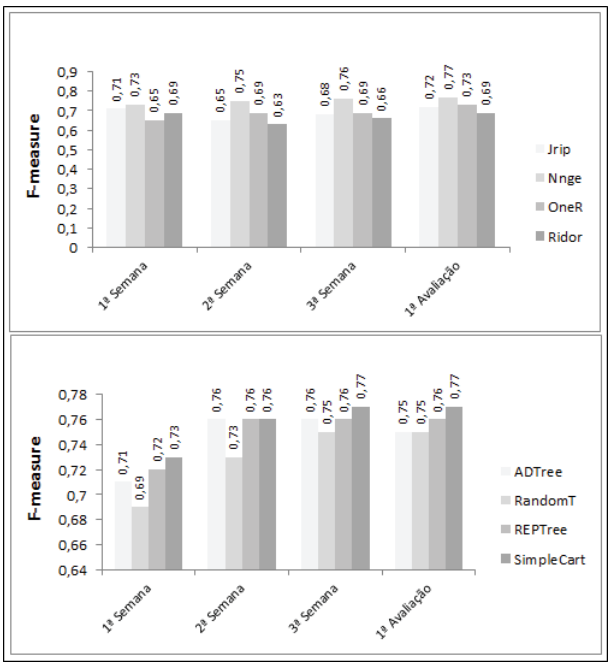
Fonte: Elaborada pelo autor

Figura 23 – Eficácia dos algoritmos de predição de caixa branca após pré-processamento do ensino a distância



Fonte: Elaborada pelo autor

Figura 24 – Eficácia dos algoritmos de predição de caixa branca após pré-processamento do ensino presencial



Fonte: Elaborada pelo autor