UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM
INSTITUTO DE COMPUTAÇÃO - ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

MINERAÇÃO DE DADOS EDUCACIONAIS: PREVISÃO DE NOTAS PARCIAIS UTILIZANDO CLASSIFICAÇÃO

Marília Maria Bastos de Araújo Cavalcanti Feitoza Fava de Sousa

Manaus - AM
2017

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S725m

Sousa, Marília Maria Bastos de Araújo Cavalcanti Feitoza Fava de
Mineração de Dados Educacionais: previsão de notas parciais utilizando classificação. / Marília Maria Bastos de Araújo Cavalcanti Feitoza Fava de Sousa. 2017
84 f.: il. color; 31 cm.

Orientadora: Fabíola Guerra Nakamura
Coorientadora: David Braga Fernandes de Oliveira
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Mineração de Dados Educacionais. 2. Previsão. 3. Classificação. 4. Ensino de Programação. I. Nakamura, Fabíola Guerra II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

"Mineração de Dados Educacionais: Previsão de Notas Parciais Utilizando Classificação"

MARÍLIA MARIA BASTOS DE ARAÚJO CAVALCANTI FEITOZA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Profa. Fabiola Guerra Nakamura - PRESIDENTE

Profa. Elaine Harada Teixeira de Oliveira - MEMBRO INTERNO

Prof. David Braga Fernandes de Oliveira - MEMBRO EXTERNO

Prof. Leandro Silva Galvão de Carvalho - MEMBRO EXTERNO

Manaus, 29 de Setembro de 2017

Marília Maria Bastos de Araújo Cavalcanti Feitoza Fava de Sousa

Mineração de Dados Educacionais: Previsão de Notas Parciais Utilizando Classificação

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de mestre em Informática.

Orientadora: Dra. Fabíola Guerra Nakamura

Universidade Federal do Amazonas – UFAM
Instituto de Computação – IComp

Manaus - AM
2017

Dedico este trabalho ao meu esposo, meu sol e estrelas.

Agradecimentos

O tempo passou rápido, como de costume, quase quatro anos de aprendizado, de mudanças e de superação. Uma vida pessoal emaranhada a uma vida acadêmica, com fortes dependências, difíceis de serem dissociadas e que me fizeram vacilar inúmeras vezes. Foram duas tromboses, três *stents*, quatro cirurgias, um zumbido no ouvido enlouquecedor e tantas outras coisas. Meu corpo mudou rápido, minha mente não tanto, mas sempre tive o que e a quem agradecer, principalmente com a conclusão deste trabalho. Essa conclusão, representa um novo começo, onde tenho mais experiência, resiliência e compaixão. "O vento que faz naufragar é o que me faz mover".

Agradeço ao meu esposo, Vitri, por me ajudar a "remar" desde o início. Ele me apoiou na decisão de começar o mestrado e se manteve ao meu lado, apesar de todas as adversidades. Sempre me incentivando, me alegrando e me amando, quando eu mesma já não era capaz. Vitri, eu te amo! Muito obrigada! (Em Manaus, somos somente eu, ele e nosso "bulldoguinho" Rodney McKay).

Aos professores da minha graduação, Vládia Pinheiro, Pedro Porfírio, Adbeel, Júlio Tôrres e Sandra Freitas pelo incentivo e pelas cartas de recomendação enviadas ao IComp.

Ao professor Alberto Castro, por sempre acreditar que eu seria capaz de concluir o mestrado. A primeira trombose ocorreu no início do processo de inscrição e mesmo assim, tive seu apoio. Obrigada também por todos os ensinamentos!

Aos professores Eduardo Feitosa, Fabíola Nakamura e David Oliveira, por serem as estrelas que me guiaram nesta dissertação, sempre cheios de disposição a ajudar e orientar.

Ao meu amigo e companheiro de estudo Caio Gregoratto, pelo apoio e amizade, também além do mestrado.

Aos meus pais, Geraldo e Isabel; aos meus sogros, Gener e Márcia; aos meus avós, Plínio e Nenenzinha; às minhas tias, Sandra e Núbia; às minhas irmãs, Jordanna e Evelinne; ao meu primo Renan, aos meus amigos, Marília, Camilla e Gato, a distância não os impediu de se tornarem presentes e de me acompanharem durante todos esses anos. Vocês foram e são muito importantes para mim!

A Deus, por todas as pessoas maravilhosas que são colocadas em minha vida.

Serenidade para aceitar o que não posso mudar, coragem para mudar o que posso e sabedoria para perceber a diferença.

Oração da Serenidade

Mineração de Dados Educacionais: Previsão de Notas Parciais Utilizando Classificação

Autor: Marília Maria Bastos de Araújo Cavalcanti Feitoza Fava de Sousa
Orientadora: Dra. Fabíola Guerra Nakamura

RESUMO

O presente trabalho tem o intuito de apresentar a Mineração de Dados Educacionais e um experimento envolvendo previsão de provas parciais. O experimento é realizado através dos dados da disciplina de Introdução à Programação de Computadores da Universidade Federal do Amazonas e busca classificar os alunos de acordo com as notas obtidas, em no máximo três classes: satisfatório, insatisfatório e sem conceito (alunos evadidos). Como conclusão, tem-se uma análise quantitativa com os dados da previsão.

Palavras-chave: Mineração de Dados Educacionais, Previsão, Classificação, Ensino de Programação.

Educational Data Mining: Predicting Partial Notes Using Classification

Author: Marília Maria Bastos de Araújo Cavalcanti Feitoza Fava de Sousa
Advisor: Dra. Fabíola Guerra Nakamura

ABSTRACT

The present work introduces the Educational Data Mining and an experiment involving prediction of partial exams. The experiment uses data of the Introduction to Computer Programming course of the Federal University of Amazonas and seeks to classify the students according to their grade, in a maximum of three classes: satisfactory, unsatisfactory and without concept (dropout students). As conclusion, there is a quantitative analysis with the predictive data.

Keywords: Educational Data Mining, Prediction, Classification, Introductory Programming Teaching.

Lista de Figuras

Lista de Quadros

Lista de Tabelas

Lista de Abreviaturas e Siglas

AVA	Ambiente Virtual de Aprendizagem
CART	*Classification and Regression Trees*
CBIE	Congresso Brasileiro de Informática na Educação
CHC	Cattell-Horn-Carroll
EAD	Educação a Distância
EDM	Mineração de Dados Educacionais
IComp	Instituto de Computação
IPC	Introdução a Programação de Computadores
JEDM	*Journal of Educational Data Mining*
LA	*Learning Analytics*
L	*Logistic*
MIL	*Multiple Instance Learning*
RBIE	Revista Brasileira de Informática na Educação
RF	*Random Forest*
ROC	*Receiver Operating Characteristic Curve*
SAEB	Sistema de Avaliação da Educação Básica
SBIE	Simpósio Brasileiro de Informática na Educação
SGBD	Sistemas de Gerenciamento de Bancos de Dados
SMO	*Sequential Minimal Optimization*
SVM	*Support Vector Machine*
TISE	Conferência Internacional sobre Informática na Educação
UFAM	Universidade Federal do Amazonas
Weka	*Waikato Environment for Knowledge Analysis*
WIE	*Workshop* de Informática na Escola

Sumário

1 Introdução

Ambientes Virtuais de Aprendizagens (AVA) são comumente utilizados para o acompanhamento dos alunos e realização de atividades, estando presentes em diversas universidades independentemente da metodologia adotada: presencial, semipresencial ou a distância. Juízes Online também já estão auxiliando disciplinas que envolvem programação, disponibilizando mais atividades aos alunos e realizando a correção automática, com *feedback* em tempo real. Essas duas tecnologias apoiam várias disciplinas da Universidade Federal do Amazonas (UFAM) e elas estão presentes através de duas plataformas:

- ***ColabWeb***: é um AVA personalizado a partir do Moodle[1]. Ele dispõe de fóruns para debates, *quizzes* para a realização de atividades com *feedback* automático e um *log* que guarda internamente todas as ações realizadas pelos seus usuários [Castro & Fuks 2009].
- ***CodeMeistre***: é um Juiz Online criado por professores da UFAM para a realização de atividades de programação. Ele suporta seis linguagens (C, C++, Python, Java, SQL e Haskell), possui diversos problemas de programação pré-cadastrados e possui um mecanismo de gamificação. Assim como o *ColabWeb*, ele também guarda um *log* com as ações dos usuários [Carvalho et al. 2016].

O uso do *ColabWeb* e do *CodeMeistre* tornou a disciplina de Introdução à Programação de Computadores (IPC) parcialmente virtual. O conteúdo das aulas não se encontra mais no caderno dos alunos ou nos computadores dos professores, atividades e avaliações não são mais impressas em papéis e as correções não são mais realizadas por professores.

Com essa virtualização[2] a disciplina de IPC, tornou-se convidativa para a aplicação da Mineração de Dados Educacionais (EDM). Com ela, é possível ter uma maior compreensão dos processos de ensino, aprendizagem e motivação dos alunos, tanto em ambientes individuais quanto em ambientes colaborativos de ensino.

---

[1] É um software livre de AVA e pode ser acessado em https://moodle.org.

[2] De acordo com filósofo francês Michel Serres [Serres 2013] temos que a disciplina de IPC se estabelece plenamente na atual revolução mundial. A primeira, foi a transição da linguagem oral para a escrita. A segunda, da escrita para a impressa. A terceira e atual, da impressa para o virtual. Serres não consegue prever as mudanças que a terceira revolução irá proporcionar, mas é interessante saber que ele acredita que a verdadeira transformação ocorrerá na pedagogia!

Esta dissertação tenta compreender o processo de aprendizagem através das notas parciais com a busca de evidências que demonstrem um aprendizado satisfatório do aluno e pretende dar suporte à aprendizagem através do desenvolvimento de um modelo de previsão. A previsão é realizada através da classificação dos alunos de acordo com um conceito: satisfatório, insatisfatório e sem conceito (ausência de informação de nota, possível evasão). Como conclusão, há uma análise quantitativa através do modelo desenvolvido.

1.1 Motivação

A disciplina de IPC na UFAM é ministrada para quatorze cursos de engenharia e ciências exatas. Ela é ofertada durante o primeiro semestre de alguns desses cursos e apresenta elevados índices de reprovação. Em média, 50% dos alunos são reprovados por semestre.

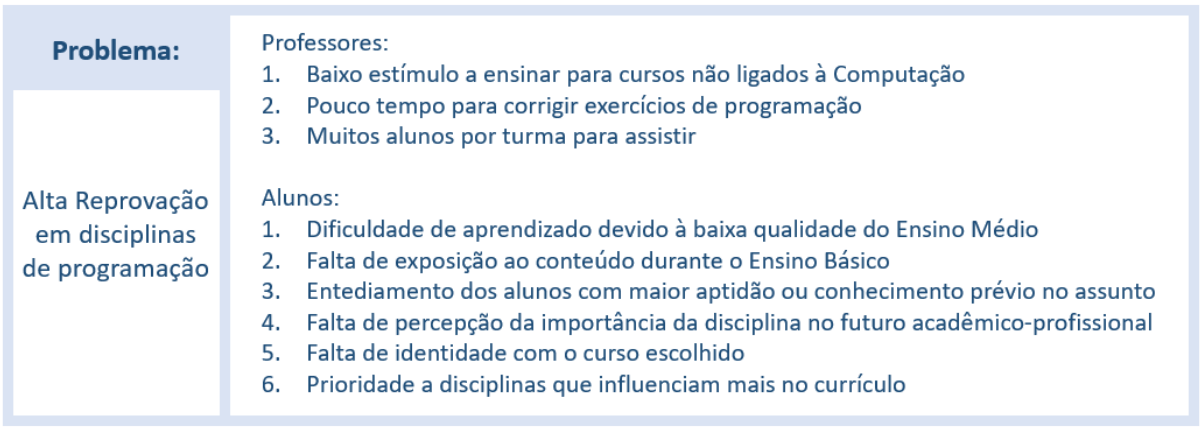


Figura 1. A alta reprovação em disciplinas de programação. Adaptada de [Carvalho et al. 2016].

De acordo com alguns pesquisadores [Chaves et al. 2103, Paes et al. 2013, Pelz et al. 2012, Píccolo et al. 2010] há diversos motivos gerais que podem desencadear o problema da reprovação em disciplinas introdutórias à programação, eles foram resumidos por Carvalho et al. (2016) e podem ser vistos na Figura 1. Quanto a IPC, alguns desses motivos são minimizados:

- Todas as aulas e exercícios são construídos de maneira colaborativa entre os professores e todas as turmas recebem o mesmo conteúdo;
- O uso de ferramentas como o *ColabWeb* e o *CodeMeistre* facilitam o trabalho do professor quanto ao número de alunos a serem assistidos, através da correção automática;
- Tais ferramentas tornam possível a disponibilização de mais atividades para os

alunos, o que melhora a aprendizagem através da prática;

- A gamificação presente no *CodeMeistre* pode estimular os alunos entediados a resolverem exercícios propostos pelos professores;
- O ensino híbrido dá autonomia aos alunos para conciliar a disciplina de IPC com outras de maior interesse, uma vez que a presença não é obrigatória nas aulas destinadas aos exercícios práticos.

Apesar desses tópicos apresentarem boas estratégias adotadas para a disciplina de IPC, a reprovação ainda se mantém elevada e a cada semestre os professores buscam motivos que a justifiquem. Por isso, a motivação desta dissertação é identificar antecipadamente os alunos com chances de resultados insatisfatórios em cada prova parcial. Dessa forma, professores e tutores podem direcionar seus esforços aos discentes com dificuldade, aumentando as chances de eles terem um bom rendimento em uma prova seguinte e consequentemente, diminuindo as taxas de reprovação.

1.2 Problemática

Um dos grandes problemas em pesquisas educacionais é caracterizado pela composição dos dados, que tendem a ser utilizados em vários níveis hierárquicos e não possuem independência estatística [Baker et al. 2011] (Seção 2.1). Por esse motivo, ocorreu a dissociação da Mineração de Dados em Mineração de Dados Educacionais (EDM).

Na EDM, os problemas empíricos geralmente podem ser decompostos em três tipos [Junker 2011]:

- Realização de inferências sobre as características de atividades. Exemplo: a atividade consegue medir o que queremos? A medição é significativa? Para quais alunos?
- Realização de inferências sobre as características dos alunos. Exemplo: quais habilidades, proficiências ou outros componentes de conhecimento eles possuem? Uma intervenção pode melhorar o desempenho?
- Previsões sobre o desempenho dos alunos em tarefas futuras. Exemplo: tema desta dissertação.

Junker (2011) afirma que, infelizmente, o progresso em qualquer um desses problemas empíricos é dificultado por duas fontes de dependência estatística:

1. Diferentes ações de um estudante podem não fornecer informações

independentes sobre ele. Por exemplo, se esse estudante calcular corretamente a área de um retângulo através dos seus dois lados, podemos obter pouca informação adicional e independente sobre uma outra tarefa, como a multiplicação (que está inclusa no cálculo da área).

2. Pode existir dependência entre múltiplas respostas (dos mesmos alunos ou diferentes), devido a contextos comuns - cognitivos, sociais ou institucionais. Dois tipos de dependência podem ser combinados hierarquicamente para a construção de modelos de dados educacionais, e cada tipo de dependência deve ser explicada ao fazer uma inferência na EDM.

Previsões que não explicam as dependências tendem a produzir previsões muito otimistas, difíceis de defender e menos propensas a generalizar. Por isso, a natureza dos dados educacionais ainda é tratada como um dos grandes desafios da EDM.

Quanto a esta dissertação, o problema mais preocupante está relacionado à fraca dependência das diferentes ações de um estudante com as suas notas parciais. Por exemplo, na amostra analisada, não é possível utilizar apenas as notas obtidas em exercícios práticos para prever a nota parcial. Para realizar uma boa previsão, é necessário agregar mais evidências além das notas dos exercícios (mesmo que algumas delas possuam dependência entre si) e verificar qual composição de evidências pode ser mais significativa para obter uma melhor acurácia.

1.3 Objetivos

1.3.1 Objetivo Geral

Criar um modelo de previsão de notas parciais para disciplina de IPC, a partir de evidências de exercícios de múltipla escolha, codificação e conhecimento prévio.

1.3.2 Objetivos Específicos

- Apresentar a Mineração de Dados Educacionais;
- Buscar evidências que estejam relacionadas ao processo de aprendizagem dos alunos.

1.4 Contribuição

Esta dissertação mostra que a criação de um modelo de previsão através da classificação de alunos pode ser realizada sem grande esforço e que a partir das análises realizadas é possível:

- Disponibilizar o modelo desenvolvido para alertar os professores sobre a aprendizagem dos alunos e realizar possíveis mediações;
- Conseguir verificar se a mediação foi benéfica através da alteração da previsão;
- Embasar as alterações das estratégias didáticas adotadas para disciplina[3];
- Identificar evidências que contribuem para o desempenho dos alunos;
- Dar origem a trabalhos futuros que apresentem uma melhor acurácia na previsão, ao continuar o estudo sobre a identificação das evidências ou através da otimização das abordagens/algoritmos.

1.5 Organização do trabalho

Este trabalho está organizado em seis capítulos:

- **Capítulo 1 - Introdução:** tem-se a introdução, motivação, problemática, objetivos, contribuição e organização do trabalho.
- **Capítulo 2 - Mineração de Dados:** explana o que é Mineração de Dados como uma forma introdutória para explicar o que é a Mineração de Dados Educacionais (EDM). Sendo então dividido em quatro seções. A primeira sobre EDM com suas principais aplicações, processo de descoberta do conhecimento educacional e métodos de mineração, categorizados pela EDM. A segunda apresenta o *Learning Analytics* e suas diferenças com a EDM. A terceira seção explica como ocorre a previsão. A quarta e última seção fala sobre as ferramentas existentes para apoiar a análise de dados.
- **Capítulo 3 - Trabalhos Relacionados:** traz os trabalhos correlatos com o tema desta dissertação, além de preocupar-se em apresentar os principais trabalhos e livros sobre EDM.
- **Capítulo 4 - Metodologia:** aborda a metodologia em que a pesquisa foi desenvolvida. Apresenta as evidências que tratam do desempenho do aluno e os algoritmos utilizados.
- **Capítulo 5 - Experimento:** trata especificamente do tema central desta dissertação. Sendo relatado o cenário de seu contexto, a previsão realizada através dos algoritmos apresentados no capítulo 4 e os resultados de cada um

[3] Seiji Isotani (informação verbal), afirma que novas técnicas/aplicativos/ferramentas criadas com o objetivo de aprimorar a educação, geralmente não apresentam um estudo que comprovam um melhor desempenho dos alunos, com o seu uso. (Palestra realizada na UFAM, em março de 2017).

dos algoritmos.

- **Capítulo 6 - Considerações Finais:** é realizado as considerações finais do trabalho.

2 Mineração de Dados

É possível imaginar a quantidade de dados existentes no mundo e que a criação deles não é uma invenção da computação, mas com ela é possível criá-los automaticamente e acessá-los de uma maneira mais prática. Toda empresa quer conhecer o seu cliente, e para isso cria cadastros com os dados pessoais dele e pode até guardar informações do que ele comprou. Os médicos também fazem isso: mantêm um registro de todas as consultas de seus pacientes. As escolas guardam as notas dos alunos, os pagamentos das mensalidades e várias informações pessoais. Esses são alguns exemplos de dados e bases de dados que existem há muito tempo.

Hoje, com os computadores, *tablets*, celulares e principalmente com a internet, é possível coletar muitos dados e de forma indireta. Temos as redes sociais, que podem traçar um perfil de acordo com as postagens de um usuário; lojas virtuais, que percebem o interesse dos clientes em um produto a partir de um simples clique; cursos a distância, que conseguem saber se o aluno assistiu ao vídeo proposto sem pausá-lo ou adiantá-lo. Sim, todas essas pessoas estão sendo vigiadas e muitas vezes nem percebem.

É importante saber que dado é a menor unidade obtida em uma observação do mundo. Informação é o uso dos dados dentro de um contexto que possa lhes dar um significado e auxiliar na tomada de uma decisão. Já o conhecimento é o uso da informação sendo aplicada, é a informação com um propósito ou uma utilidade (Figura 2).

Figura 2. Metáfora sobre dado, informação, apresentação e conhecimento [EpicGraphic].

A Mineração de Dados[4], como o nome sugere, é o processo ou a tarefa de encontrar ouro ou até mesmo pedras preciosas. Quer dizer, descobrir padrões possivelmente desconhecidos em dados/informações e conseguir extrair informação/conhecimento. Ela combina métodos tradicionais de análise de dados com algoritmos sofisticados para processamento de grandes volumes de dados. Os métodos tradicionais possuíam alguns desafios que motivaram o desenvolvimento da Mineração de Dados [Tan et al. 2009]:

- Necessidade de escalabilidade para trabalhar com grandes quantidades de dados;
- Alta dimensionalidade dos dados;
- Dados heterogêneos e complexos que precisavam considerar a relação que possuem entre si;
- Análise em dados distribuídos;
- Realização de análise não-tradicional[5], ou seja, gerar e avaliar hipóteses com os dados já existentes.

De forma geral, a Mineração de Dados de acordo com Tan et al. (2009), baseia-se em ideias tais como:

1. Amostragem, estimativa e teste de hipóteses a partir de Estatística;
2. Algoritmos de busca, técnicas de modelagem, teorias de aprendizagem usadas em Inteligência Artificial, reconhecimento de padrões e Aprendizagem de Máquina.

Figura 3. Áreas relacionadas à Mineração de Dados, adaptada de [Tan et al. 2009].

[4] Mineração de Dados também é conhecida como "Descoberta de Conhecimento em Banco de Dados", do inglês, *Knowledge Discovery in Databases*. Mas alguns estudiosos veem a Mineração de Dados simplesmente como um passo essencial no processo de descoberta de conhecimento [Han et al. 2011].

[5] Na abordagem estatística tradicional, é proposto uma hipótese e em seguida é realizado um experimento com o objetivo de reunir os dados. Em seguida, os dados são analisados com respeito a essa hipótese.

A Mineração de Dados também se relaciona com diversas áreas como mostra a Figura 3. É possível perceber que as suas aplicações possuem um caráter multidisciplinar, podendo ser utilizada em diversos campos: educação, marketing, antropologia, judiciário, saúde, economia, política, ações contra o terrorismo, etc.

Geralmente, as aplicações podem ser divididas entre [Tan et al. 2009]:

- **Tarefas preditivas**: prevê o valor de um atributo específico com base nos valores dos demais atributos. O atributo a ser previsto é normalmente conhecido como alvo, variável preditiva ou dependente, enquanto os demais, como variáveis preditoras, explicativas ou independentes (tema desta dissertação).
- **Tarefas descritivas**: deriva padrões (associações, correlações, tendências, agrupamentos, trajetórias e anomalias) que resumem as relações subjacentes aos dados. Ou seja, consiste na identificação de características intrínsecas do conjunto de dados, definidas por padrões compreensíveis por humanos. As tarefas descritivas são de natureza exploratória e frequentemente necessitam de técnicas de pós-processamento para validar e explicar os resultados.

Um passo necessário para a realização da Mineração de Dados é o pré-processamento nos dados. Han et al. (2011) definiu alguns passos:

- **Seleção dos dados**: recupera os dados relevantes ao domínio desejado.
- **Limpeza dos dados**: remove o ruído e dados inconsistentes.
- **Enriquecimento dos dados**: complementa os dados existentes com outros de diferentes bancos de dados.
- **Transformação dos dados**: codifica os dados de acordo com as entradas dos métodos utilizados na mineração.

Após o pré-processamento, os métodos de Mineração de Dados poderão ser aplicados. Os padrões descobertos pelos métodos são extraídos e pode haver a necessidade da aplicação de um pós-processamento [Han et al. 2011], onde ocorrerá:

- **Avaliação dos padrões encontrados**: verifica os padrões que são realmente interessantes para a descoberta de conhecimento.
- **Apresentação do conhecimento**: utiliza técnicas de visualização e representação de conhecimento para apresentar o conhecimento extraído.

Han et al. (2011) considera a Mineração de Dados como parte de um processo iterativo, que ocorre entre as etapas de pré-processamento e pós-processamento, podendo ser realizada através da interação com o usuário ou com uma base de dados. A Figura 4 exemplifica o processo descrito.

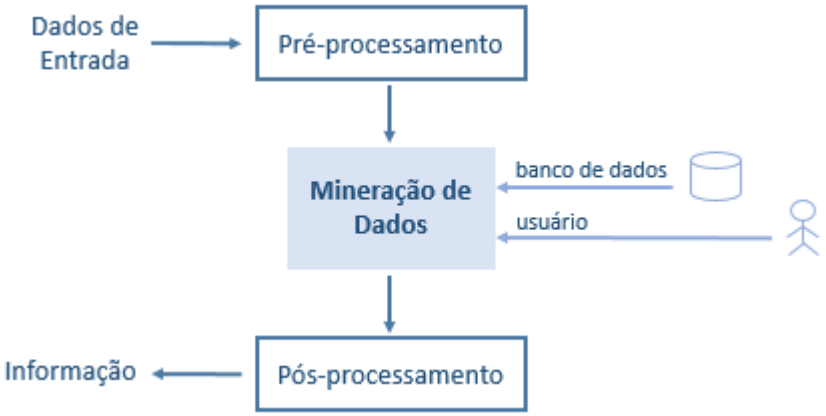


Figura 4. Mineração de dados: processo de descoberta de conhecimento.

A área da Mineração de Dados surge para continuar a descobrir padrões e gerar conhecimento quando o homem não possui a capacidade de processar a enorme quantidade de dados existente nos dias atuais; e principalmente com a possibilidade de gerar hipóteses de forma automática a partir desses dados. Pesquisas científicas, envolvendo genética e mudanças climáticas, por exemplo, têm sido evoluídas com a ajuda da Mineração de Dados. A competitividade em várias áreas também foi alterada, tornou-se mais acirrada e, por isso, às vezes é difícil encontrar casos práticos diferentes dos já conhecidos, como o caso "da cerveja e da fralda"[6]. Albert Einstein certa vez falou que "a única fonte de conhecimento é a experiência", hoje, sabe-se que a Mineração de Dados também pode ser uma fonte de conhecimento.

2.1 Mineração de Dados Educacionais

A partir dos métodos tradicionais de análise de dados, a Mineração de Dados abriu oportunidades excitantes para explorar/analisar novos tipos de dados e para analisar antigos tipos de dados de novas maneiras. Assim, a Mineração de Dados Educacionais surge com o recente aumento dos cursos a distância e do suporte computacional aos cursos presenciais. Os

[6] Uma das maiores redes de varejo dos Estados Unidos descobriu, através dos dados que tinham armazenados, que à venda de fraldas descartáveis estava associada à venda de cerveja. Percebeu que os compradores eram homens que saiam a noite para comprar fraldas e aproveitavam para levar latas de cerveja. Com a descoberta, os produtos foram colocados um ao lado do outro e as vendas aumentaram!

pesquisadores da área da Educação descobriram que os métodos da Mineração de Dados precisam ser constantemente modificados para atender às particularidades dos dados educacionais [Baker 2010; Romero & Ventura 2013]:

- Há uma variedade de contextos educacionais, onde os dados podem, por exemplo, ser organizados em termos de estrutura do material de aprendizagem (habilidades, problemas, unidades, aulas) e da estrutura de contexto de aprendizagem (alunos, professores, pares de colaboração, classes e escolas) [Costa et al. 2012].
- A variedade de níveis hierárquicos dos dados sugere que a informação pode estar entre vários níveis. Por exemplo, em uma análise de como um aluno utiliza um software educacional, pode ser interessante considerar simultaneamente os níveis de digitação, de resposta, de sessão, dos alunos, das salas de aula e da escola. Além das questões sobre o tempo, sequência e contexto[7].
- Há uma constante falta de independência estatística entre os dados educacionais, provavelmente pela variedade de níveis hierárquicos. A estatística aponta que dados não independentes devem ser analisados em conjunto [Vicini & Souza 2005]. Dizer que um dado é dependente significa que ele pode ser usado para inferir informação sobre um outro. A estatística também aponta que a falta de independência pode ocasionar interpretações equivocadas nas análises de inferência estatística.

O principal objetivo da EDM é antigo e estudado há muito tempo, que é conseguir compreender como ocorre o processo da aprendizagem. A diferença é que agora os pesquisadores possuem uma grande escala de dados e conseguem analisar uma aprendizagem prática e real (sem a antiga necessidade de realizar experimentos para obter dados). Sendo possível, por exemplo, observar estudantes em um curso durante oito meses e descobrir quais atividades geram melhor aprendizado a longo prazo, verificar o desempenho do aluno em sala de aula de acordo com o tempo que ele inicia a tarefa de casa ou se revisões do conteúdo visto em sala trazem algum benefício ou não [Romero et al. 2011].

Os dados educacionais analisados pela EDM tornam-se mais específicos proporcionalmente com o uso da tecnologia na realização das aulas. Quanto mais tecnologia,

[7] Geralmente as modificações dos métodos de Mineração de Dados para essa particularidade, incluem métodos da literatura psicométrica integrados com métodos de aprendizagem de máquina [Baker 2010].

mais dados podem ser armazenados e coletados automaticamente. Sendo possível, por exemplo, controlar os diferentes recursos de aprendizagem utilizados, como vídeos, áudios, arquivos de texto, questionários, testes de auto avaliações, etc. Convencionalmente, as instituições de ensino possuem os dados pessoais dos alunos (ex.: endereço, situação econômica, idade), as notas obtidas em provas, a frequência às disciplinas, as atividades realizadas, os planos de aula e provavelmente, essas instituições, também possuem dados que tornam provável a realização de uma análise de sua organização, dando detalhes sobre o sistema de ensino de forma geral. Mesmo com todos esses dados disponíveis, algumas pesquisas com metodologia qualitativa, por exemplo, necessitam da aplicação de questionários, geralmente, aos alunos.

Baker & Inventado (2014) consideram que a EDM pode ser vista tanto como uma área de investigação científica, quanto como uma comunidade de pesquisa, que para Romero et al. (2011), é realizada/composta por pessoas de várias disciplinas, pois nenhuma disciplina específica tem a *expertise* necessária para conduzir as pesquisas. A área da Ciências da Computação oferece métodos para explorar os muitos e diversos dados do ambiente educacional, por meio de aprendizagem de máquina e mineração de dados. A Estatística e a Psicometria fornecem o conhecimento para compreender e analisar projetos de estudo complexos que utilizam dados reais. Psicólogos e educadores são os responsáveis pelo avanço no domínio científico da Educação, sendo considerados "peças-chaves" no processo da EDM, por possuírem conhecimentos básicos sobre o processo de ensino e aprendizagem.

A EDM surgiu em 2005, através de uma série de *workshops* [Baker & Inventado 2014]. Em 2008, tornou-se uma conferência internacional anual (*International Conference on Educational Data Mining*), dando origem ao *Journal of Educational Data Mining*[8] em 2009. Atualmente, a EDM conta com alguns livros, como o *Data Mining in E-Learning* de 2006, *Handbook of Educational Data Mining* de 2011 e o *Educational Data Mining: Applications and Trends* 2013. Dentre os países dos pesquisadores que possuem mais destaque, de acordo com Romero & Ventura (2013) temos: Estados Unidos, Austrália, Canadá, Alemanha e Israel. Quanto ao Brasil, de acordo com Baker et al. (2011), os primeiros trabalhos publicados ocorreram por volta de 2006 e mesmo assim, até hoje, o número de publicações nos principais anais e periódico citado é mínimo.

[8] É gratuito e pode ser acessado em http://www.educationaldatamining.org/JEDM/.

Para o Brasil conseguir uma maior representatividade no cenário mundial da Mineração de Dados é necessário um investimento na disponibilização de dados. Difundindo o uso de softwares educacionais que produzem grande quantidade de dados educacionais já estruturados e a criação de repositórios abertos padronizados com os dados das instituições públicas presentes no país. Esse cenário já é real nos Estados Unidos e em alguns outros países [Baker et al. 2011], onde enfrentam agora, novos desafios com a Mineração de Dados Educacionais: *Big Data*, Computação em Nuvem, Redes Sociais, Mineração na *Web*, Mineração de Textos, Ambientes Virtuais 3D, Mineração Espacial, Mineração Semântica, Aprendizagem Colaborativa, *Learning Companions*, etc [Peña-Ayala 2014].

2.1.1 Principais Aplicações

A EDM pode ser aplicada para resolver muitas tarefas [Romero & Ventura 2010; Romero & Ventura 2013], dentre as quais destacam-se:

- **Criação de alertas**: como forma de comunicação aos interessados no processo de aprendizagem, auxiliando administradores e educadores na tomada de decisão, através da análise das atividades realizadas pelos alunos e informações de uso dos recursos/materiais do curso, por exemplo. Os alertas podem ser sobre comportamentos indesejáveis, como baixa motivação, uso indevido dos recursos, probabilidade de evasão, etc. As técnicas mais frequentes são mineração de processo e análise exploratória dos dados, através de análise estatística e visualizações/relatórios.
- **Manutenção e melhoria dos cursos**: envolve tarefas de pesquisa científica, construção de material didático, planejamento e programação, onde se deve analisar como ocorre a aprendizagem do aluno, verificando por exemplo, o que foi e como foi utilizado. Sendo importante a realização de testes de teorias sobre a aprendizagem baseada em novas tecnologias, para a formulação de novas hipóteses científicas, que poderão apoiar a construção ou adequação dos materiais didáticos, o planejamento de futuros cursos, quais disciplinas um aluno deve cursar em um período, alocação de recursos, etc. As técnicas mais utilizadas são associação, agrupamento e classificação.
- **Geração de recomendação**: verifica as necessidades do aluno em um dado momento e gera uma recomendação, que pode lhe proporcionar aprofundamento em um determinado domínio ou auxiliá-lo em uma dúvida. As recomendações

podem ser links para visita, dicas ou a resolução para um problema, a indicação de algum recurso ou material, um curso que pode ser feito, etc. A maioria das técnicas para esta tarefa envolvem associação, sequenciamento, classificação e agrupamento.

- **Previsão de notas e resultados de aprendizagem**: é o foco desta dissertação e essa tarefa é considerada como a mais antiga e popular da Mineração de Dados na educação. Consiste em utilizar os dados de atividades do curso para prever as notas finais do aluno ou algum outro tipo de resultado de aprendizagem, como uma possível evasão ou futura capacidade de aprender algo. As técnicas utilizadas podem ser regressão linear, classificação, agrupamento e associação.
- **Criação de perfis de alunos**: utilizada para detectar o estado e as características dos estudantes, como a satisfação, motivação, progresso de aprendizagem, estilo de aprendizagem, preferências e assim por diante. Para a criação desses perfis, também é considerado alguns problemas que possam impactar negativamente nos resultados de aprendizagem, como a apresentação de muitos erros na realização de uma tarefa, a má utilização ou subutilização de tutores inteligentes, manipulação dos sistemas, exploração dos recursos de forma ineficiente, etc. As técnicas mais frequentes além de agrupamento, classificação e análise de associação, são análises estatísticas, redes Bayesianas, modelos psicométricos e aprendizado por reforço.
- **Análise da estrutura de domínio**: realizada para determinar a qualidade do conteúdo apresentado e a sequência em que ele foi dado, através da previsão do desempenho dos alunos, descrevendo o domínio de instrução em termos de conceitos, habilidades, itens de aprendizagem e suas inter-relações. As técnicas mais utilizadas são regras de associação, agrupamento e algoritmos *space-searching*.

Em algumas aplicações, as tarefas apresentadas podem se relacionar entre si, como: geração de recomendação de acordo com a criação do perfil do aluno (I), previsão de notas e a criação de alertas, com as possíveis evasões (II), manutenção e melhoria dos cursos, através dos perfis dos alunos, da previsão de notas e da estrutura de domínio (III); e assim por diante. Essas possibilidades de relacionamento, também justificam a segregação da Mineração de Dados como Mineração de Dados Educacionais e caracterizam o quão complexa e desafiante, essa área de pesquisa pode ser.

2.1.2 O Processo de Descoberta do Conhecimento Educacional

O processo de descoberta de conhecimento educacional pode variar de acordo com diferentes pontos de vista [Romero et al. 2011], mas de forma didática, pode ser representado por um ciclo iterativo, representado na Figura 5.

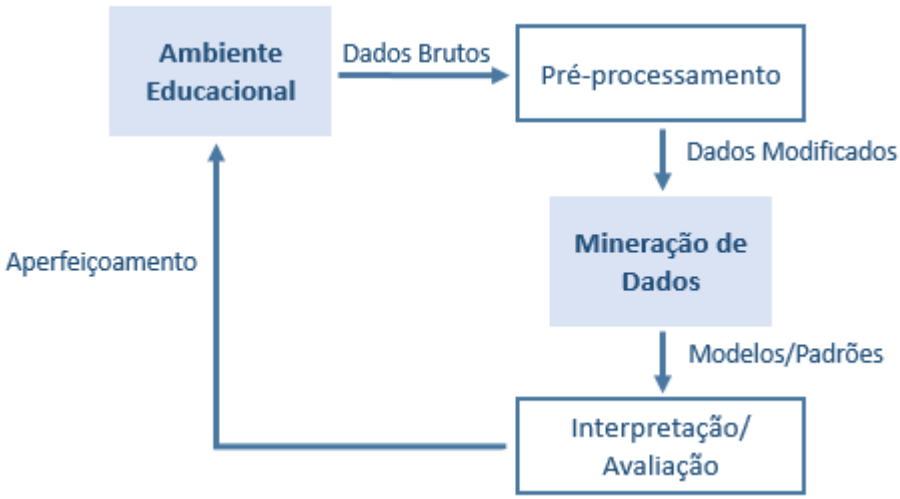


Figura 5. Processo de mineração de dados educacionais, adaptada de [Romero & Ventura 2013].

Nessa interpretação de Romero & Ventura (2013) é possível perceber que o processo de Mineração de Dados Educacionais não é apenas responsável por transformar dados em conhecimento, mas também em modificar o ambiente educacional para melhorar a aprendizagem do aluno de forma contínua. Também é notável a semelhança desse processo com o de Mineração de Dados. As principais diferenças entre eles podem ser vistas a seguir:

- **Ambiente Educacional**: pode ser formado por diferentes tipos de ambientes e sistemas de suporte, como por exemplo, sala de aula tradicional que faz uso de computadores ou web com auxílio de tutores inteligentes. Cada conjunto de ambiente educacional e sistema de suporte, gera diferentes tipos de dados, que precisam ser pré-processados de acordo com as particularidades de cada um e a questão a ser resolvida pela Mineração de Dados.
    - **Dados Brutos**: podem ser coletados a partir de dados administrativos, observações de campo, questionários motivadores, medições realizadas por experimentos controlados, provas, etc. Os ambientes educacionais podem armazenar uma enorme quantidade de dados oriundos de várias fontes, com diferentes formatos e diferentes níveis de granularidade ou múltiplos níveis

de hierarquia, que fornecem mais ou menos dados, tal como demonstrado na Figura 6.

Figura 6. Quantidade e granularidade dos dados educacionais, adaptada de [Romero & Ventura 2013].

- **Pré-processamento**: responsável por escolher, reunir e integrar os diferentes tipos de dados coletados que podem ser úteis para a solução desejada. Ou seja, converte os dados a serem utilizados, em uma forma adequada (dados modificados) para resolver um problema educacional específico.
- **Mineração de Dados**: grande parte das técnicas tradicionais de Mineração de Dados foram aplicadas ao domínio educacional com sucesso [Baker 2010]. No entanto, dependendo dos dados educacionais escolhidos, é necessário um tratamento diferenciado, graças as peculiaridades deles entre si. Como por exemplo, a utilização de dois tipos de dados: os que demonstram como se deu o aprendizado e os que mapeiam o perfil do aluno. A EDM, surge para ajudar os pesquisadores a definir quais técnicas de Mineração de Dados podem ser adotadas, adaptadas ou desenvolvidas.
- **Interpretação dos Resultados**: é utilizada para aplicar os conhecimentos adquiridos na melhoria do ambiente educacional (ou do sistema de suporte). Por isso, os modelos obtidos pelos algoritmos de Mineração de Dados devem ser apresentados de forma compreensível para a tomada de decisão, sendo por meio de uma lista de sugestões/conclusões sobre os resultados e como aplicá-los ou pela redução das regras de associações descobertas, transformadas em gráfico.

2.1.3 Métodos de Mineração de Dados

Muitas técnicas de Mineração de Dados são aplicadas com sucesso para o domínio da Educação, como previsão, classificação, agrupamento e mineração de relações [Baker & Yacef 2009; Romero et al. 2011]. No entanto, as particularidades dos sistemas educacionais fazem com que os pesquisadores envolvidos com a EDM busquem novas técnicas, extraídas das diversas áreas pertencentes à comunidade de pesquisa da EDM (computação, estatística, psicometria, psicologia e educação) [Peña-Ayala 2013].

Baker & Yacef (2009) propõem uma das primeiras categorizações para os métodos de EDM, onde os dois últimos são considerados típicos da EDM:

- Previsão[9]
  - Classificação
  - Regressão
  - Estimação de Densidade
- Agrupamento
- Mineração de relações
  - Mineração de Regras de Associações
  - Mineração de Correlações
  - Mineração de Padrões Sequenciais
  - Mineração de Causas
- Destilação de dados para facilitar decisões humanas
- Descobertas com modelos.

Bienkowski et al. (2012) e Romero & Ventura (2013) complementam essas categorias com:

- Detecção de *outlier*
- Análise de redes sociais
- Mineração de processo
- Mineração de texto
- Rastreamento do conhecimento
- Fatoração de matriz não negativa.

[9] Será discutido com mais afinco no capítulo 3, já que é o método utilizado nesta pesquisa.

2.2 *Learning Analytics*

O *Learning Analytics* (LA) surgiu após a criação da EDM [Baker & Inventado 2014]. E de acordo com Romero e Ventura (2013), o LA é responsável pela aplicação de modelos de previsão já conhecidos em Sistemas Educacionais. Ou seja, de maneira simplificada, a EDM busca novos padrões em dados educacionais para a construção de modelos e o *Learning Analytics* os aplica. A Figura 7 consegue ilustrar as definições e diferenças entre Sistemas Educacionais, Mineração de Dados e Aprendizagem de Máquina, EDM e *Learning Analytics*.

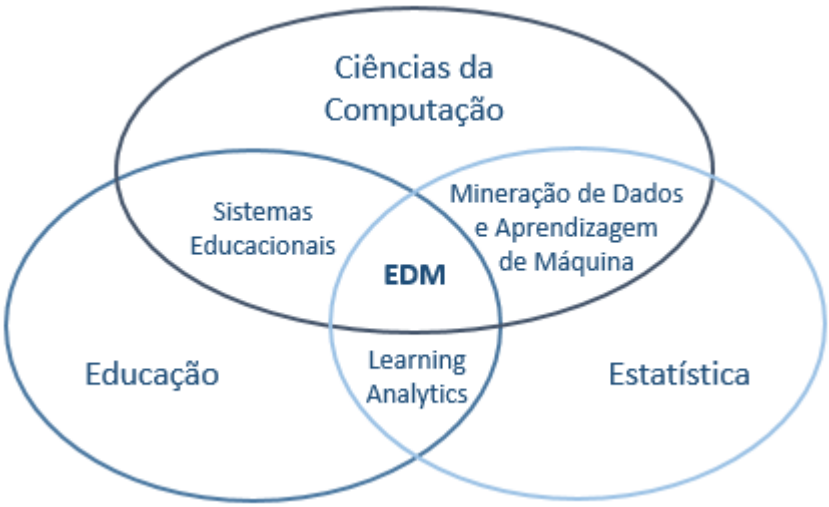


Figura 7. Áreas relacionadas com a Mineração de Dados Educacionais, adaptada de [Romero & Ventura 2013].

Há outras diferenças entre o LA e a EDM, além da ausência ou não da Ciências da Computação [Romero & Ventura 2013]:

- **Técnicas**: em LA, as técnicas mais utilizadas estão relacionadas com estatística, visualização, análise de redes sociais, de sentimentos, de influência, de discurso, de conceitos e modelos de construção de sentido. Na EDM, as técnicas mais utilizadas são a classificação, agrupamento, modelagem Bayesiana, mineração de relações e descoberta com modelos.
- **Origens**: o LA tem forte origem na Web Semântica, currículo inteligente e intervenções sistêmicas. Já a EDM, em Software Educacional, modelagem de estudantes e previsão de resultados no curso.
- **Ênfase**: LA, descrição de dados e resultados. EDM, na descrição e comparação das técnicas de DM utilizadas.
- **Tipo de descoberta**: no LA, a descoberta automatizada é a ferramenta para impulsionar o julgamento humano. E na EDM, o julgamento humano é a ferramenta para a descoberta automatizada.

Essas diferenças são apoiadas também por Baker & Inventado (2014), mas eles consideram o LA como uma comunidade independente da EDM, que não apenas aplica seus modelos. Para eles, o LA também pode criar novos modelos, de acordo com suas preferências em técnicas, ênfase e tipo de descoberta.

2.3 Previsão

O objetivo da previsão é construir um modelo que possa prever alguma situação desconhecida a partir de situações que já ocorreram (dados históricos). Para a realização dessa tarefa, é importante conhecer os tipos de dados existentes e que tipo de informação deseja-se obter/prever. Por exemplo, para prever se um aluno merece ou não uma bolsa de estudos, é necessário verificar quais alunos já receberam ou não essa bolsa e quais evidências, na época, contribuíram para essa decisão (notas, quantidade de artigos publicados, locais de publicação, etc). O recebimento ou não da bolsa é o que se deseja prever, ou seja, a saída esperada, que é a variável preditora. As evidências são os dados de entrada, elas ajudarão na previsão e são as variáveis preditivas.

Tendo identificado a variável preditora e as variáveis preditivas é possível determinar qual método é aconselhado para realizar a previsão. De acordo com a EDM e a categorização de Baker & Yacef (2009) apresentada na Seção 2.1.3, os métodos de previsão estão agrupados em classificação, regressão e estimação de densidade:

- Na **classificação**, a variável preditora é discreta ou nominal, como no exemplo da bolsa de estudos. Alguns métodos populares incluem árvores de decisão, regressão logística e máquina de vetores de suporte (SVM, do inglês: *support vector machine*).
- Na **regressão**, a variável preditora é uma variável contínua, como por exemplo, a descoberta das notas de uma disciplina, considerando todos os valores entre 0 e 10. Alguns métodos populares incluem regressão linear, redes neurais e regressão com SVM.
- Na **estimação de densidade**, a variável preditora é uma função de densidade de probabilidade que trata de variáveis aleatórias contínuas, ou seja, são descobertos a distribuição normal e o desvio padrão de uma avaliação e, a partir deles, é possível determinar a probabilidade de um aluno tirar uma determinada nota. Os estimadores podem ser baseados em uma variedade de funções *kernel*, incluindo função gaussiana. A estimação de densidade é pouco

utilizada na EDM devido à ausência de independência estatística dos dados educacionais [Baker et al. 2011].

Os métodos utilizados para previsão são executados através de uma aprendizagem supervisionada. Os dados de entrada são conhecidos (ex.: o número total de atividades realizadas e as notas obtidas) com a finalidade de realizar ajustes do que se deseja prever, já que diferentes métodos de previsão são mais efetivos, dependendo do tipo de variáveis de entrada utilizadas. E cada método a ser utilizado deve ser escolhido de acordo com o tipo de variável a ser prevista [Tan et al. 2009; Baker 2010; Han et al. 2011].

A aplicação de um método preditivo, de um modo geral, é feita em duas etapas principais, onde os dados são divididos em dados de treinamento e teste [Castro & Ferrari 2016]:

- **Treinamento**: etapa de geração do preditor. Ocorre um aprendizado a partir de uma base de treinamento, que possui pares de entrada e saída. Essa base, deve descrever/distinguir um conjunto predeterminado de classes ou valores. Por exemplo, a quantidade de atividades realizadas por um aluno em uma disciplina seria a entrada e a nota dele nessa disciplina, a saída.
- **Teste**: etapa de avaliação do preditor, uma vez que utiliza dados que não estão contidos na base de treinamento. O desempenho do preditor com os dados de teste oferece uma estimativa de sua capacidade de generalização, ou seja, se ele consegue responder corretamente as previsões a partir de dados que não foram utilizados no processo de treinamento.

Um modelo preditivo, assim como os métodos utilizados por ele, podem ser comparados e avaliados seguindo alguns critérios [Han et al. 2011]:

- **Acurácia**: é a capacidade de generalização do método e é realizada através de um cálculo de precisão. Esse cálculo é apenas uma estimativa de quão bem o preditor realizou uma dada previsão.
- **Velocidade**: refere-se aos custos computacionais envolvidos na geração e utilização do preditor.
- **Robustez**: é a capacidade do preditor em fazer previsões corretas, mesmo utilizando dados com ruído ou com valores em falta.
- **Escalabilidade**: o preditor deve conseguir ser eficiente mesmo com grande quantidade de dados.

- **Interpretabilidade**: é o nível de compreensão e percepção fornecido pelo preditor. A Interpretabilidade é subjetiva e por isso, de difícil avaliação.

A avaliação de um modelo tem como objetivo apresentar um valor quantitativo da sua qualidade e, na maior parte das vezes, tem o foco na generalização [Castro & Ferrari 2016]. Isso leva à conclusão de que a acurácia é a avaliação mais importante a ser realizada, apesar dela ser feita apenas com um percentual de previsões corretas e existir outras métricas (também provenientes da matriz confusão [Han et al. 2011]) que podem dar uma melhor explicação sobre as previsões obtidas. Métodos populares para realizar essa avaliação incluem correlação linear, Kappa de Cohen e curva de Precisão x Revocação (curva ROC) [Baker 2010].

A partir dos conceitos apresentados, pode-se ilustrar de uma forma geral que o processo de previsão ocorre de acordo com a Figura 8.

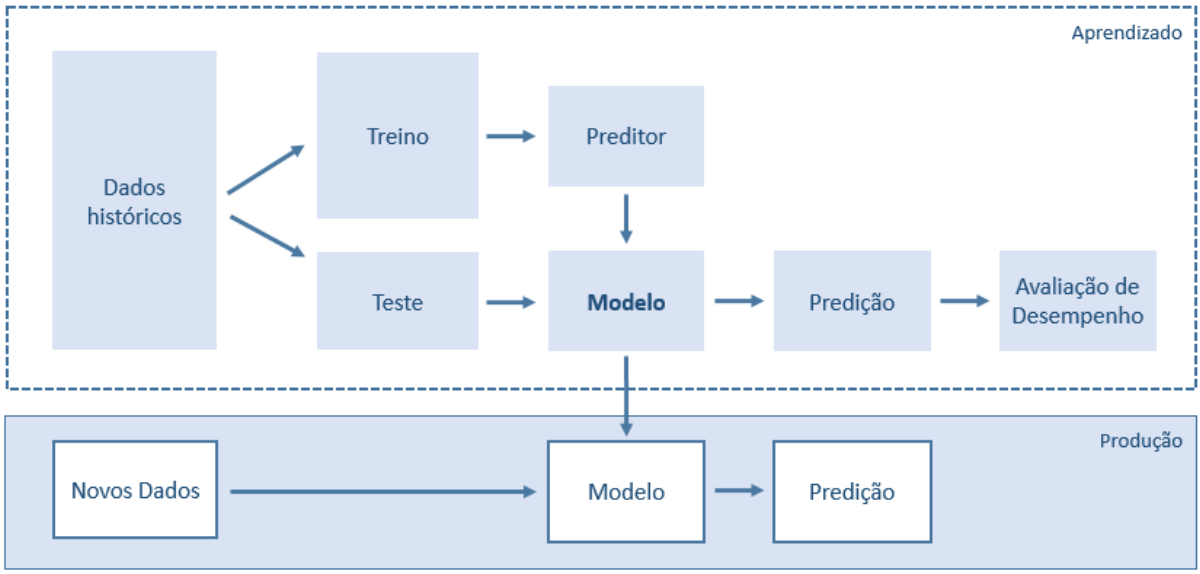


Figura 8. Processo de previsão, adaptada de [Amaral 2016].

2.3.1 Dados de Treinamento e Teste

Como visto anteriormente durante a aplicação de um método preditivo, é necessário dividir os dados, para garantir que o modelo desenvolvido não seja superestimado. É como treinar para uma prova quando já se conhece as questões exatas que a compõem. Às vezes, para assegurar que um método possui uma boa acurácia, sua base de dados é dividida em vários conjuntos de treinamento e teste, que são validados entre si. E o conjunto de treinamento que for mais satisfatório é escolhido para a composição do modelo.

O particionamento dos dados antes da execução de um método é importante não apenas como uma garantia de sua acurácia, mas também para evitar anomalias nos resultados como o *overfitting* (sobreajuste). Esse problema ocorre quando o algoritmo executado gera um modelo muito ajustado aos dados de treinamento, memorizando as relações e estruturas realizadas com esses dados, assim como ruídos ou coincidências. Um modelo preditivo deve ser construído para representar dados de treinamento e não para reproduzi-los [Ratner 2011]. No domínio da educação, o *overfitting* ocorre com frequência quando há uma grande quantidade de atributos disponíveis para a construção de um modelo complexo e poucos dados para o modelo aprender sobre esses atributos com uma boa precisão [Hämäläinen & Vinni 2011].

Balanceamento

Na classificação, as variáveis previstas são discretas ou categóricas. Ou seja, podem ser divididas em classes e ocorrer que essas classes possuam tamanhos diferentes. Por exemplo, em um problema que se deseja prever os alunos aprovados e reprovados, temos que a base de dados tem uma quantidade de alunos aprovados bem maior que a quantidade de alunos reprovados. Isso significa que a acurácia da previsão pode ser influenciada por esse desbalanceamento das classes.

Os algoritmos tradicionais da classificação foram desenvolvidos para maximizar a sua taxa de precisão, independentemente da distribuição de classes. Ou seja, durante a etapa de treinamento pode ocorrer um maior aprendizado sobre os alunos aprovados, já que eles representam a maior proporção dos dados. E assim, a minoria dos dados, representada pelos alunos reprovados, pode ser prevista com maior taxa de erro durante a etapa de teste.

Geralmente as classes majoritárias são favorecidas, enquanto as classes minoritárias possuem baixa taxa de reconhecimento [Castro & Braga 2011]. Em grande parte das vezes, são estas as classes de maior interesse para a tarefa de previsão, fazendo com que o custo envolvendo os erros de classificação da classe minoritária seja normalmente maior do que os da classe majoritária [Barella 2016].

Uma maneira de resolver esse problema é atuar durante o pré-processamento dos dados, transformando a amostragem em igualitária ou balanceando a distribuição das classes [Márquez et al. 2013]. Existem vários algoritmos de balanceamento e um dos mais utilizados é o SMOTE (*Synthetic Minority Oversampling Technique*) [Chawla et al. 2002]. Esse algoritmo

ajusta a classe minoritária através da criação de dados “sintéticos”, que são gerados com base na similaridade entre “n” vizinhos.

Validação Cruzada

Taxas de precisão estão sujeitas a margens de erro, para mais ou para menos. Quando os dados de treinamento são escassos, a validação cruzada pode ser escolhida para treinar e avaliar o algoritmo escolhido. Os dados de treinamento têm suas instâncias particionadas aleatoriamente em *k* partições mutuamente exclusivas. A quantidade *k* de partições geradas determina a quantidade de treinamentos e testes a serem realizados. Por exemplo: com um total de quatro partições, teremos quatro interações de treinamento e teste, em cada interação teremos uma partição utilizada para teste e as demais para treinamento. Assim, tem-se que cada partição é usada (k-1) vezes para treinamento e uma vez para teste (Figura 9), ao final das interações, o desempenho é medido a partir da média aritmética das avaliações.

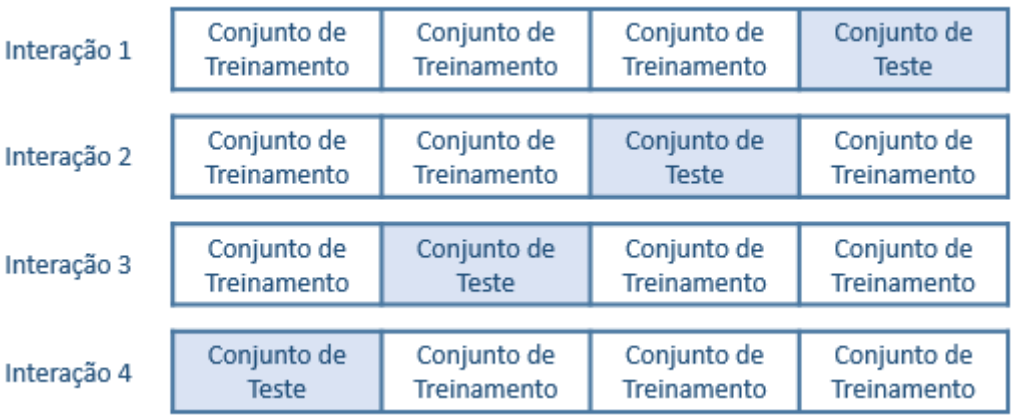


Figura 9. Validação cruzada, adaptada de [Quilici-Gonzalez & Zampirolli 2015].

A validação cruzada consegue estimar de forma precisa por integrar todos os dados nas etapas de treinamento e teste. Uma variação especial da validação cruzada é o *leave-one-out*. Nessa variação cada partição é composta por uma única instância dos dados. Se existem dez instâncias, existirão dez partições. Cada instância é utilizada como teste das demais que participarão do treinamento. Dessa forma, o treinamento pode ser utilizado com uma quantidade maior de dados [Quilici-Gonzalez & Zampirolli 2015].

2.3.2 Classificação

Classificar é reunir instâncias em classes a partir de atributos/características semelhantes. Para os modelos preditivos, essa classe é a variável a ser prevista. Ou seja, um

algoritmo de classificação verifica as características de uma instância e retorna a classe à qual ela pertence.

Essa classe é formada por uma variável discreta ou nominal, que pode ser binária ou categórica. Entende-se que uma variável binária, não admite valores fracionários e que ela varia através de unidades inteiras, por exemplo, a contagem de pessoas, o número de respostas positivas/negativas ou o número de pontos ao se lançar dois dados. E uma variável categórica também pode ser denominada como qualitativa. Ela indica uma qualidade presente ou ausente e possui categorias mutuamente exclusivas, por serem geralmente, bem delimitadas e completas. Por exemplo, o conceito em uma disciplina (aprovado/reprovado) ou o estágio de uma doença (inicial, intermediário ou terminal). A ferramenta Weka também apresenta alguns exemplos:

Quadro 1. Dados de exemplos do Weka [Amaral 2016].

Arquivo ARFF	**Descrição**	**Atributos**	**Classe**
contact-lenses.arff	Tipos de lentes de contatos que melhor se ajustam em pacientes.	Idade, astigmatismo, produção de lágrima.	Se o paciente deve receber lentes de contato macias, duras ou se não deve receber lentes.
iris.arff	Dados de medidas de sépala e pétala de flores íris.	Largura e comprimento da pétala e da sépala.	A espécie da classe: setosa, versicolor, virginica.
soybean.arff	Dados de plantações de soja.	Diversas características da plantação, como temperatura, germinação, folhas, etc.	Diversos tipos de doenças de soja.

Cada linha do Quadro 1 representa um modelo de previsão onde ocorre uma aprendizagem de máquina para prever/classificar novos dados que serão apresentados ao algoritmo de classificação. A diferença entre um algoritmo comum e um que utiliza aprendizagem de máquina, está na necessidade de utilização de dados históricos para conhecer uma determinada situação.

Um algoritmo que calcula o melhor aproveitamento de papel para uma impressão, precisa apenas das medidas das impressões a serem feitas, para retornar a melhor disposição destas impressões no papel e ele pode realizar essa tarefa sem nunca ter calculado uma

impressão antes. Um algoritmo que prevê se uma pessoa será boa ou má pagadora de um empréstimo, precisa de dados históricos para a máquina conhecer/aprender sobre o perfil de um bom ou mau pagador, através de seu histórico de empréstimos anteriores, sua renda, número de filhos, se tem casa própria ou não, etc [Amaral 2016].

O desempenho do algoritmo de impressão, provavelmente será sempre o mesmo, desde que o processo de impressão não seja alterado e espera-se que o aproveitamento do papel seja sempre de 100%. Já o algoritmo de empréstimo poderá sofrer mudanças em seu desempenho, uma vez que o perfil da pessoa que busca o empréstimo pode alterar de acordo com as situações sociais/econômicas ou ainda com o tipo de banco utilizado, sendo necessário por exemplo, que um banco público e privado, tenham seus próprios modelos de previsão.

Para o contexto educacional, as questões de desempenho não diferem, mas Baker (2010) enfatiza que é importante considerar a não independência de diferentes observações envolvendo o mesmo aluno. Nesse artigo, o autor diz que pesquisadores de mineração de dados educacionais frequentemente aplicam métodos meta-analíticos que podem explicar a independência parcial ou selecionam estimadores excessivamente conservadores que assumem a não dependência completa.

2.4 Ferramentas de apoio

Na Figura 5 da Seção 2.1.2, foi visto que o processo de descoberta de conhecimento possui as seguintes etapas: pré-processamento, mineração de dados e interpretação/avaliação. Cada uma dessas etapas possui técnicas que podem ser específicas à determinadas aplicações. Além disso, cada técnica pode apresentar um ou mais algoritmos que agregam diferentes estratégias para melhorar a performance de cada etapa.

Implementar todos esses algoritmos sempre que for necessário extrair conhecimento de uma base é extremamente dispendioso. Por isso, softwares, frameworks e bibliotecas foram construídos, para auxiliar esse processo:

- **Sistemas de Gerenciamento de Bancos de Dados (SGBDs)**: como o *Oracle*, *Sql server*, *MySQL* ou o *PostgreSQL*, são conhecidos por administrar e manter bases de dados além de gerenciar os acessos e as manipulações aos dados. Atualmente, alguns deles também são capazes de realizar tarefas de pré-processamento de dados, análise descritivas e visualização (gráficos e relatórios), permitindo até mesmo, programar algoritmos para tarefas mais complexas de mineração, como agrupamento, classificação, detecção de

anomalias, entre outras.

- **Weka**: é um software gratuito de código aberto, desenvolvido em Java e mantido pela Universidade de Waikato (www.cs.waikato.ac.nz/ml/weka). Com ele, é possível realizar tarefas de pré-processamento, classificação, regressão, agrupamento e visualização dos dados. Também é possível planejar e executar análises/experimentos mais complexos através de fluxogramas que encadeiam as tarefas de mineração de dados. Por se tratar de um software livre, as suas bibliotecas podem ser integradas a ambientes de desenvolvimento Java (*Eclipse* ou *NetBeans*) para a realização de alterações ao processo.
- **RapidMiner**: é um software que possui versões gratuitas e pagas (rapidminer.com). Em sua plataforma, há três produtos principais: *RapidMiner Studio, RapidMiner Server e RapidMiner Radoop*. O primeiro é um ambiente visual de programação, que constrói projetos de análise de dados por meio de blocos e fluxogramas, possui conexão direta à base de dados, ferramentas específicas para a realização de um pré-processamento dos dados e pode realizar a mineração através da classificação, regressão, agrupamento e associação. O segundo produto, é utilizado para a replicação e compartilhamento dos modelos construídos pelo primeiro, possui recursos para agendamento, controle de versão, acesso remoto, etc. O terceiro visa análises de *Big Data* e possui *plug-ins* com funções de mineração de dados da *web*, mineração de textos e integração com o Weka, assim como, com as linguagens *Python* e R, sendo possível o desenvolvimento de algoritmos customizados.
- **Python**: é uma linguagem de programação, orientada a objetos, que por meio de suas bibliotecas pode realizar coleta de dados, engenharia de dados, análise, *web scraping* (extração de dados/conteúdo em *sites*), construção de aplicativos na *web*, etc. Alguns de seus pacotes úteis para a Mineração de Dados são o *SciPy/NumPy* (computação científica), Pandas (manutenção/análise de dados), *Matplotlib* (gráficos) e *Sckit-learn* (aprendizado de máquina). Essa linguagem é utilizada para a análise de dados principalmente, quando se deseja ter o acompanhamento das análises por meio de aplicativos na *web* ou quando códigos estatísticos precisam estar integrados com servidores em ambiente de produção.
- **R**: também é uma linguagem de programação e um ambiente de

desenvolvimento integrado para a realização de cálculos estatísticos e gráficos. Ela foi desenvolvida por estatísticos para estatísticos, engenheiros e cientistas sem conhecimento de programação de computadores. Possui um grande número de pacotes para análise de dados, com modelos, fórmulas e testes estatísticos. Também é possível executar pacotes do Python através do pacote rPython, outros pacotes importantes são *dplyr*, *plyr*, *data.table* (manipulação de dados), *stringr* (manipulação de *strings*), *zoo* (*time-series*), *ggvis*, *lattic* e *ggplot2* (gráficos) e *caret* (aprendizado de máquina). Com o R é fácil escrever fórmulas complexas e praticamente todos os tipos de testes e modelos estatísticos estão disponíveis para o uso.

- **Anaconda**: é uma plataforma de código aberto que une a linguagem Python e R, com várias bibliotecas para a análise de dados.

A existência dos softwares para apoio a análise de dados faz com que muitas pesquisas sejam realizadas de forma empírica, já que não há necessidade de implementação dos algoritmos e há diversos deles a serem testados de forma rápida e eficiente, sendo necessário apenas conhecer em que categoria da Mineração de Dados o problema se enquadra (ex.: previsão, agrupamento e mineração de relações). Dependendo do objetivo da pesquisa, também é importante conhecer os algoritmos das abordagens escolhidas, para conseguir realizar os ajustes de seus parâmetros de acordo com cada software e assim, melhorar a performance da mineração.

Quando é decido realizar análises mais específicas e customizadas, é preferível o uso das linguagens de programação disponíveis para a análise dos dados. No caso desta dissertação, a existência dos softwares torna possível o foco na seleção de evidências para um posterior avanço no estudo da abordagem mais apurada (trabalho futuro).

3 Trabalhos Relacionados

Como forma de incentivar e guiar futuros pesquisadores da área de EDM, esta seção apresentará os principais livros e artigos da área. Além, é claro, de analisar os trabalhos mais relacionados ao tema principal.

Para o descobrimento do material apresentado foram utilizados o Google, o Google Acadêmico, a Revista Brasileira de Informática na Educação (RBIE) e uma verificação em alguns anais de eventos na área de Informática e Educação:

- CBIE – Congresso Brasileiro de Informática na Educação
  - SBIE – Simpósio Brasileiro de Informática na Educação
  - WIE – *Workshop* de Informática na Escola
- TISE – Conferência Internacional sobre Informática na Educação

O site da sociedade internacional de Mineração de Dados Educacionais [http://www.educationaldatamining.org] também foi utilizado. Ele dispõe o *Journal of Educational Data Mining* (JEDM) e anais de congressos, além de indicar alguns recursos como: dados educacionais abertos, dicas de ferramentas, notas de discussões dos *workshops*, inscrição para listas de discussão (*EDM-ANNOUNCE* e *EDM-DISCUSS*) e a chance de qualquer um sugerir/disponibilizar dados abertos ou ferramentas para serem adicionadas ao site.

3.1 Mineração de Dados Educacionais

Um excelente artigo para iniciar os estudos na EDM foi escrito por Cristóbal Romero e Sebastian Ventura, intitulado "*Data mining in Education*", escrito em 2013 [Romero & Ventura 2013]. Com ele, é possível conhecer os *workshops*, congressos e *journals* da área, além de aprender de forma didática sobre o contexto da EDM e como o processo de descoberta de conhecimento educacional é feito. Os métodos (categorias), aplicações e ferramentas também são relatados nesse artigo.

Alguns livros sobre EDM foram escritos colaborativamente por pesquisadores internacionais que obtiveram destaque com os seus trabalhos:

- "*Data mining in e-learning*", 2006 – Cristóbal Romero e Sebastian Ventura;
- "*Handbook of Educational Data Mining*", 2011 – Cristóbal Romero, Sebastian Ventura, Mykola Pechenizkiy e Ryan Baker;

- "*Educational Data Mining Applications and Trends*", 2014 – Alejandro Peña-Ayala;
- "*Data Mining and Learning Analytics*", 2016 – Samira Elatia, Donald Ipperciel e Osmar R. Zaïane
- "*Learning Analytcs: Fundaments, Applications and Trends*", 2017 – Alejandro Peña-Ayala;

Todos os livros apresentados têm o mesmo objetivo de relatar o estado da arte e fomentar trabalhos futuros. Também apresentam uma estrutura similar, onde os capítulos iniciais introduzem a EDM, *Learning Analytics* ou *e-learning*, com as técnicas básicas e tutoriais. Em seguida, expõem aplicações práticas e estudos de caso.

Os dois primeiros livros são considerados como marcos na história da EDM [Baker & Inventado 2014] e são modelos para os demais, dando a ideia de que muitos outros livros surgirão com o passar dos anos, seguindo o mesmo formato. Abaixo, uma breve apresentação deles:

- "*Data mining in e-learning*": é primeiro livro sobre EDM e mostra como os professores podem melhorar os sistemas de *e-learning*, descobrindo novos conhecimentos baseados nos dados de utilização dos alunos. Os autores concluem, de uma maneira geral, que a extração do conhecimento é uma das áreas mais promissoras em EDM. [Romero & Ventura 2006].
- "*Handbook of Educational Data Mining*": foi o primeiro manual sobre EDM. Ele reflete a natureza multidisciplinar da área, conseguindo aproximar as comunidades de educação e de mineração de dados. Os especialistas em educação entendem que tipos de questões a EDM pode abordar e os mineradores de dados entendem que tipos de questões são importantes para o projeto educacional e a tomada de decisões educacionais. O *handbook* também aponta que o principal objetivo da EDM é usar os dados educacionais em larga escala [Romero et al. 2011].

Todas essas publicações estão em inglês e apresentam um contexto internacional, onde pesquisadores brasileiros não estão inclusos. Por sinal, são pouquíssimos os que possuem trabalhos relatados na sociedade internacional de Mineração de Dados Educacionais. Geralmente, as bases de dados utilizadas nesses trabalhos também não são brasileiras. Mas um importante artigo para o Brasil é o "Mineração de Dados Educacionais - Oportunidades

para o Brasil", escrito em 2011, por Ryan Baker, Seiji Isotani e Adriana Carvalho. Nele, é apresentado o cenário da área e as principais dificuldades para o desenvolvimento dela, comparado aos Estados Unidos e a Europa [Baker et al. 2011]. Outros dois trabalhos interessantes que revelam o estado da arte no Brasil, são:

- [Magalhães et al. 2013] Caracterizando a Pesquisa em Informática na Educação no Brasil - Um Mapeamento Sistemático das Publicações do SBIE;
- [Gutiérrez Posada et al. 2016] A informática na Educação: o que revelam os trabalhos publicados no Brasil.

Apesar do primeiro trabalho considerar apenas as publicações do SBIE, ele verifica 12 edições (2001 – 2012, 835 trabalhos) através de aspectos históricos, conceituais e metodológicos. Esse artigo possui um bom modelo de análise a ser seguido. Os autores apresentam vinte e quatro tópicos de interesse da Informática na Educação, onde a Mineração de Dados pertence ao tópico "Mineração de Dados, Padrões e Repositórios Digitais de Materiais Educacionais" e ocupa a décima quinta posição das áreas mais exploradas no simpósio, com 44 trabalhos. Uma das conclusões formuladas pelos autores do artigo, diz que o desenvolvimento de arquiteturas/softwares educativos, em muitos casos, não possuem validação dos resultados obtidos e que mais da metade das publicações demonstram que a pesquisa em Informática na Educação está se desenvolvendo em publicações de caráter avaliativo.

O segundo trabalho, utiliza *tagclouds* (nuvens de palavras e expressões) geradas a partir dos títulos e resumos de artigos publicados no SBIE, WIE, CBIE e RBIE. Através das *tagclouds* os autores ilustram e discutem as principais diferenças entre essas fontes; descobrem os principais autores da área, as redes de cooperação da comunidade e o interesse de cada fonte em determinados temas (inclusão digital, tutor inteligente, educação a distância, objetos de aprendizagem, por exemplo). Ou seja, é proposto uma análise do domínio de publicações em temas da informática na educação. Deixando a avaliação de métodos ou tecnologias usadas nas pesquisas, como trabalho futuro.

3.2 Previsão

Foram selecionados trabalhos que constroem modelos de previsão que analisam o desempenho dos alunos ou alunos passíveis de evasão. Tendo em vista que para verificar as chances de um aluno abandonar um curso, é necessário verificar o seu desempenho.

Em um primeiro momento, foram selecionados trabalhos correlatos realizados no Brasil. Eles foram fonte de inspiração para a busca de evidências relacionadas ao problema desta dissertação. Por isso, após a descrição dos trabalhos são apresentadas as evidências utilizadas por cada um deles.

Posteriormente, com a seleção dos trabalhos no âmbito internacional, foi possível perceber o estado da arte cronologicamente. Os estudos mais antigos são os mais relacionados com os que estão sendo feitos no Brasil, atualmente. Os trabalhos nacionais e internacionais, foram separados e relatados seguindo a ordem de publicação.

Trabalhos nacionais:

Souto e Duduchi (2009) desenvolveram uma ferramenta computacional para a aplicação de uma avaliação de raciocínio lógico, que é utilizada como evidência na previsão das notas das provas de desempenho acadêmico na disciplina de Programação I. A avaliação de raciocínio lógico não utiliza nenhuma linguagem de programação específica e as provas de desempenho acadêmico são formuladas de forma semelhante à avaliação citada, apesar de possuir questões discursivas sobre conceitos básicos e sobre a criação de algoritmos. A correlação realizada entre o questionário e a prova é satisfatória (0,62) e a regressão linear mostra que a avaliação de raciocínio lógico proposta pode ser utilizada para previsão das notas das provas. **Evidências**: notas das avaliações de programação e notas de avaliações de lógica.

Martins et al. (2012) aplicam duas abordagens para a detecção de alunos passíveis a evasão. Na primeira, é montada com a ajuda de um especialista uma Rede Bayesiana com o perfil dos alunos desistentes e apenas ela é utilizada para a previsão. Na segunda, a Rede Bayesiana recebe mais evidências e é utilizada Mineração de Dados. Os autores apresentam que a segunda abordagem obteve melhor acuraria e concluem que o uso da Mineração de Dados é imprescindível para a tarefa de previsão e que o acréscimo de mais evidências pode melhorar o modelo construído. **Evidências**: acessos ao tutor (algumas seções possuem o tempo de permanência), notas (exercícios, provas, médias), quantidade de questões respondidas (discursivas e objetivas), frequência e situação do aluno ao final da disciplina (reprovado, desistente ou aprovado).

Santos et al. (2012) analisam uma disciplina com metodologia híbrida para verificar os alunos que possuem risco de reprovação. Os dados utilizados são semelhantes ao desta dissertação, com exercícios práticos e provas, mas apenas o primeiro módulo é analisado por

ser considerado o de maior dificuldade. Os autores concluem, através das análises realizadas, que os alunos de melhor desempenho são os que mais se dedicam nos exercícios práticos. **Evidências**: indicador de repetência, quantidade/percentual de presença nas aulas presenciais, quantidade/percentual de exercícios respondidos, quantidade/percentual total de presenças, nota de todos os exercícios (total de 8), nota média dos exercícios, nota da avaliação e rótulo da nota da avaliação (aprovado/reprovado).

Brito et al. (2014) realizam a previsão das notas dos alunos do primeiro período do curso de Ciências da Computação com base nas notas de ingresso através do vestibular. Os alunos são agrupados em duas classes: os aprovados em todas as disciplinas e os que reprovaram em ao menos uma disciplina. O objetivo do trabalho é diminuir a evasão. **Evidências**: nota de ingresso e notas das disciplinas do primeiro semestre.

Guércio et al. (2014) analisam duas disciplinas, ministradas a distância, a partir dos acessos ao AVA e das interações do fórum. Os autores descobriram que o acesso às atividades e a quantidade de postagens realizadas pelo aluno, respectivamente, são os atributos de maior relevância para análise proposta. A classificação ocorre a partir da discretização das notas em três classes de alunos: com maior tendência a reprovação, com tendência a reprovação com potencial para serem aprovados e com tendência a aprovação. **Evidências**: dados das disciplinas e acessos às disciplinas, fóruns, recursos e atividades.

Brito et al. (2015) têm o mesmo objetivo da pesquisa realizada em 2014 [Brito et al. 2014], e também utiliza as notas do primeiro semestre com as notas de ingresso. Diferente do trabalho anterior, os alunos são agrupados em evadidos e concluintes, e a taxa de acurácia dos algoritmos tem um acréscimo de mais de 10%. Apesar do trabalho anterior ser semelhante a este e ter autores em comum, não ocorre uma comparação entre as abordagens realizadas, uma vez que a segregação dos alunos demonstra ganho na acurácia. **Evidências**: nota de ingresso e notas das disciplinas do 1º semestre.

Manhães (2015) têm-se uma tese de doutorado que propõe uma arquitetura para a descoberta de conhecimento em dados, como prever o desempenho acadêmico do aluno em cada semestre e identificar o que ele faz para ter sucesso ou não. São realizados 6 estudos de caso, onde um deles avalia 12 algoritmos de previsão, um outro analisa os fatores que influenciam o desempenho dos alunos e os demais investigam os modelos de dados mais adequados para serem utilizados pelo maior número de cursos de graduação. Cada modelo de previsão gerado utiliza evidências diferentes, por isso seguem apenas, algumas delas.

Evidências: situação da matrícula, disciplinas que está cursando, notas, aprovações/reprovações, qual curso da graduação, coeficientes de rendimentos.

Sousa et al. (2015) utilizam notas do Enem relacionadas com habilidades acadêmicas e específicas de acordo com o modelo psicométrico de Cattell-Horn-Carroll (CHC - desempenho acadêmico envolve habilidades cognitivas e um domínio acadêmico específico), demonstrando que o processo do vestibular serve como preditor de desempenho para o primeiro ano universitário e que a nota da redação é boa preditora para o processo seletivo. **Evidências**: notas do Enem, do vestibular e do curso.

Ferreira (2016) propõe o modelo "Md-pread", usado para a previsão de grupos de reprovação em um ambiente EAD. Ele realiza um relatório semanal para um sistema de recomendação educacional, utilizando o classificador J48, considerado como o melhor entre os algoritmos testados. **Evidências**: dados pessoais, curso, matriz curricular, período letivo inicial, renda familiar, tipo de escola de origem, turma, hora da interação no AVA, IP, ação realizada no AVA, informação (nome do professor, disciplina, atividade, recurso) e média das atividades.

Trabalhos internacionais:

Myller et al. (2002) realizam a previsão para disciplina introdutória de programação, a partir de uma análise das habilidades/conhecimentos aplicados nas questões dos exercícios práticos e das provas. Foi criada uma fórmula que atribui pesos aos exercícios conforme a habilidade necessária para a sua resolução, como por exemplo o uso de *if* ou *loop*, métodos/funções, estrutura básica de dados, etc. Os pesos dos exercícios práticos são considerados para uma semana de exercícios, devido a uma limitação na plataforma que não armazena as notas individuais de cada atividade. O objetivo, além de realizar a previsão sobre a aprovação ou não do aluno, é conseguir descobrir quais habilidades são necessárias para ir bem nas provas e quais delas são consideradas significativas durante o aprendizado de programação. Há também a aplicação de algoritmos de agrupamento, para sugestão de grupos de estudo, conforme as habilidades de cada aluno. **Evidências**: habilidades/conhecimentos das questões de exercício prático e de provas.

Minaei e Punch (2003) demonstram duas estratégias viáveis de otimização para a classificação do desempenho dos alunos, a combinação de múltiplos classificadores e o uso de algoritmos genéticos para selecionar evidências através do reconhecimento de padrões. Em todos os casos gerados para o experimento, é verificado que há uma melhoria significativa da

precisão com o uso da combinação dos classificadores. Da mesma forma, há melhoria na previsão de 10% com o uso de algoritmos genéticos. Foram analisados 12 exercícios de casa, incluindo 184 problemas resolvidos por 261 estudantes (alunos que não realizaram as provas são excluídos da análise). **Evidências**: taxa de sucesso, sucesso na primeira tentativa, número de tentativas antes da resposta correta, tempo em que o aluno obteve o problema correto em relação a data final de entrega do exercício, tempo total gasto com o problema e número de interações on-line do aluno tanto com outros alunos quanto com o instrutor.

Kotsiantis et al. (2003) executam um estudo comparativo entre seis algoritmos, utilizados para prever o abandono dos estudantes na disciplina de introdução a informática em um curso EAD. Na conclusão do experimento têm-se que o algoritmo Naive Bayes (devido a acurácia de 82,89% e significância estatística) é o mais apropriado para a previsão e que as informações obtidas durante o curso são melhores evidências que os dados demográficos. **Evidências**: dados demográficos, participação nas atividades (quatro trabalhos e quatro reuniões presenciais, que são opcionais) e notas nas avaliações.

Hämäläinen e Vinni (2006) realizaram a continuação de um estudo de 2004 [Hämäläinen et al. 2004] que utiliza os resultados de exercícios e notas finais, das disciplinas de programação 1 e 2 (EAD), para a comparação de cinco técnicas de previsão. Os autores concluem que o classificador *Naive Bayes* foi o melhor para prever potenciais alunos desistentes. **Evidências**: resultados de exercícios e notas finais.

Dekker e Vleeshouwers (2009) aplica técnicas de Mineração de Dados para identificar alunos que não obtiveram sucesso (abandono ou reprovação) no primeiro ano da graduação de engenharia elétrica. Os autores concluem que dados pré-universitários e do primeiro semestre podem ser úteis para a previsão. Os autores salientam que disciplinas já conhecidas por atrasar o avanço dos alunos demonstraram ter maior preditor de sucesso para a classificação. **Evidências**: são apresentadas em um apêndice que contém apenas uma tabela, com o nome do atributo, tipo e observação. Não há uma descrição que ajude a entendê-las.

Huang (2011) analisa uma disciplina do curso de engenharia durante quatro semestres e gera 24 modelos de previsão, através de quatro algoritmos e da combinação entre todas as evidências, independentemente da significância estatística. Os modelos criados demonstram que o algoritmo utilizado tem pouca importância para a previsão. O mais importante são as evidências utilizadas, no caso, aquelas que caracterizam o conhecimento prévio dos alunos.

Evidências: média global do curso (GPA), notas das provas parciais, nota final e notas das disciplinas pré-requisito.

Zafra et al. (2011) utilizam uma abordagem baseada em *Multiple Instance Learning* (MIL) para melhorar a previsão no desempenho acadêmico dos estudantes, em detrimento do uso de algoritmos clássicos. **Evidências**: *quizzes*, atividades realizadas e participações nos fóruns.

Márquez et al. (2013) propõe um algoritmo para evoluir classificadores baseados em regras usando programação genética. Para isso, realiza vários experimentos com o objetivo de descobrir regras relacionadas ao problema de evasão em um determinado curso. A base de dados utilizada possui 77 atributos de 670 alunos e os experimentos variam de acordo com a utilização desses atributos e balanceamento ou não dos dados. Para a avaliação do desempenho da classificação são utilizadas outras métricas além da acurácia: sensibilidade (taxa de verdadeiro positivo), especificidade (taxa de verdadeiro negativo) e média geométrica (indica equilíbrio entre os desempenhos da classificação em classes desbalanceadas). Por fim, os autores concluem, em vista de todos os experimentos executados, o quão difícil é realizar a tarefa de previsão do desempenho dos alunos. Eles conseguem excelentes resultados finais, como uma acurácia de 97,3%. **Evidências**: questionário específico (dados pessoais, turma, quantidade de alunos por turma, quantidade de amigos, espaço utilizado para estudar, escolaridade dos membros da família, nível de atenção durante as aulas, etc), questionário geral (dados pessoais, informações sobre dificuldade e tempo gasto na realização de exercícios, notas de um "pré-teste", etc) e notas finais de várias disciplinas.

Romero et al. (2013) tem o objetivo de construir uma ferramenta de Mineração de Dados Educacionais dentro do ambiente virtual de aprendizagem Moodle. Para isso, inicialmente, são implementados quatro algoritmos que realizam a previsão do desempenho de alunos em sete cursos de engenharia. Os autores concluem que não há, em geral, nenhum algoritmo que obtenha uma melhor classificação com a utilização de todos as evidências, mesmo com o uso de técnicas de pré-processamento dos dados (filtragem, discretização ou balanceamento), por isso a acurácia de 65% é apresentada como uma "média" entre os algoritmos. Mas acreditam que adição dos dados de mais alunos e mais informações off-line sobre eles pode ajudar. As informações off-line seriam sobre atendimentos em sala de aula, pontualidade, participação, atenção, predisposição, etc. O mesmo autor deste estudo

apresentou um trabalho semelhante em 2008 [Romero et al. 2008]. **Evidências:** *quizzes*, atividades realizadas e participações nos fóruns.

Moradi et al. (2014) apresentam uma abordagem de divisão multi-canal para a análise das evidências. Cada evidência possui um nível de desempenho com base nas atribuições relacionadas a elas. Por exemplo, de maneira geral, verifica-se qual o nível de desempenho (especialista, bom, médio e fraco) do aluno para cada evidência, como a realização de "n" exercícios práticos ou "n" *quizzes*. Cada nível de desempenho "intermediário" encontrado pode ser utilizado para a previsão do desempenho "final". Não há uma comparação entre previsões realizadas com e sem multi-canais e a acurácia é apresentada para cada nível de desempenho final. **Evidências**: exercícios práticos, *quizzes* e projetos.

Bayazit et al. (2014) consideram o trabalho importante por utilizar estatística descritiva, diferente dos pesquisadores que nas últimas décadas tem preferido investigar relações não-lineares entre variáveis. Os autores utilizam informações de um rastreador de olhos durante a realização de um teste (com o uso de *tablet*), aplicado especificamente para o experimento. **Evidências**: métricas relacionadas a movimentação dos olhos, clicks no mouse e um teste com 9 questões.

Os trabalhos nacionais apresentados estão em busca de melhores evidências e abordagens de como utilizá-las [Souto & Duduchi 2009; Martins et al. 2012; Souza et al. 2015], que os trabalhos internacionais estão preocupados em testar abordagens de otimização dos classificadores [Minaei & Punch 2003; Zafra et al. 2011; Márquez 2013; Moradi et al. 2014], que já existem alguns autores tentando evoluir suas pesquisas [Hämäläinen & Vinni 2006; Romero 2013] e que informações vindas de sensores estão sendo utilizadas como evidências [Bayazit et al. 2014]. Todos os trabalhos apresentados também são importantes para perceber que:

> (...) não existe um algoritmo que sempre mostre desempenho superior aos demais para qualquer Base de Dados. Muitos autores consideram a Mineração de Dados mais uma arte que uma ciência, porque via de regra é preciso certo traquejo do operador e uma boa dose de experimentos empíricos para melhorar os resultados práticos.
>
> (Quilici-Gonzalez 2015, p. 103)

Por isso, ao analisar os Quadros informativos 2 e 3 sobre as abordagens, técnicas/algoritmos com melhores acurácias obtidas e ferramentas utilizadas, é necessário ter em mente que os modelos construídos são focados em pesquisas específicas e que não há um método de classificação mais apropriado a uma determinada tarefa [Hämäläinen & Vinni 2006].

Quadro 2. Quadro informativo dos trabalhos nacionais.

Referência	Técnicas/Algoritmos	Acurácia	Ferramenta
[Souto & Duduchi 2009]	Regressão	40%	-
[Martins et al. 2012]	Redes Bayesianas, OneR e NNge	-	Weka
[Santos et al. 2012]	K-means, REP Tree e J48	-	Weka
[Brito et al. 2014]	Naive Bayes	75,00%	Weka
[Guércio et al. 2014]	Random Forest, Random Tree e J48	73%	Weka
[Brito et al. 2015]	Decision Table	86,90%	Weka
[Manhães 2015]	Naive Bayes (média dos experimentos)	80%	Weka
[Souza et al. 2015]	Regressão linear	-	Minitab XLStat
[Ferreira 2016]	W-J48	84%	RapidMiner

Quadro 3. Quadro informativo dos trabalhos internacionais.

Referência	Técnicas/Algoritmos	Acurácia	Ferramenta
[Myller et al. 2002]	Regressão Linear	-	Ausente
[Minaei & Punch 2003]	C5.0, CART, QUEST, CRUISE, Quadratic Bayesian, 1-NN, k-NN, Parzen-Window e MLP	95%	MATLAB e outro software não especificado.
[Kotsiantis et al. 2003]	Redes Neurais	83,89%	-
[Hämäläinen & Vinni 2006]	Naive Bayes	80%	-
[Dekker & Vleeshouwers 2009]	J48 e RandomForest	80%	Weka
[Huang 2011]	Regressão Linear Múltipla (MLR), Multi-layer Perceptron (MLP), Rede Neural (RBF) e Máquina de Vetores de Suporte (SVM)	>80%	SPSS 18 (MLR, MLP e RBF) MATLAB (SVM)
[Zafra et al. 2011]	*Multiple Instance Learning*: métodos baseados em aprendizado supervisionado simples	73%	-
[Márquez et al. 2013]	ADTree	97,30%	Weka
[Romero et al. 2013]	Árvore de Decisão, Regras de Indução, Lógica Fuzzy, Métodos Estatístico e Redes Neurais	65%	Programação em Java com auxílio do KEEL framework
[Moradi et al. 2014]	CHAID	87,50%	Weka
[Bayazit et al. 2014]	Random Forest (acurácia calculada para cada questão de um teste)	57 a 78%	-

O Quadro informativo 2, apresenta os trabalhos nacionais e o Quadro informativo 3 apresenta os trabalhos internacionais. Optou-se por informar apenas a técnica ou algoritmo

que obteve a melhor acurácia. Quando houver mais de uma técnica/algoritmo, significa que a acurácia foi obtida através de uma média de todos experimentos. A ausência da acurácia quer dizer que a pesquisa utilizou alguma outra métrica de avaliação.

A utilização de técnicas de mineração de dados aplicada a educação, ainda pode ser vista como um assunto recente. Há dúvidas, inclusive, sobre quais evidências devem ser utilizadas e sobre quais técnicas são mais adequadas [Baker et al. 2011].

Romero e Ventura (2013) falam sobre os educadores e instituições desenvolverem uma cultura baseada em dados para melhorar o ensino, com tomada de decisões de forma embasada. Além disso, os autores frisam que os resultados das pesquisas em EDM, geralmente são alcançados em contextos limitados e projetos específicos. Sendo necessário obter resultados mais gerais, como por exemplo, o uso das mesmas evidências de um modelo com diversos estudantes com outras configurações escolares ou até mesmo, o uso confiável de um modelo preditivo em um contexto diferente. Os autores apontam uma necessidade crescente de estudos de replicação para testar generalizações que promovam o intercâmbio de dados e modelos. Para esses avanços acontecerem, é necessário a criação de repositórios abertos e formatos de dados padrão.

Tendo em vista o cenário apresentado, os trabalhos relacionados inspiraram esta dissertação em busca de evidências que pudessem realizar uma boa previsão sobre o desempenho dos alunos. Utilizar as mesmas técnicas/algoritmos seria inviável, como discutido nos parágrafos acima, mas as estratégias sobre otimização de classificadores podem ser replicadas em trabalhos futuros, uma vez que a busca e análise por evidências sejam imprescindíveis para um primeiro estudo.

4 Metodologia

A criação de um modelo de previsão necessita da definição da variável dependente (o que se deseja prever), das variáveis independentes (as evidências), do método de previsão a ser utilizado, dos algoritmos e da avaliação do modelo. Para a disciplina de Introdução à Programação de Computadores (IPC) da UFAM, as notas parciais foram selecionadas para compor a variável dependente.

A disciplina de IPC é dividida em sete módulos e cada módulo possui uma prova parcial. A previsão das provas parciais por módulo se faz importante por determinar sequencialmente o desempenho dos alunos e o professor pode intervir antes que ocorra a reprovação na disciplina. As notas parciais obtidas pela correção automática do *CodeMeistre* geram no máximo quatro tipos diferentes de notas (0, 3.3, 6.6, 5 e 10) para cada questão presente na avaliação. Um exemplo de correção pode ser visto no Apêndice A.

Devido a essa discretização das notas, a classificação foi o método escolhido para a previsão juntamente com os algoritmos que envolvem Floresta Aleatória, Regressão Logística e Máquina de Vetores de Suporte, de acordo com Baker (2010), esses algoritmos são os mais populares na tarefa de classificação, assim como a validação cruzada.

As variáveis independentes ou evidências são detalhadas na Seção 4.1 (devido à quantidade de evidências, totalizando em 19). Elas foram selecionadas a partir de exercícios de codificação, de múltipla escolha e de conhecimento prévio, que ficam armazenados em duas bases de dados de acordo com as ferramentas de apoio à disciplina de IPC:

- ***CodeMeistre***: como dito anteriormente, é um juiz online desenvolvido na UFAM. Um juiz online é um sistema de correção automática de código-fonte que recebe os códigos desenvolvidos pelos alunos, como resposta a um exercício de programação e para avaliá-lo, executa um código de comparação. Essa análise de corretude é feita com a saída retornada do código do aluno, com a saída esperada pelo problema do exercício (apêndice A). O *CodeMeistre* consegue guardar vários "rastros" de uso relacionados a construção de códigos, como quantidade de execuções, execuções com sucesso, digitação, trechos copiados, etc.
- ***ColabWeb***: as evidências são provenientes do uso da ferramenta *quiz*, usada tanto para realização de exercícios, como para questionários sobre

conhecimento prévio da disciplina. Ele dispõe de uma tabela de *log* que mantém alguns registros de utilização como a quantidade de tentativas de resolução dos *quizzes*, o tempo para realizá-los, número de revisões realizadas, etc.

Atualmente, a disciplina de IPC utiliza apenas a ferramenta *CodeMeistre*, então é analisado se as evidências do *ColabWeb* causam algum diferencial na previsão de desempenho dos alunos, assim como evidências de conhecimento prévio sobre a disciplina. A princípio, para essas configurações de evidências, as notas parciais são analisadas como um todo e posteriormente, a partir da configuração com melhor acurácia, analisam-se as provas parciais, por módulo (há uma melhor explicação sobre as configurações no Capítulo 6, por tratar do estudo de caso).

4.1 Evidências

As evidências selecionadas de acordo com cada base de dados, foram:

- ***CodeMeistre***
  - ***nota_lista***: média da nota dos exercícios práticos realizado a cada laboratório de codificação;
  - ***numero_acessos***: número de logins realizados durante o prazo de entrega dos exercícios;
  - ***tempo_uso_ide***: tempo total em minutos que o aluno permaneceu logado no *CodeMeistre* tentando solucionar (digitando alguma coisa) os exercícios;
  - ***media_submissao_lista***: número total de submissões feitas durante as tentativas de solucionar todas as questões de uma lista dividido pelo total de número de questões;
  - ***media_teste_lista***: número total de testes feitos durante as tentativas de solucionar todas as questões de uma lista pelo total de número de questões;
  - ***proporção_caracteres_colados***: proporção entre o número de caracteres colados com o número de caracteres digitados, durante as tentativas de solucionar a lista de exercícios;
  - ***velocidade_digitacao***: velocidade com que o aluno digita durante a tentativa de solucionar as questões;

- ***linhas_log***: total de linhas de log gerados durante a resolução de uma questão da lista dividido pelo total de número de questões;
- ***media_delete***: total de caracteres apagados (através da tecla *delete* ou *backspace*) pelo número total de questões da lista;
- ***erro_sintaxe***: proporção entre o número total de submissões e o número total de submissões com erros de sintaxe;
- ***media_sucesso_submissao***: total de submissões com sucesso dividido pelo total de questões da lista;
- ***media_dificuldade***: cada aluno, durante o ano de 2016, tinha que reportar o grau de dificuldade de uma questão (0 para uma questão fácil, 1 para mediana e 2 para difícil);
- ***nota_lista_real***: a média da *nota_lista* passa a considerar apenas os exercícios que geraram no mínimo 50 linhas de log. Exercícios respondidos pelos alunos, ao digitarem diretamente no *CodeMeistre*, geram mais de 50 linhas de log, uma quantidade menor que essa, sugere que o aluno burlou a resposta do exercício.

- ***ColabWeb***
  - ***nota_quiz***: nota de cada *quiz* realizado;
  - ***tempo_resposta_quiz***: tempo que o aluno demorou para responder os *quizzes* de um módulo;
  - ***dias_para_responder_quiz***: contagem dos dias decorridos entre a abertura do *quiz* e o dia em que o aluno começou a respondê-lo;
  - ***reviewed_quiz***: quantidade de vezes que o aluno revisou o *quiz* depois dele ter sido respondido;
  - ***quiz_nivelamento***: conhecimento prévio sobre a disciplina;
  - ***quiz_algebra_precalculo***: conhecimento prévio envolvendo álgebra e pré-cálculo.

Cada nota não apresentada por ausência do aluno foi alterada para apresentar o valor "–1", assim como as variáveis independentes que possuem relação direta com as notas (caso da falta de independência estatística apresentada na seção 2.1). Por exemplo, o aluno que não respondeu o *quiz*, além de não possuir *nota_quiz*, também não possuí informação sobre *tempo_resposta_quiz* e *dias_para_responder_quiz*, de forma que essas variáveis também apresentam valor -1.

4.2 Algoritmos de Classificação

As subseções apresentam uma explicação dos algoritmos considerados como os mais populares de classificação [Baker 2010] e utilizados nesta dissertação. Uma apresentação detalhada não se faz necessária, uma vez que há diversas ferramentas de acesso livre que os implementam, mas é importante conhecer o funcionamento de cada um deles para que seja possível realizar ajustes de parâmetros de forma adequada e uma interpretação correta de seus resultados.

4.2.1. Floresta Aleatória

A floresta aleatória (*Random Forest*) foi desenvolvida por Breiman [Breiman 2001] e pode ser utilizada tanto para a classificação, quanto para a regressão. Ela faz parte do paradigma *ensemble learning* de aprendizado de máquina, onde vários modelos são agrupados com o objetivo de alcançar uma melhor generalização. Sendo assim, as florestas aleatórias são formadas por um conjunto de árvores de decisão.

Uma árvore de decisão é uma representação dos possíveis resultados de uma série de escolhas de uma determinada questão. Por exemplo, na Figura 10 são representadas algumas evidências fictícias [10] obtidas através de "dados históricos", para determinar se um novo aluno será aprovado ou reprovado.

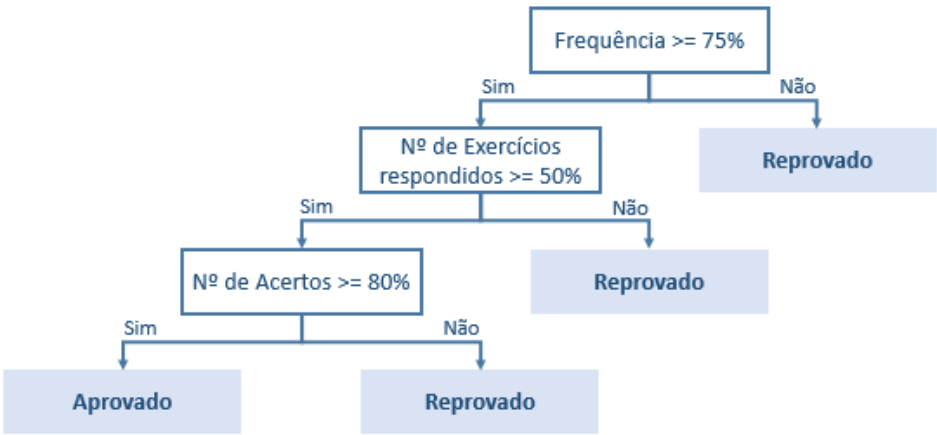


Figura 10. Exemplo de árvore de decisão.

[10] A disposição das evidências na árvore é realizada por meio de técnicas estatísticas (ex.: Índice Gini, Qui-Quadrado, Cálculo de Entropia e Redução de Variância), que identificam quais são as variáveis mais significativas e que divisões devem ser realizadas. O algoritmo particiona um conjunto de dados heterogêneos (raiz) em classes homogêneas (folhas), gerando regras de classificação com base em atributos (nós). Os algoritmos de árvore de decisão mais conhecidos são: ID3, C4.5 e CART [Tan et al. 2009 e Han et al. 2011].

Enquanto uma árvore de decisão comum utiliza todos os dados disponíveis para a construção de uma árvore, o *Random Forest* divide os dados aleatoriamente em subconjuntos e cada subconjunto gera uma árvore com atributos selecionados, também, aleatoriamente. Ou seja, os subconjuntos:

- Possuem *n* instâncias de treinamento, selecionadas de forma aleatória na base de dados;
- Têm *m* atributos/evidências selecionadas aleatoriamente, para a geração de uma árvore (o valor de *m* deve ser menor que o número total de atributos, dessa forma, há a geração de árvores distintas);
- Geram uma árvore através do algoritmo CART (*Classification and Regression Trees*).

A Figura 11 apresenta exemplos genéricos das árvores criadas pelo *Random Forest*.

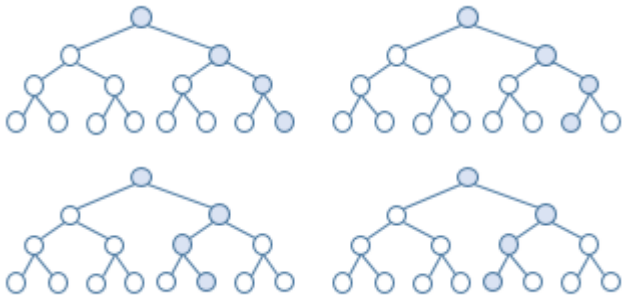


Figura 11. Exemplos de árvores da floresta aleatória.

Cada subconjunto utilizado para gerar uma árvore é formado através de um *bootstrap* com reposição [Han et al. 2011]. Isso quer dizer que um subconjunto pode conter instâncias que já foram incluídas por outros conjuntos e que podem haver instâncias não incluídas em nenhum dos conjuntos. Mas a amostragem por *bootstrap*, garante que 1/3 dos registros são separados e utilizados em testes (exclui a necessidade da validação cruzada).

Com as árvores de decisão formadas, elas são submetidas a testes e recebem um "voto", que é responsável por eleger a árvore com melhor capacidade de previsão. No caso da classificação, é elegida a árvore com mais votos. Na regressão, obtém-se a média das saídas das diferentes árvores. Esse voto é formado através da similaridade entre cada árvore e pela precisão individual de cada uma delas, sendo desejável uma menor similaridade entre duas árvores. A ideia é manter a precisão das árvores sem aumentar a sua similaridade [Han et al. 2011].

As implementações do *Random Forest* geralmente possuem alguns parâmetros que podem ser ajustados para um melhor desempenho, mas eles variam de acordo com ferramentas e bibliotecas. Segue alguns deles:

- Número máximo de evidências: é possível selecionar o número máximo de evidências a serem consideradas na construção das árvores. Às vezes, um maior número de evidências pode melhorar o desempenho do modelo. Lembrando que isso acarreta, na diminuição da variedade das árvores e da velocidade do algoritmo. Algumas ferramentas permitem a utilização de uma porcentagem para esse parâmetro.
- Número de estimadores: é o número de árvores que se deseja construir. Quanto maior, melhor será a previsão e pior será o desempenho do processador.
- Número de processadores: algumas ferramentas ou bibliotecas, possuem um parâmetro para definir restrições a utilização do processador ou não. Se houver, geralmente, utilizam apenas um processador.
- Estado aleatório: é possível gravar um valor que sempre irá reproduzir uma mesma execução do algoritmo.

4.2.2. Regressão Logística

A Regressão Logística faz parte do paradigma de aprendizado de máquina de modelos funcionais, também conhecido como paradigma do aprendizado estatístico, por utilizar modelos estatísticos. A Regressão Logística, provavelmente, tornou-se reconhecida a partir do trabalho de Truett et al. (1967) que realizou um estudo sobre o risco de doenças coronarianas.

Essa regressão realiza previsões por meio de um modelo matemático que retorna à probabilidade de uma instância pertencer a uma determinada classe. Ela é categorizada de acordo com a variável preditora (dependente) e as variáveis preditivas (independentes). A Regressão Logística simples possui apenas uma variável preditiva; a múltipla possui mais de uma. Nessas duas, a variável preditora é dicotômica, ou seja, possui apenas duas classes (ex.: empréstimo aceito ou não aceito; aluno aprovado ou reprovado). Quando a variável preditora admite mais de duas classes, ela é chamada de politômica e a Regressão Logística utilizada é a multinomial [Hosmer et al 2013].

Geralmente, os algoritmos de Mineração de Dados implementam a Regressão Logística Multinomial e adotam algumas estratégias que podem melhorar a classificação. Por exemplo, o *Simple Logistic* [Landwehr et al. 2005] utiliza as variáveis independentes de forma interativa até que o erro da classificação pare de diminuir, e o *Logistic* [Cessie & Van Houwelingen 1992] integra uma técnica estatística chamada de estimador *ridge*, que auxilia classificações onde há uma forte dependência entre as variáveis preditivas, sem que necessite excluí-las.

4.2.3. Máquina de Vetores de Suporte

Assim como a Regressão Logística, a Máquina de Vetores de Suporte (do inglês Support Vector Machina - SVM), também faz parte do paradigma de modelos funcionais. A teoria matemática dessa abordagem pode ser vista em Cortes e Vapnik (1995) e as ideias fundamentais desse algoritmo são apresentadas de acordo com Quilici-Gonzalez & Zampirolli (2015).

O SVM trata os dados da base de treinamento como sendo vetores de treinamento e os dados da base de teste como vetores de teste. Como visto anteriormente na Seção 2.3.2, a classificação é realizada para dados discretos e podem ser representados tanto por coordenadas como por um vetor, através de um plano cartesiano (Figura 12).

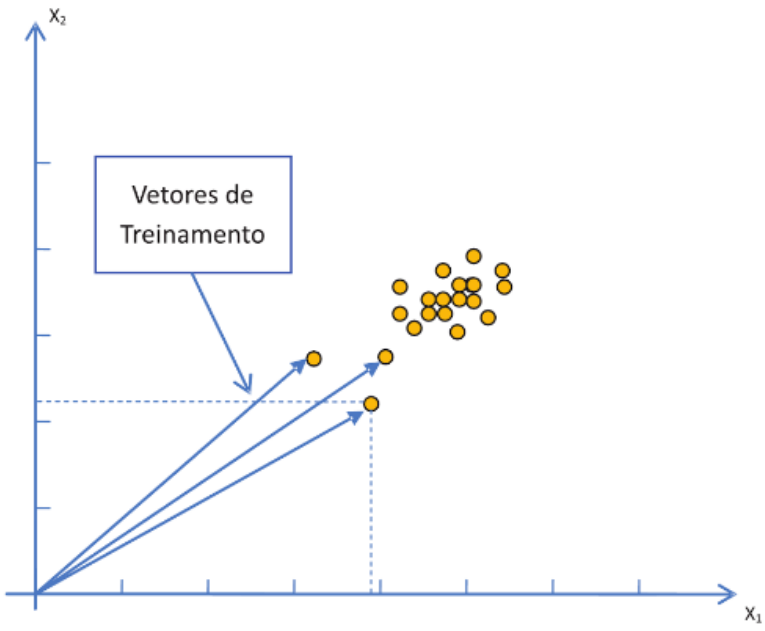


Figura 12. Vetores de treinamento [Quilici-Gonzalez & Zampirolli 2015].

A partir dos vetores de treinamento, o SVM cria retas que dividem as classes conforme suas particularidades. Usar uma reta ou função linear, para separar classes, resulta em um

custo computacional mais baixo do que utilizar uma curva ou funções polinomiais. E também torna menos provável, a necessidade de realizar ajustes no modelo desenvolvido (problema do *overfitting*). Cada reta é conhecida como hiperplano e o objetivo do algoritmo é encontrar um hiperplano ótimo, onde a distância entre ele e as classes sejam semelhantes e a maior possível. Essa distância é chamada de margem máxima, como apresentada na Figura 13. A descoberta da margem máxima proporciona uma maior chance de classificação correta dos vetores de teste, pois é a partir da delimitação do hiperplano que se define o espaço reservado para uma classe e o possível aparecimento de novos dados (diferentes dos utilizados da base de treinamento).

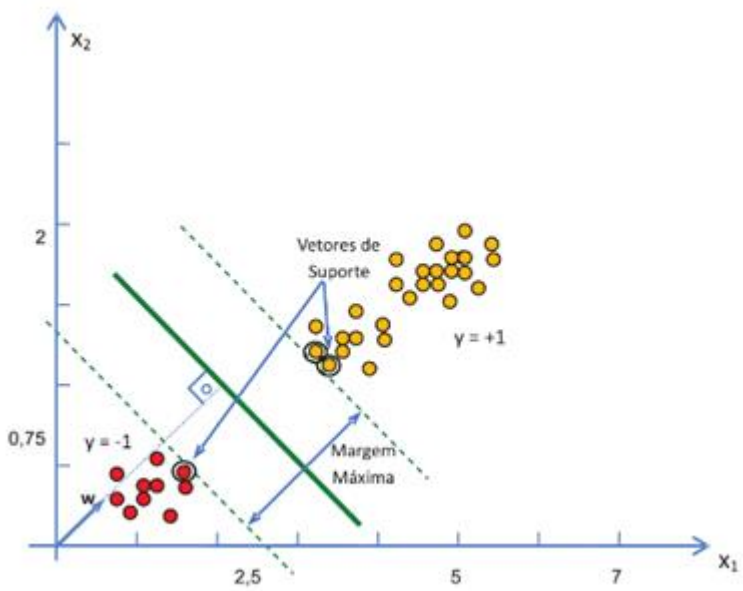


Figura 13. Margem máxima [Quilici-Gonzalez & Zampirolli 2015].

Para encontrar a margem máxima, são utilizados vetores de suporte (Figura 13) e conjuntos convexos (Figura 14). Os vetores de suporte encontram a extremidade do conjunto convexo formado pelas classes e define a margem máxima. Os conjuntos convexos garantem a integridade das classes. Com isso, o modelo aprendido é representado pelo vetor w (vetor peso ou normal) que será sempre perpendicular ao hiperplano (Figura 13).

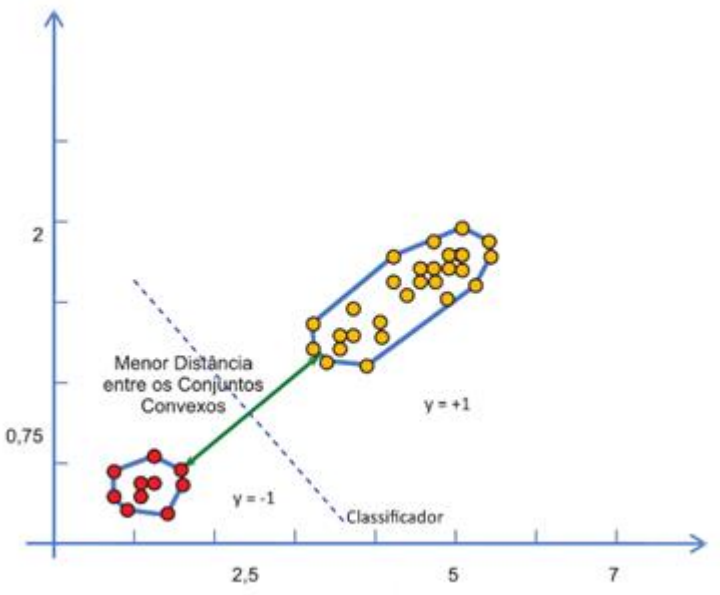


Figura 14. Menor distância entre conjuntos convexos [Quilici- Gonzalez & Zampirolli 2015].

As Figuras 13 e 14 apresentam classes linearmente separadas, mas quando isso não é possível, como apresentado na Figura 15, o SVM mapeia os dados em um outro espaço de maior dimensão, conhecido como espaço de característica. Ele utiliza um recurso matemático conhecido como *Kernel Trick* ("truque do núcleo").

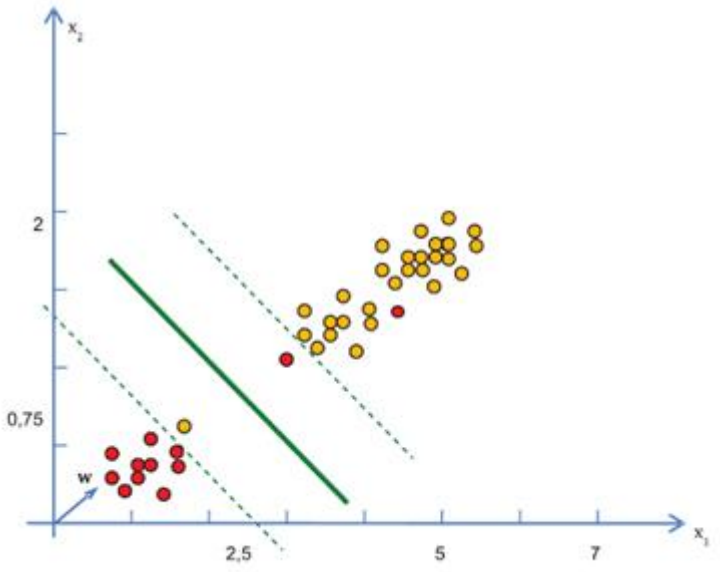


Figura 15. Classes não separáveis linearmente [Quilici- Gonzalez & Zampirolli 2015].

O *Kernel Trick* realiza uma transformação não linear, por exemplo, de um espaço bidimensional (espaço de entrada) para um tridimensional (espaço de características). Onde se torna possível a divisão das classes, de forma linear (Figura 16).

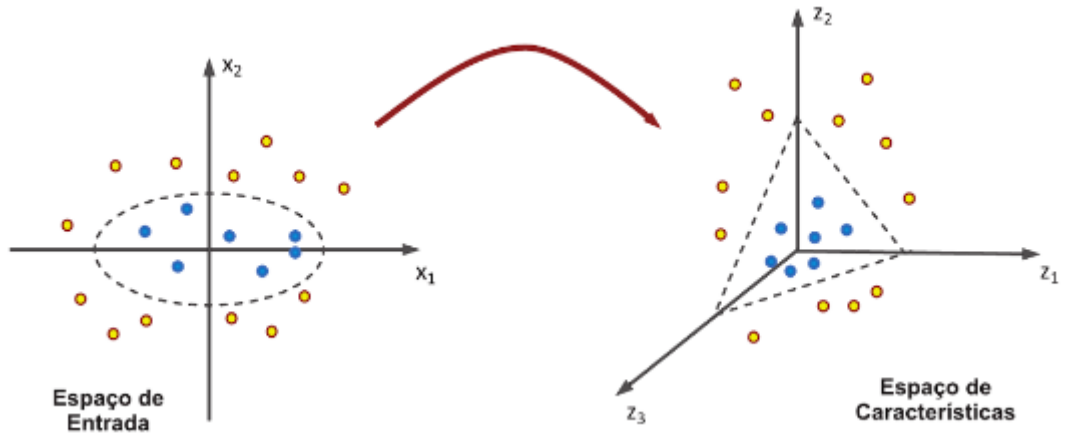


Figura 16. Transformação de um espaço bidimensional para tridimensional [Quilici- Gonzalez & Zampirolli 2015].

Não há criação de novos dados, temos apenas o acréscimo de redundâncias às coordenadas. As variáveis z1, z2 e z3 aparecem apenas para ajudar no entendimento da transformação, não sendo necessário computar a transformação gerada. Para efeitos práticos, não há efetivamente o aumento da dimensão. O produto interno entre dois vetores no espaço característica pode ser computado como se ainda estivesse no espaço de entrada.

O *Kernel Trick* possibilita a criação de quantas dimensões forem necessárias para a resolução de um problema. É possível desenvolver *kernels* que aumentam a taxa de sucesso de um modelos específicos para alguns domínios, como no caso da biologia ou para a classificação de imagens ou caracteres.

Mesmo com a utilização do *Kernel Trick*, há casos em que as classes não conseguem ser separadas linearmente. Para elas, o SVM adota uma "margem suave", que admite ruídos. Essa margem é construída a partir de uma constante C, conhecida como parâmetro de complexidade ou de regularização. O valor de C é descoberto empiricamente, a partir da validação cruzada. Ele é ajustado entre a maximização da margem do modelo e a minimização dos erros de treinamento. Quanto maior for o valor de C, menor será a ocorrência de erros, resultando também, em uma redução da margem do hiperplano. Na Figura 17 temos o modelo de classificação ajustado com base em um valor de C alto, representado pelo tracejado vermelho. Note que dessa forma, há menos pontos classificados erroneamente.

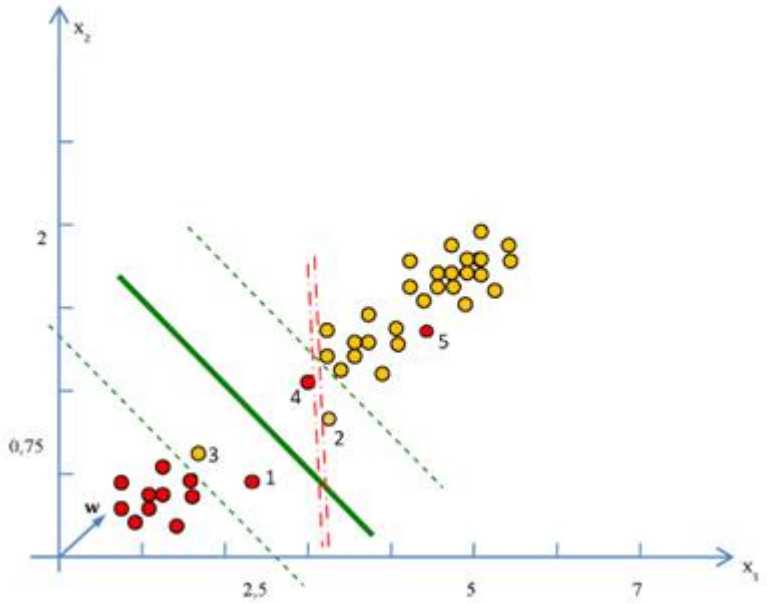


Figura 17. Diferentes margens com diferentes capacidades de generalização [Quilici-Gonzalez & Zampirolli 2015].

Uma das implementações do algoritmo do SVM, incorporadas ao Weka, é proposta por Platt (1998) e conhecida como SMO (*Sequential Minimal Optimization*). A diferença entre ela e o SVM está na utilização dos vetores de treinamento. O SVM considera todos os vetores conjuntamente. Já o SMO os divide em pares e seus valores ótimos são deduzidos analiticamente. Dessa forma, os vetores não precisam ser carregados simultaneamente na memória principal do computador, reduzindo as chances de uma sobrecarga com aplicações reais, com milhares e milhares de vetores.

5 Estudo de Caso

5.1 Contextualização

A disciplina de IPC na UFAM é ministrada para quatorze cursos de engenharia e ciências exatas. Atualmente a metodologia de ensino é híbrida [Carvalho et al. 2016], e as aulas se dividem em dois momentos:

- **Presencial**: o professor é o elemento central da aula, ele apresenta o conteúdo aos alunos, que dispõem de computadores para acompanhar os slides e executar exercícios de exemplo. As avaliações também são presenciais e realizadas através do computador.
- **Online**: o aluno tem a flexibilidade para decidir onde irá realizar os exercícios práticos, que pode ser em laboratório ou em qualquer outro lugar que tenha um computador conectado à internet. Os exercícios são liberados após a apresentação do conteúdo e podem ser realizados até a data da avaliação. O aluno que decidir realizá-los em laboratório, sabe que lá terá o apoio de um tutor e fácil acesso ao professor.

A cada semestre, os professores da disciplina discutem que estratégias podem adotar para diminuir a reprovação. Dentre elas, temos por exemplo:

- O uso do *ColabWeb* para realização de *quizzes* conceituais;
- Criação do *CodeMeistre* (juiz online);
- Desistência do uso do *ColabWeb* e consequentemente dos *quizzes;*
- Uso de gamificação.

As aulas são construídas de maneira colaborativa entre os professores e todas as turmas recebem o mesmo conteúdo, que é dividido em sete módulos. A cada módulo, os alunos resolvem exercícios de fixação antes de realizarem uma avaliação parcial.

O objetivo da disciplina, de acordo com a sua ementa, é:

> "Auxiliar os alunos a aprenderem a resolver problemas algorítmicos. Oferecer a capacidade de elaborar, verificar e implementar algoritmos em uma linguagem de programação de alto nível. Ao final da disciplina os alunos deverão estar aptos a elaborar programas para manipular estruturas de dados básicas armazenadas em memória principal".

Em 2014, quando o ensino era apenas presencial e as atividades eram realizadas sem um juiz online, 50% dos alunos conseguiam obter a aprovação na disciplina (considerando apenas os cursos que ministravam IPC no primeiro semestre). Em 2015, com a implantação do *CodeMeistre* e da metodologia híbrida em quatro das quatorze turmas do primeiro semestre, houve um aumento de 20% na aprovação [Carvalho et al. 2016]. Com o *CodeMeistre*, é possível:

- Disponibilizar mais atividades para os alunos, melhorando a aprendizagem através da prática;
- Fornecer *feedback* automático das atividades respondidas por eles;
- Diminuir o trabalho dos professores com correções;
- Dar autonomia aos alunos para conciliar a disciplina de IPC com outras de maior interesse.

A atual metodologia da disciplina de IPC é inspirada no trabalho de Píccolo (2010). No primeiro dia de aula, ela é apresentada da seguinte forma:

- Ponto de vista do estudante
  - É uma metodologia voltada à prática de programação.
- Ponto de vista do professor
  - Padronização metodológica favorece a divisão de tarefas;
  - Divisão de tarefas reduz tempo e trabalho de planejamento;
  - Laboratórios são cobertos pelos tutores;
  - Aulas podem ser cobertas por outro colega.

A linguagem de programação adotada na disciplina é o *Python 3.x* e o conteúdo é dividido em sete módulos:

1. Variáveis e programação sequencial
2. Condicional simples (*if-then*) e composta (*else*)
3. Condicional encadeada (*elif*) e aninhada
4. Repetição por condição
5. Vetores e s*trings*
6. Repetição por contagem
7. Matrizes

Cada módulo é composto por uma aula teórica, duas aulas de exercício prático e uma aula de avaliação:

- **Aula teórica**: presencial e ministrada pelo professor. Para todas as turmas, os conteúdos e slides da disciplina são os mesmos, por isso, o professor pode facilmente ser substituído por um colega.
- **Exercício prático**: também é o mesmo para todos os alunos. Ele é usado na composição da nota e pode ser feito a distância, mas quando o aluno decide realizá-lo presencialmente, há o acompanhamento de um tutor. Os tutores estão presentes durante todos os dias de aula e geralmente são alunos do Programa de Pós-Graduação em Informática (PPGI) do Instituto de Computação.
- **Avaliação**: presencial, com supervisão do professor e apoio do tutor. Compõe a maior parte da nota do aluno e é realizada através do computador, por meio de um juiz online. Este possibilita sortear questões aleatoriamente, dentro de algumas possibilidades de problemas. Por exemplo, a primeira questão pode possuir cinco problemas diferentes[11] cadastrados e um deles será escolhido aleatoriamente para ser apresentado ao aluno, como a questão um. O objetivo é diminuir a chance de resoluções compartilhadas entre os alunos durante a avaliação, que é realizada de forma individual e sem pesquisa.

Os exercícios práticos e as avaliações são os que mais sofreram modificações com o passar dos semestres. Atualmente, eles são realizados através do *CodeMeistre* e por isso, os exercícios práticos passam a se chamar de laboratório de codificação, já que todos os exercícios são desenvolvidos no *CodeMeistre*. Em 2015 e 2016 as duas aulas reservadas para os exercícios práticos eram divididas da seguinte forma:

- **Laboratório de codificação**: realizado no *CodeMeistre*, com problemas que podem ser do domínio da matemática, física, química, financeiro, jogos, estatística, entre outros. A resolução de tais problemas exige habilidades de criação e complementação de códigos.
- **Quiz**: realizado no *ColabWeb*, com questões de múltipla escolha que envolvem o assunto principal do módulo e de cunho conceitual. Nele, as habilidades de rastreamento, correção de erros, completamento de lacunas e ordenamento de código eram exploradas em diferentes graus de dificuldade.

A nota final da disciplina e posterior situação do aluno não é analisada neste trabalho,

[11] As diferenças entre os problemas são sutis e não interferem (“fator sorte”) na nota dos alunos. A lógica dos problemas é a mesma, o que muda por exemplo, é a sua apresentação através da alteração de alguns nomes, da ordem dos itens, etc.

mas é importante saber como ela é calculada. São sete avaliações parciais, uma prova final, quatorze exercícios entre laboratórios práticos e *quizzes*:

- 7 Avaliações Parciais (A1, A2, A3, A4, A5, A6 e A7)
- 7 Laboratórios de Codificação
- 7 Laboratórios de *Quizzes*
- 1 Prova Final (PF)

A Média dos Laboratórios (ML) é calculada através da média aritmética de todos os laboratórios (codificação e *quizzes*).

A Média Parcial (MP) é calculada da seguinte maneira:

$$MP = \frac{(A1 + A2) * 1 + (A3 + A4 + A5) * 2 + (A6 + A7) * 3 + ML * 2}{16}$$

A Média Final (MF) é calculada pela fórmula:

$$MF = \frac{2 * MP + PF}{3}$$

O aluno é considerado aprovado caso a MF seja maior ou igual a cinco e tenha frequência superior a 75% do total de aulas, presenciais e virtuais. Caso contrário, é considerado reprovados.

5.2 Cenário

O estudo de caso foi realizado com base na disciplina de IPC durante o primeiro semestre de 2016, com dados de 338 alunos. Outros 110 alunos, foram desconsiderados, por ausência de registro no *CodeMeistre*, o que significa que esses alunos não realizaram nenhum exercício prático ou avaliação parcial.

As evidências foram coletadas por módulo da disciplina, ou seja, cada aluno possui sete linhas de registros, totalizando 2366 instâncias. A variável dependente é representada pela palavra “conceito” e é dividida, no máximo, em três classes (Figura 18):

- Satisfatório: quando a nota da avaliação é maior e igual a 5.
- Insatisfatório: quando a nota da avaliação é menor que 5.
- Sem conceito: quando o aluno não apresenta nota para a avaliação.

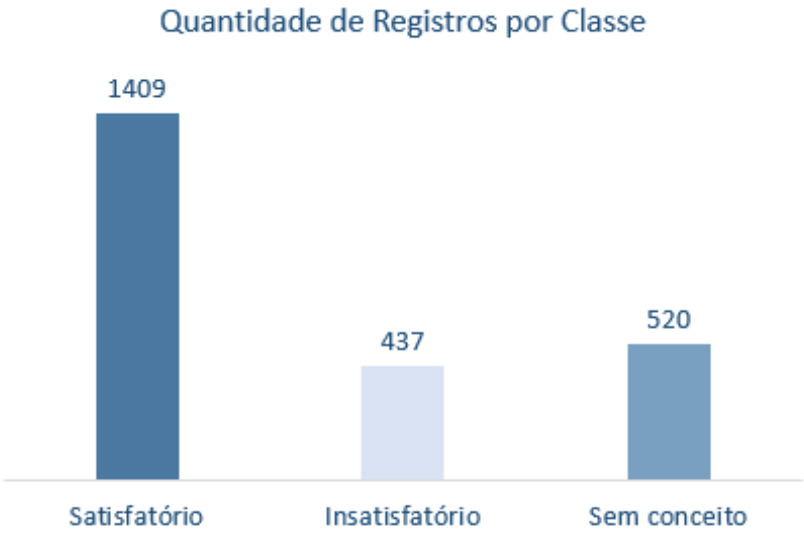


Figura 18. Quantidade de registros por classe.

O gráfico da Figura 18 apresenta que mais da metade das avaliações parciais possuem conceito satisfatório, o que não reflete no número de aprovações na disciplina. É importante lembrar do cálculo para a realização da média final, 1409 registros satisfatórios equivalem, em média, a quatro questões com nota acima de cinco para cada aluno, o que não garante a aprovação. Para a aprovação ocorrer, sem considerar a frequência do aluno, é necessário que ele obtenha, no mínimo, a nota cinco em todas as avaliações, laboratórios e prova final.

A Tabela 1 mostra a quantidade de atividades ou avaliações não realizadas por algum aluno. A linha e coluna "alunos", funciona como uma interseção dos dados e indica, por exemplo, que 9 alunos não realizaram nenhuma das atividades ou avaliações para o módulo 1 ou que 81 alunos não realizam nenhum dos *quizzes*.

Tabela 1. Quantidade de atividades ou avaliações não realizadas por módulo.

	1	2	3	4	5	6	7	Total	Alunos
Quiz	61	60	68	64	66	61	60	440	81
Lab. Codificação	10	21	23	32	45	56	73	260	165
Avaliação	50	33	54	75	94	100	114	520	85
Alunos	9	8	13	18	26	30	34	138	41

É interessante perceber como a quantidade de alunos ausentes por *quiz* é praticamente constante, em todos os módulos, enquanto a ausência nas avaliações é crescente a partir do módulo 2 e os laboratórios de codificação desde o módulo 1.

5.3 Previsão

O processo geral do estudo de caso pode ser visto na Figura 19:

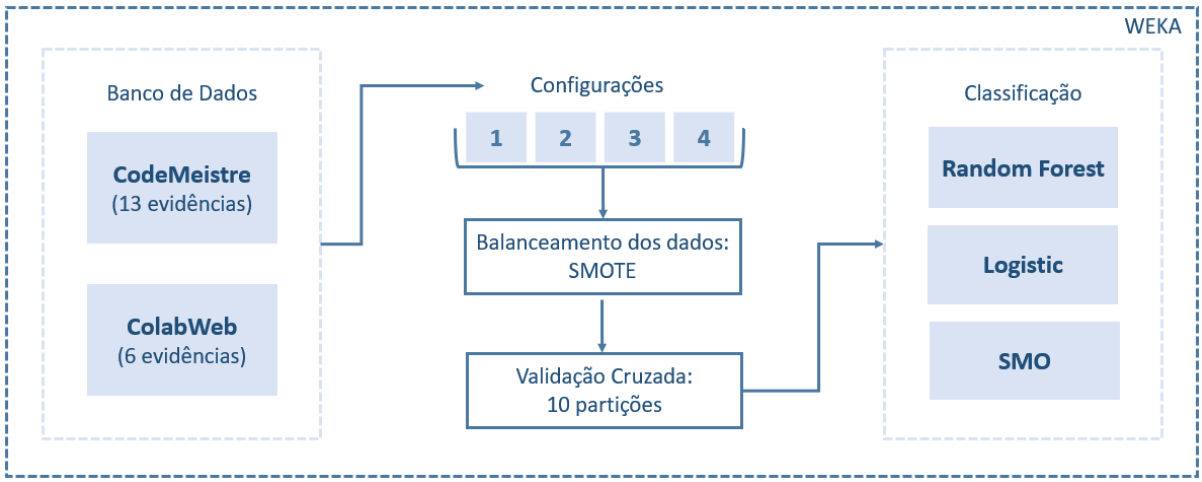


Figura 19. Processo geral do estudo de caso.

A previsão dos conceitos a partir das avaliações parciais, foi realizada por meio da classificação com o apoio da ferramenta Weka, utilizando três algoritmos: *RandomForest* (RF), *Logistic* (L) e o *Sequential Minimal Optimization* (SMO). Para o balanceamento dos dados, o algoritmo SMOTE (*Synthetic Minority Oversampling Technique*) [Chawla et al. 2002] foi escolhido. Todos os parâmetros foram utilizados com valores padrão do Weka, com exceção do SMOTE, que precisa receber a porcentagem de dados sintéticos a serem criados. A validação cruzada é *default* do Weka e com 10 partições.

O estudo de caso considera quatro "configurações" gerais de evidências para analisá-las a fim de determinar qual configuração pode fornecer uma boa previsão, além de uma posterior análise por módulos. Todas elas consideram o uso ou não das evidências de conhecimento prévio (através das evidências *quiz_nivelamento* e *quiz_algebra_precalculo*). Segue as configurações:

- **Configuração 1**: apenas as evidências *nota_lista* e *nota_quiz*;
- **Configuração 2**: evidências do *ColabWeb*;
- **Configuração 3**: evidências do *CodeMeistre*;
- **Configuração 4**: evidências do *CodeMeistre* e do *ColabWeb*.

Todas as configurações utilizam os mesmos dados afim de prover uma melhor validação das acurácias, uma vez que o SMOTE cria novos dados a partir de seus vizinhos

(Seção 2.3.1). Cada configuração é executada e analisada de acordo com a quantidade de classes da variável independente “conceito” :

- **Politômica**: com as três classes apresentadas (satisfatório, insatisfatório e sem conceito) na Seção 5.2;
- **Dicotômica**: com duas classes (satisfatório e insatisfatório), onde a classe “satisfatório” permanece igual e a classe “insatisfatório” é acrescida da classe “sem conceito”.

A previsão por módulos também é realizada de acordo com as classes apresentadas e a partir da configuração de evidências com a melhor acurácia. Os módulos são selecionados com os dados desbalanceados e, em seguida, são balanceados por módulo. A Figura 20 apresenta por módulo, a quantidade de alunos divididos pela variável “conceito”.

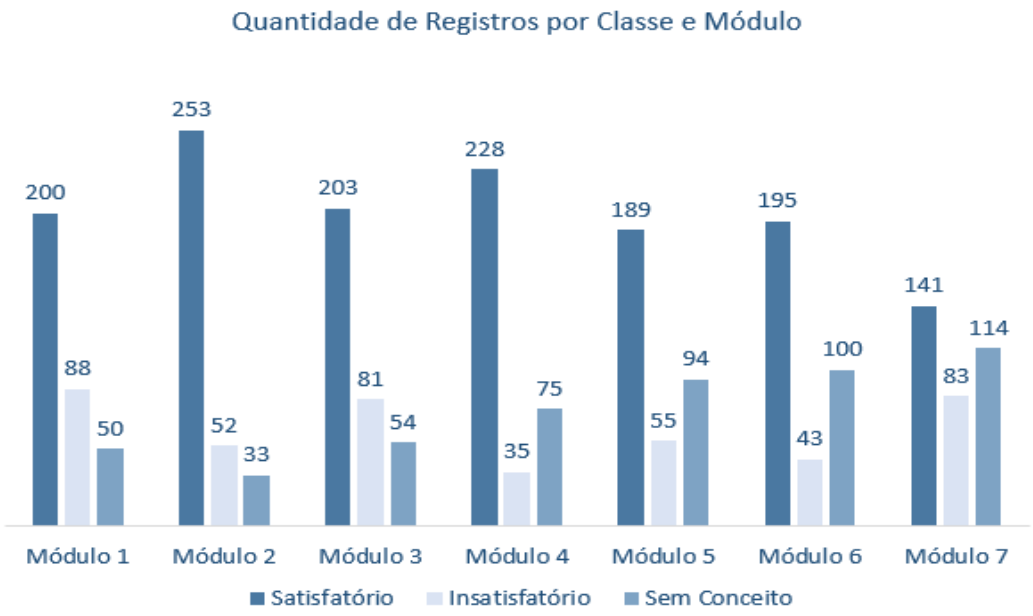


Figura 20. Quantidade de registro por classe e módulo.

5.4 Resultados

As Tabelas 2 e 3 apresentam as acurácias das previsões realizadas por cada algoritmo de acordo com as configurações gerais e o uso ou não das evidências de conhecimento prévio (*quiz_nivelamento* e *quiz_algebra_precalculo*), sendo possível perceber pouca alteração nas acurácias com o uso dessas evidências.

Tabela 2. Acurácia dos modelos testados com três classes.

Configurações das Evidências	Conhecimento Prévio	RF	L	SMO
1. *nota_lista* e *nota_quiz*	Não	66,93%	58,06%	55,60%
	Sim	67,64%	56,42%	55,95%
2. *ColabWeb*	Não	58,32%	48,02%	47,27%
	Sim	60,54%	48,88%	49,02%
3. *CodeMeistre*	Não	74,76%	60,29%	58,39%
	Sim	76,01%	60,42%	58,62%
4. *CodeMeistre* e *ColabWeb*	Não	77,97%	60,85%	58,60%
	Sim	**78,64%**	60,80%	59,36%

Tabela 3. Acurácia dos modelos testados com duas classes.

Configurações das Evidências	Conhecimento Prévio	RF	L	SMO
1. *nota_lista* e *nota_quiz*	Não	70,97%	70,40%	70,40%
	Sim	71,08%	70,44%	70,40%
2. *ColabWeb*	Não	64,41%	63,24%	63,45%
	Sim	63,41%	62,74%	63,41%
3. *CodeMeistre*	Não	75,02%	70,30%	70,08%
	Sim	76,08%	69,69%	70,01%
4. *CodeMeistre* e *ColabWeb*	Não	**76,44%**	70,62%	70,05%
	Sim	72,34%	71,32%	70,05%

Dentre os modelos apresentados nas Tabelas 2 e 3, a Configuração 4 junto ao uso de evidências de conhecimento prévio apresentou a melhor acurácia com o classificador *Random Forest*. Por isso, foi eleita para ser executada por módulo.

O estudo de caso foi realizado a partir da base de dados original, ou seja, sem os dados acrescidos pelo SMOTE na Configuração 4. Com a separação dos dados por módulo e cada módulo possuindo diferentes quantidades de dados por classe, o balanceamento foi realizado de forma específica para cada conjunto de dados originado pelos sete módulos da disciplina de IPC. A utilização da mesma base de dados das configurações gerais faria com que houvesse a criação de uma grande quantidade de novos dados se distanciando da realidade.

As Tabelas 4 e 5 apresentam as acurácias das previsões realizadas por módulo através do algoritmo *Random Forest*, que ainda se mantém como o algoritmo mais apropriado para a

implementação e disponibilização de uma ferramenta de previsão de avaliações parciais. Os módulos 2, 4 e 6 estão destacados por apresentarem as melhores acurácias.

Tabela 4. Acurácia por módulo através da melhor configuração com 3 classes.

Módulos	RF	L	SMO
1. Variáveis e programação sequencial	79,83%	64,50%	63,50%
2. Condicional simples (*if-then*) e composta (*else*)	**88,14%**	73,39%	71,81%
3. Condicional encadeada (*elif*) e aninhada	78,33%	65,02%	66,50%
4. Repetição por condição	**83,19%**	61,55%	63,01%
5. Vetores e *Strings*	76,54%	64,02%	63,14%
6. Repetição por contagem	**84,27%**	65,62%	60,34%
7. Matrizes	66,43%	59,81%	57,21%

Tabela 5. Acurácia por módulo através da melhor configuração com 2 classes.

Módulos	RF	L	SMO
1. Variáveis e programação sequencial	77,00%	70,75%	69,50%
2. Condicional simples (*if-then*) e composta (*else*)	**86.17%**	74,90%	73,52%
3. Condicional encadeada (*elif*) e aninhada	77,34%	72,41%	73,65%
4. Repetição por condição	**79,39%**	69,08%	69,74%
5. Vetores e *Strings*	76,72%	72,49%	71,43%
6. Repetição por contagem	**80,51%**	74,36%	74,62%
7. Matrizes	72,36%	71,32%	70,05%

5.5 Discussão

Os dados da Tabela 1 são importantes para perceber que apesar dos *quizzes* serem menos utilizados pelos alunos do que os laboratórios de codificação, o número de atividades não realizadas se manteve constante enquanto as ausências nas atividades de codificação tornaram-se crescentes. Indicando que os alunos podem adquirir uma certa resistência a realização de exercícios de codificação.

A falta da realização das atividades de *quizzes* e de codificação, podem interferir na qualidade dos modelos gerados, uma vez que os conceitos “insatisfatório” e “sem conceito” são melhores classificados que o conceito “satisfatório”. Quanto mais evidências apresentarem comportamentos padronizados em relação a uma determinada classe, melhor será o desempenho e a acurácia dos algoritmos de previsão. Ou seja, a não realização das

atividades implica em evidências com valores "-1" (último parágrafo da Seção 4.1) e geralmente os alunos que não realizam as atividades, possuem conceito "insatisfatório" ou "sem conceito".

As Tabelas 2 e 3 demonstram que independentemente das classes serem politômicas ou dicotômicas, a quantidade de evidências utilizadas está relacionada com uma melhor previsão e as Tabelas 4 e 5 indicam que a diminuição dos registros, por conta da divisão por módulos, não foi um impedimento para o aumento da acurácia na maior parte das previsões.

As Tabela 2 e 3, também demonstram que as evidências do *ColabWeb* podem ser facilmente desconsideradas quando comparadas às evidências do *CodeMeistre*. Provavelmente pela semelhança entre os exercícios de codificação e a prova. Para a disciplina de IPC, esta comprovação é importante, pois atualmente nenhuma atividade do tipo *quiz* é disponibilizada através do *ColabWeb*, sendo o *CodeMeistre* a única ferramenta de apoio adotada.

Vale ressaltar que as evidências utilizadas no estudo de caso, como de costume na EDM (ver terceiro item do primeiro parágrafo da Seção 2.1), possuem algumas dependências entre si, por exemplo:

- Ausência das notas causa a ausência de várias outras evidências;
- A velocidade de digitação é influenciada pela quantidade de deletes e pelo número de linhas de *log* geradas (muitos trechos copiados e colados, geram uma menor quantidade de *log*);
- Média de deletes com a média de realização de execuções de um código;
- Média de dificuldade inferida de acordo com a quantidade de erros da sintaxe;
- Erros de sintaxe em relação à média de sucesso nas submissões.

O ideal seria buscar apenas evidências independentes entre si e com forte dependência ao que se deseja prever, mas esse é um dos grandes desafios da EDM. Tendo em vista as evidências selecionadas para esta dissertação seria importante em estudos futuros verificar o nível de relação de proximidade entre o que é abordado nos exercícios práticos com as questões das avaliações, apontando a realização ou não de tais atividades. Talvez essa equivalência possa justificar as acurácias das Tabelas 4 e 5, obtidas nas previsões por módulo.

As evidências de conhecimento prévio também carecem de atenção, pois o uso delas não apresentou uma melhoria considerável nos modelos gerados. É sabido que na prática elas

são importantes para o sucesso do aluno e algumas pesquisas sobre previsão também constataram essa informação [Dekker & Vleeshouwers 2009; Huang 2011; Márquez 2013; Brito et al. 2014 e 2015].

Outra evidência interessante que precisa ser melhor utilizada e estudada, é a evidência "*nota_lista_real*". Ela demonstra o real esforço do aluno ao responder os exercícios de codificação, em detrimento da evidência "*nota_lista*" que é redundante e pode conter resultados inconsistentes (ver Seção 4.1). Em um teste para viabilizar essa proposta, foi utilizada a configuração 4 com o *RandonForest*, desconsiderando a evidência "*nota_lista*" e a acurácia obtida foi de 78,76%, não muito diferente da acurácia já apresentada, de 78,64%.

A princípio, os experimentos realizados foram importantes para demonstrar que a seleção de evidências causa impacto na exatidão de um preditor gerado independente dos algoritmos utilizados. Mas é importante ressaltar que eles foram escolhidos basicamente por serem os mais populares na EDM e em testes empíricos realizados pela autora desta dissertação, ficaram dentro dos algoritmos de maior acurácia. Com exceção da regressão logística, os demais são considerados como algoritmos "caixa-preta", ou seja, são de difícil entendimento e consequentemente de difícil interpretabilidade. Os algoritmos de "caixa-branca" podem ser utilizados diretamente para a tomada de decisão, fornecendo explicação para a classificação que pode ser revisada e acordada por um especialista [Márquez et al 2013].

6 Considerações Finais

A EDM existe há pelo menos doze anos e possui muitos desafios. Como já foi visto, ela foi dissociada da Mineração de Dados graças às características específicas dos dados, derivados da área da Educação. No âmbito nacional, as pesquisas apresentadas ainda estão tratando da utilização desses dados para a tarefa de previsão, como é o caso desta dissertação.

Um dos grandes problemas da EDM é a existência de modelos específicos a um contexto educacional, o que dificulta a replicação de outros estudos [Romero & Ventura 2013]. Talvez por isso, alguns dos trabalhos internacionais relacionados estão em busca de simplesmente otimizar seus próprios modelos.

Por isso, os métodos descritos neste trabalho são as primeiras etapas de mineração de dados para uma posterior otimização e implementação. A princípio, o foco está nas avaliações parciais e evidências que possam estar relacionadas a elas. O melhor modelo geral desenvolvido, atingindo o objetivo desta dissertação, obteve acurácia de 78,64% com o algoritmo *Random Forest* e os modelos específicos, por módulo da disciplina entre 66,43% e 88,14%. Esses resultados demonstram que a Mineração de Dados com as evidências coletadas está na direção correta e pode evoluir, conforme os trabalhos futuros.

Espera-se que esta dissertação motive outros alunos da UFAM a estudarem e pesquisarem sobre o assunto com a finalidade de realizarem mais contribuições para a previsão apresentada, para a EDM e consequentemente para o ensino da disciplina de IPC.

A Mineração de Dados Educacionais está disponível para fornecer uma nova visão sobre o ensino e a aprendizagem. Praticamente todo tipo de dado é guardado até mesmo sem saber se terá utilidade para alguma constatação futura ou não. Novas técnicas, ferramentas e abordagens são criadas, com o intuito de melhorar a educação e inseri-la no contexto computacional. Os pesquisadores da computação anseiam por mais informações que possam desvendar os mistérios da educação.

Acima de muitos desafios encontrados pela EDM, ela necessita de pesquisadores da área da Educação. Atualmente, eles se encontram em menor número e por isso Elatia et al. (2016) acredita que durante a emersão da EDM, os avanços na pesquisa pedagógica e educacional permaneceram tangenciais. Desempenhando até agora, um papel periférico nessa área tão promissora que poderia beneficiar e moldas a educação. Possibilitando a formação de novas ideias e pesquisas sobre o ensino superior no século XXI.

6.1 Trabalhos Futuros

Apesar dos resultados serem satisfatórios, acredita-se que é possível aumentar a previsibilidade dos modelos através de mais análise de evidências e dos algoritmos utilizados para balanceamento dos dados e classificação. Também é imprescindível encontrar formas de motivar o aluno a resolver as atividades de forma contínua, para a geração de dados mais precisos sem *outliers*. Além da utilização de métodos estatístico para aumentar a confiabilidade dos modelos com evidências dependentes entre si.

Trabalhos futuros devem analisar o impacto do balanceamento através do algoritmo SMOTE, testar técnicas de classificação combinadas com ou sem algoritmos genéticos [Minaei & Punch 2003; Márquez 2013], implementar uma ferramenta capaz de suportar várias combinações de evidências a partir do modelo desenvolvido e disponibilizá-la aos docentes.

Referências

[Amaral 2016] AMARAL, Fernando. **Aprenda Mineração de Dados: Teoria e prática**. Alta Books Editora, 2016.

[Baker & Yacef 2019] BAKER, Ryan SJD; YACEF, Kalina. **The state of educational data mining in 2009: A review and future visions**. JEDM-Journal of Educational Data Mining, v. 1, n. 1, p. 3-17, 2009.

[Baker 2010] BAKER, Ryan. **Data mining for education**. International encyclopedia of education, v. 7, n. 3, p. 112-118, 2010.

[Baker & Inventado 2014] BAKER, Ryan Shaun; INVENTADO, Paul Salvador. **Educational data mining and learning analytics**. In: Learning analytics, p. 61-75. Springer New York, 2014.

[Baker et al. 2011] BAKER, Ryan; ISOTANI, Seiji; CARVALHO, Adriana. **Mineração de dados educacionais: Oportunidades para o Brasil**. Revista Brasileira de Informática na Educação, v. 19, n. 02, p. 03, 2011.

[Barella 2016] BARELLA, Victor Hugo. **Técnicas para o problema de dados desbalanceados em classificação hierárquica**. Dissertação de Mestrado. Universidade de São Paulo, 2016.

[Bayazit et al. 2014] BAYAZIT, Alper; ASKAR, Petek; COSGUN, Erdal. **Predicting learner answers correctness through eye movements with random forest**. In: Educational Data Mining. Springer International Publishing, 2014. p. 203-226.

[Bienkowski et al. 2012] BIENKOWSKI, Marie; FENG, Mingyu; MEANS, Barbara. **Enhancing teaching and learning through educational data mining and learning analytics: An issue brief**. US Department of Education, Office of Educational Technology, p. 1-57, 2012.

[Breiman 2001] BREIMAN, Leo. **Random forests**. Machine Learning, Boston, v.45, n.1, p.5-32, 2001.

[Brito et al. 2014] BRITO, Daniel Miranda et al. **Previsão de desempenho de alunos do primeiro período baseado nas notas de ingresso utilizando métodos de aprendizagem de máquina**. In: Simpósio Brasileiro de Informática na Educação-SBIE. 2014. p. 882.

[Brito et al. 2015] BRITO, Daniel Miranda et al. **Identificação de estudantes do primeiro semestre com risco de evasão através de técnicas de Data Mining**. In: Nuevas Ideas em Informática Educativa, TISE. 2015.

[Carvalho et al. 2016] CARVALHO, Leandro; FERNANDES, David; GADELHA, Bruno. **Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação**. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2016. p. 140.

[Castro & Braga 2011] CASTRO, Cristiano Leite de; BRAGA, Antônio Pádua. **Aprendizado supervisionado com conjuntos de dados desbalanceados**. Rev. Controle Autom, v. 22, n. 5, p. 441-466, 2011.

[Castro & Ferrari 2016] CASTRO, Leandro Nunes; FERRARI, Daniel Gomes. **Introdução à Mineração de Dados: Conceitos básicos, algoritmos e aplicações**. Saraiva, 2016.

[Castro & Fuks 2009] CASTRO, Thais; FUKS, Hugo. Inspeção semiótica do *ColabWeb*: proposta de adaptações para o contexto da aprendizagem de programação. **Brazilian Journal of Computers in Education**, v. 17, n. 01, p. 71, 2009.

[Cessie & Van Houwelingen 1992] LE CESSIE, Saskia; VAN HOUWELINGEN, Johannes C. **Ridge estimators in logistic regression**. Applied statistics, p. 191-201, 1992.

[Chaves et al. 2013] CHAVES, José Osvaldo M.; CASTRO, Angélica F.; LIMA, Rommel W.; LIMA, Marcos Vinícius A.; FERREIRA, Karl H. A. **Integrando Moodle e Juízes Online no Apoio a Atividades de Programação**. In Anais do Simpósio Brasileiro de Informática na Educação, Vol. 24, No. 1, p. 244, 2013.

[Chawla et al. 2002] CHAWLA, Nitesh V.; BOWYER, Kevin. W.; HALL, Lawrence O.; KEGELMEYER, W. Philip. **SMOTE: synthetic minority over-sampling technique**. Journal of artificial intelligence research, v. 16, p. 321-357, 2002.

[Cortes & Vapnik 1995] CORTES, Corinna; VAPNIK, Vladimir. **Support-vector networks**. Machine learning, v. 20, n. 3, p. 273-297, 1995.

[Costa et al. 2012] COSTA, Evandro et al. **Mineração de Dados Educacionais: Conceitos, Técnicas, Ferramentas e Aplicações**. Jornada de Atualização em Informática na Educação, v. 1, n. 1, p. 1-29, 2012.

[Dekker & Vleeshouwers 2009] DEKKER, G. W., Pechenizkiy, M., and VLEESHOUWERS, J. M. **Predicting Students Drop Out: A Case Study**. In: Proceedings of the International Conference on Educational Data Mining, 2009. p 41–50.

[Elatia et al. 2016] ELATIA, Samira et al. (Ed.). **Data Mining and Learning Analytics: Applications in Educational Research**. John Wiley & Sons, 2016.

[EpicGraphic] Disponível em <http://www.epicgraphic.com/data-cake/>

[Ferreira 2016] FERREIRA, João Luiz Cavalcante. **Md-pread: um modelo para previsão de reprovação de aprendizes na educação a distância usando árvore de decisão**. Dissertação, 2016.

[Guércio et al. 2014] GUÉRCIO, Hugo et al. **Análise do Desempenho Estudantil na Educação a Distância Aplicando Técnicas de Mineração de Dados**. In: Anais dos *Workshops* do Congresso Brasileiro de Informática na Educação. 2014. p. 641.

[Gutiérrez Posada et al. 2016] GUTIÉRREZ POSADA, Julián E.; BUCHDID, Samuel B.; BARANAUSKAS, M. Cecília C. **A informática na educação: o que revelam os trabalhos publicados no Brasil**. Revista Brasileira de Informática na Educação, v. 24, n. 1, 2016.

[Hämäläinen & Vinni 2006] HÄMÄLÄINEN, Wilhelmiina; VINNI, Mikko. **Comparison of machine learning methods for intelligent tutoring systems**. In: Intelligent tutoring systems. Springer Berlin/Heidelberg, 2006. p. 525-534.

[Hämäläinen & Vinni 2011] HÄMÄLÄINEN, Wilhelmiina; VINNI, Mikko. **Classifiers for educational data mining**. Handbook of Educational Data Mining, p. 57-74, 2011.

[Hämäläinen et al. 2004] HAMALAINEN, W., LAINE, T.H., SUTINEN, E., **Data mining in personalizing distance education courses**. In world conference on open learning and distance education, Hong Kong, 1–11, 2004 and in Data Mining in E-learning, v. 4, p. 157, 2006.

[Han et al. 2011] HAN, Jiawei; KAMBER, Micheline; PEI, Jian. **Data mining: concepts and techniques**. 3ª ed. Elsevier, 2011.

[Hosmer et al. 2013] HOSMER JR, David W.; LEMESHOW, Stanley; STURDIVANT, Rodney X. **Applied logistic regression**. John Wiley & Sons, 2013.

[Huang 2011] HUANG, Shaobo. **Predictive modeling and analysis of student academic performance in an engineering dynamics course**. Ph.D. Thesis dissertation, Utah State University, Logan, Utah, USA 2011.

[Junker 2011] JUNKER, Brian W. **Modeling hierarchy and dependence among task responses in educational data mining**. Handbook of Educational Data Mining, p. 143-155, 2011.

[Kotsiantis et al. 2003] KOTSIANTIS, Sotiris B.; PIERRAKEAS, C. J.; PINTELAS, Panayiotis E. **Preventing student dropout in distance learning using machine learning techniques. In: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems**. Springer, Berlin, Heidelberg, 2003. p. 267-274.

[Landwehr et al. 2005] LANDWEHR, Niels; HALL, Mark; FRANK, Eibe. **Logistic model trees**. Machine learning, v. 59, n. 1-2, p. 161-205, 2005.

[Magalhães et al. 2013] MAGALHÃES, Cleyton VC et al. **Caracterizando a pesquisa em informática na educação no Brasil: um mapeamento sistemático das publicações do SBIE**. Anais do 24º Simpósio Brasileiro de Informática na Educação (SBIE 2013). Campinas, 2013.

[Manhães 2015] MANHÃES, Laci Mary Barbosa. **Previsão do Desempenho Acadêmico de Graduandos Utilizando Mineração De Dados Educacionais**. 2015. Tese de Doutorado. Universidade Federal do Rio de Janeiro.

[Márquez et al. 2013] MÁRQUEZ, Carlos; CANO, Alberto; ROMERO, Cristóbal; VENTURA, Sebastian. **Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data**. Applied intelligence, v. 38, n. 3, p. 315-330, 2013.

[Martins et al. 2012] MARTINS, Luis Carlos; LOPES, Diogo Altoé; RAABE, André. **Um Assistente de Previsão de Evasão aplicado a uma disciplina Introdutória do curso de**

Ciência da Computação. In: Simpósio Brasileiro de Informática na Educação-SBIE. 2012.

[Minaei & Punch 2003] MINAEI-BIDGOLI, Behrouz; PUNCH, William F. **Using genetic algorithms for data mining optimization in an educational web-based system**. In: Genetic and evolutionary computation conference. Springer, Berlin, Heidelberg, 2003. p. 2252-2263.

[Moradi et al. 2014] MORADI, H.; MORADI, S. Abbas; KASHANI, L. **Students' Performance Prediction Using Multi-Channel Decision Fusion**. In: Educational Data Mining. Springer International Publishing, 2014. p. 151-174.

[Myller et al. 2002] MYLLER, Niko; SUHONEN, Jarkko; SUTINEN, Erkki. **Using data mining for improving web-based course design**. In: Computers in Education, 2002. Proceedings. International Conference on. IEEE, 2002. p. 959-963.

[Paes et al. 2013] PAES, R.B.; Malaquias, R.; GUIMARÃES, M.; ALMEIDA, H. **Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores**. In Anais dos *Workshops* do Congresso Brasileiro de Informática na Educação, Vol. 2, No. 1, 2013.

[Pelz et al. 2012] PELZ, F. D.; JESUS, E. A.; RAABE, A. L. **Um Mecanismo para Correção Automática de Exercícios Práticos de Programação Introdutória**. In Anais do XXIII Simpósio Brasileiro de Informática na Educação, Vol. 23, No. 1, 2012.

[Peña-Ayala 2013] PEÑA-AYALA, Alejandro (Ed.). **Educational Data Mining: Applications and Trends**. Springer, 2013.

[Peña-Ayala 2014] PEÑA-AYALA, Alejandro. **Educational data mining: A survey and a data mining-based analysis of recent works**. Expert systems with applications, v. 41, n. 4, p.

[Peña-Ayala 2017] PEÑA-AYALA, Alejandro. **Learning Analytics: Fundaments, Applications, and Trends**. Springer, 2017.

[Píccolo et al. 2010] PÍCCOLO, H. L.; SENA, V. F.; NOGUEIRA, K. B.; SILVA, M. O.; MAIA; Y. A. N. **Ambiente Interativo e Adaptável para ensino de Programação**. In: Simpósio Brasileiro de Informática na Educação-SBIE, 2010.

[Platt 1998] PLATT, John. **Sequential minimal optimization: A fast algorithm for training support vector machines**, 1998.

[Quilici-Gonzalez 2015] QUILICI-GONZALEZ, José Artur; DE ASSIS ZAMPIROLLI, Francisco. **Sistemas Inteligentes e Mineração de Dados**. Triunfal Gráfica e Editora, 2015.

[Ratner 2011] RATNER, Bruce. **Statistical and machine-learning data mining: Techniques for better predictive modeling and analysis of big data**. CRC Press, 2011.

[Romero & Ventura 2006] ROMERO, Cristobal; VENTURA, Sebastian (Ed.). **Data mining in e-learning**. Wit Press, 2006.

[Romero & Ventura 2013] ROMERO, Cristobal; VENTURA, Sebastian. **Data mining in education**. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, v. 03, n. 1, p. 12-27, 2013.

[Romero et al. 2013] ROMERO, Cristobal; ESPEJO, Pedro G.; ZAFRA, Amelia; VENTURA, Sebastian. **Web usage mining for predicting final marks of students that use Moodle courses**. Computer Applications in Engineering Education, v. 21, n. 1, p. 135-146, 2013.

[Romero et al. 2008] ROMERO, Cristóbal et al. **Data mining algorithms to classify students**. In: Educational Data Mining 2008.

[Romero & Ventura 2010] ROMERO, Cristóbal; VENTURA, Sebastián. **Educational data mining: a review of the state of the art**. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v. 40, n. 6, p. 601-618, 2010.

[Romero et al. 2011] ROMERO, Cristobal et al. **Handbook of educational data mining**. CRC Press, 2011.

[Santos et al. 2012] SANTOS, Henrique; CAMARGO, Fabiane; CAMARGO, Sandro. **Minerando Dados de Ambientes Virtuais de Aprendizagem para Previsão de Desempenho de Estudantes**. Conferencias LACLO, v. 3, n. 1, 2012.

[Serres 2013] SERRES, Michel. **Polegarzinha**. Rio de Janeiro: Bertrand Brasil, 2013.

[Souto & Duduchi 2009] SOUTO, Aletéia Vanessa Moreira; DUDUCHI, Marcelo. **Um processo de avaliação baseado em ferramenta computadorizada para o apoio ao ensino de programação de computadores**. In: *Workshop* de Educação em Computação. 2009.

[Souza et al. 2015] SOUZA, Adilson Martins; VENDRAMINI, Claudette Maria Medeiros; DA SILVA, Marjorie Cristina Rocha. **Validade preditiva de um processo seletivo em relação ao desempenho de universitários de Psicologia**. In: Revista de Psicologia, v. 16, n. 24, p. 55-68, 2015.

[Tan et al. 2009] TAN, P. N.; STEINBACH, M.; KUMAR, V. **Introduction to data mining**. Pearson, 2009.

[Tinto 1987] TINTO, Vincent. **Leaving college: Rethinking the causes and cures of student attrition**. University of Chicago Press, 5801 S. Ellis Avenue, Chicago, IL 60637, 1987.

[Truett et al. 1967] TRUETT, Jeanne; CORNFIELD, Jerome; KANNEL, William. **A multivariate analysis of the risk of coronary heart disease in Framingham**. Journal of chronic diseases, v. 20, n. 7, p. 511-524, 1967.

[csUnplugged] Disponível em <http://www.csunplugged.org />

[Vicini & Souza 2005] VICINI, Lorena; SOUZA, Adriano Mendonça. **Análise multivariada da teoria à prática**. Santa Maria: UFSM, CCNE, 2005.

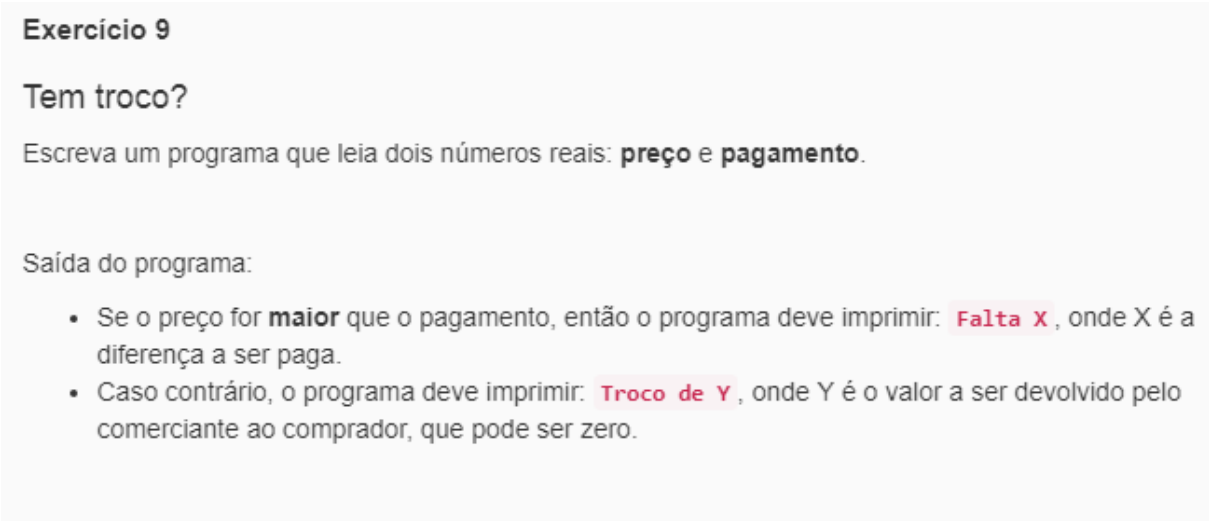
[WITTEN 2016] WITTEN, Ian H. et al. **Data Mining: Practical machine learning tools and techniques**. Morgan Kaufmann, 2016.

[Zafra et al. 2011] ZAFRA, Amelia; ROMERO, Cristóbal; VENTURA, Sebastián. **Multiple instance learning for classifying students in learning management systems**. Expert Systems with Applications, v. 38, n. 12, p. 15020-15031, 2011.

APÊNDICE A – Primeiro Apêndice

Exemplo de como o *CodeMeistre* realiza a correção das notas

Abaixo, uma ilustração de uma questão do módulo 2 com o código desenvolvido pelo aluno:



Exercício 9

Tem troco?

Escreva um programa que leia dois números reais: **preço** e **pagamento**.

Saída do programa:

- Se o preço for **maior** que o pagamento, então o programa deve imprimir: `Falta X`, onde X é a diferença a ser paga.
- Caso contrário, o programa deve imprimir: `Troco de Y`, onde Y é o valor a ser devolvido pelo comerciante ao comprador, que pode ser zero.

```
1
2   preco = float(input("Digite o preco:"))
3   pagamento = float(input("Digite o pagamento:"))
4
5   if(preco > pagamento):
6     x = float(preco - pagamento)
7     print("Falta ", round(x,2))
8   else:
9     y = float(pagamento - preco)
10    print("Troco de ", round(y,2))
11
```

O *CodeMeistre* pode testar o código com até três casos de teste, de acordo com o número de condições apresentadas pela questão. No caso desse exemplo, há apenas duas condições: falta de dinheiro para a realização do pagamento conforme o preço de um produto ou a presença de troco. Seguem os casos, apresentados pelo *CodeMeistre*:

- Primeiro Caso
  - Entrada: 12.3
    45.6
  - Saída correta: Troco de 33.3
- Segundo Caso
  - Entrada: 98.76
    54.32
  - Saída correta: Falta 44.44

Para a resolução de uma questão, o aluno recebe algumas dicas:

1. Atenção para a ordem de leitura de valores.
2. Observe se seu programa exibe a mensagem de erro exatamente como consta no enunciado.
3. Os valores em moeda devem ser arredondados em **duas casas decimais**

E ainda um exemplo de caso de teste:

- Caso de Teste
  - Entrada: 10.0
    50.0
  - Saída correta: Troco de 40.0

Note que o aluno detém todas as informações necessárias para resolver a questão de acordo com as saídas que serão avaliadas pelo *CodeMeistre*. Mas em alguns casos, os exercícios não são tão bem detalhados a ponto de dizer que o aluno precisa arredondar um número “em duas casas decimais”, o que significa que ele precisa de uma variável do tipo *float*. Em algum momento, é importante o aluno ter a maturidade de perceber que a questão envolve uma variável relacionada a dinheiro, logo será do tipo *float*. Assim como no caso de teste que apresenta o valor “10.0” em vez de “10”, que não seria errado. Apenas optou-se por dar mais uma dica ao aluno, para não ter o risco de ele utilizar uma variável do tipo *int*.

Caso o aluno não perceba e utilize uma variável *int* para a variável preço e pagamento, ele terá 0 como pontuação da questão. Caso ocorra em apenas uma variável, ele conseguirá metade da pontuação da questão. Claro que o aluno pode realizar diversos testes e obter a resposta da questão de forma instantânea! De toda forma, percebe-se que as notas geradas pelos exercícios do *CodeMeistre* podem ser: 0, 3.3, 5, 6.6 e 10.