Capítulo

12

Mineração de dados educacionais: avaliação e interpretação de modelos de classificação

Cristian Cechinel (UFSC), Sandro da Silva Camargo (UNIPAMPA)

Objetivo do Capítulo

Este capítulo tem como objetivo apresentar uma visão geral sobre os principais aspectos que devem ser considerados na avaliação e interpretação de modelos de classificação. Ao final da leitura deste capítulo, você deverá ser capaz de:

- Realizar pequenos experimentos de geração de modelos de classificação utilizando técnicas de mineração de dados.
- Entender as principais medidas (métricas) existentes que devem ser consideradas ao avaliar a qualidade e desempenho de modelos de classificação.
- Ler e avaliar de maneira correta os resultados obtidos no processo de mineração de dados educacionais.
- Entender os principais equívocos comumente recorrentes na avaliação e interpretação de resultados obtidos na tarefa de classificação.

Era uma vez... *um professor que possuía vários conjuntos de dados sobre as disciplinas que tinha ministrado ao longo dos anos, assim como também sobre informações curriculares dos estudantes do curso de graduação em que foi coordenador. O professor possuía noções gerais de mineração de dados e utilizou esses conjuntos de dados para gerar alguns modelos de predição de comportamentos futuros de seus estudantes, como o risco de evasão e reprovação, além de predição das possíveis disciplinas em que os mesmos iriam se matricular nos próximos semestres. Entretanto, o professor não estava seguro sobre a validade dos modelos gerados e tinha dúvidas sobre como deveria realizar a avaliação dos mesmos. O professor também gostaria de verificar se a metodologia que ele utilizou para a geração dos modelos era adequada, ou se ele havia cometido algum equívoco ao longo do processo de mineração.*

1 As diferentes abordagens da mineração de dados

Embora tenhamos observado um crescimento significativo na quantidade de publicações brasileiras envolvendo experimentos de mineração de dados educacionais nos últimos anos, é ainda comum encontrarmos problemas metodológicos na montagem desses experimentos e também na apresentação, interpretação e discussão dos seus resultados. Em muitos dos casos, os erros cometidos ao longo do processo poderiam ser facilmente corrigidos com a adoção de algumas técnicas e/ou estratégias específicas nas diferentes etapas do ciclo de vida da mineração dos dados. Em outros casos, a própria interpretação e discussão dos resultados encontrados poderia ser melhorada a partir de um conhecimento mais panorâmico das diferentes métricas de avaliação existentes e de como as mesmas devem ser observadas em consonância com as características dos dados disponíveis. É relativamente comum encontrar trabalhos que desconsideram que o balanceamento dos dados influencia no desempenho dos modelos gerados, ou trabalhos que não aplicam princípios básicos de separação dos dados em conjuntos de treinamento, teste e validação. Há trabalhos em que os modelos são treinados e avaliados utilizando o mesmo conjunto de dados, comprometendo assim os resultados apresentados. Outro equívoco também recorrente é a avaliação de modelos de classificação utilizando exclusivamente a acurácia média geral, desconsiderando taxas baixas obtidas na classificação de uma das categorias em benefício de taxas altas obtidas na classificação da(s) outra(s) categoria(s). Ainda, a facilidade que atualmente encontramos em utilizar *frameworks* livres e robustos para realizar os experimentos de mineração de dados também pode oferecer algumas armadilhas para os novos pesquisadores da área, que eventualmente ignoram o significado de muitos dos parâmetros que são inicializados de maneira automática por essas ferramentas e acabam executando seus experimentos utilizando o fluxo padrão proposto pelas mesmas, sem necessariamente observar as características de seus dados. Este capítulo propõe discutir algumas das principais características dos dados educacionais que devem ser observadas no momento de realizar algum experimento envolvendo a sua mineração. Também serão apresentados alguns cenários possíveis de geração de modelos de classificação dentro do contexto educacional, além de algumas das principais métricas de avaliação que podem ser observadas no momento de avaliar e interpretar esses modelos. O objetivo principal é o de fornecer subsídios para os pesquisadores em informática na educação realizarem experimentos de mineração de dados educacionais sempre considerando as características gerais de seus dados, respeitando as regras necessárias para o correto treinamento, teste e validação de seus modelos, e interpretando os resultados obtidos em consonância com as características dos dados que foram utilizados nesses experimentos.

Ainda que o objetivo deste capítulo não seja o de explicar conceitos introdutórios de mineração de dados e de aprendizado de máquina, cabe aqui uma breve explicação de alguns dos principais conceitos existentes para melhor situar o leitor dentro do contexto específico da presente proposta. Na literatura relacionada a área, normalmente encontramos uma distinção entre duas abordagens principais para as tarefas de aprendizado de máquina, sendo elas o aprendizado supervisionado e o aprendizado não supervisionado.

O **aprendizado supervisionado** trabalha com conjuntos de dados em que os

exemplos para o treino dos modelos possuem classes rotuladas, de maneira que é possível identificar claramente uma variável alvo que se deseje classificar ou modelar. Em outras palavras, possuímos um conjunto de variáveis de entrada e um conjunto de variáveis de saída, e tentamos inferir uma função capaz de mapear esse conjunto de variáveis de entrada de maneira a predizer a variável de saída. Por exemplo, consideremos um conjunto de dados relacionados aos estudantes de uma determinada disciplina e que contém as informações das presenças dos mesmos ao longo das primeiras semanas, além das atividades que foram entregues durante este período da disciplina, e as situações finais após o término da disciplina (aprovado ou reprovado). Considerando que conhecemos a situação final dos estudantes, é possível utilizar algoritmos de mineração de dados para inferir funções capazes de predizer, com algum grau de precisão, aqueles que irão aprovar ou reprovar com base em suas presenças e na entrega das atividades exigidas neste período inicial da disciplina. Nesse sentido, teríamos como resultado o mapeamento de uma função capaz de predizer a variável de saída, *situação na disciplina* (*aprovado* ou *reprovado*), com base nas variáveis de entrada, *presenças ao longo das semanas* e *atividades entregues*. Um outro exemplo pode ser a predição de *evasão do aluno* (*evadido* ou *não evadido*) em um determinado curso com base em seu histórico escolar. Considerando que são conhecidas as disciplinas já cursadas pelo aluno, suas respectivas notas, frequência, e situação final como aprovado, reprovado por nota ou reprovado por frequência. Assim, inferidas funções que mapeiam estes atributos conhecidos na variável de saída: *evasão*. A abordagem de aprendizado supervisionado normalmente envolve dois tipos de problemas, sendo eles o de classificação e o de regressão. A diferença principal entre os mesmos está relacionada com o tipo da variável alvo (variável de saída) que se deseja predizer. Problemas de classificação possuem variáveis de saída categóricas, enquanto problemas de regressão possuem variáveis de saída numéricas. Dentre os algoritmos que são utilizados no aprendizado supervisionado podemos citar: redes neurais artificiais, árvores de decisão, redes bayesianas, máquinas de vetor de suporte, entre outros.

No aprendizado **não supervisionado** a base de dados não possui uma variável alvo ou de saída e o objetivo dos algoritmos de aprendizado consiste em descobrir as relações existentes entre as variáveis da base de dados. Isso é tipicamente feito através do agrupamento (*clustering*) dos exemplos de acordo com alguma métrica de semelhança. Estes algoritmos visam satisfazer dois objetivos principais: minimizar as diferenças intragrupos e maximizar as diferenças intergrupos. Esse tipo de aprendizado pode servir para descrever os dados existentes, permitindo uma melhor compreensão dos mesmos, ou também como forma de geração de variáveis categóricas, a partir de variáveis numéricas, que posteriormente podem ser utilizadas em tarefas de classificação. Por exemplo, consideremos uma base de dados semelhante a anterior, contendo dados de estudantes de uma determinada disciplina relacionados às presenças dos mesmos ao longo das semanas, além das atividades que foram entregues pelos estudantes durante um determinado período da disciplina. Essa nova base não contém, entretanto, as situações finais dos estudantes após o término da disciplina. O aprendizado não supervisionado pode tentar encontrar grupos de estudantes que possuam características similares entre si sem necessariamente estabelecer o significado de cada grupo encontrado. Ainda assim, um analista de dados experiente e que conheça de maneira aprofundada o contexto de onde os dados foram extraídos pode ser capaz de inferir o significado por trás de cada um desses grupos gerados. Para o exemplo em

questão, estudantes que frequentaram poucas aulas e que entregaram poucas atividades podem potencialmente ser alocados em um mesmo grupo, indicando assim que pertencem a um conjunto de estudantes em risco de reprovação, enquanto estudantes que frequentaram de maneira regular às aulas e entregaram a maior parte das atividades podem ser alocados em um outro grupo, indicando que pertencem a um conjunto de estudantes com menor risco de reprovação. Tarefas de clusterização (agrupamento), associação e de extração de características pertencem à abordagem de aprendizado de máquina não supervisionado.

Este capítulo tratará exclusivamente da abordagem de aprendizado supervisionado e dentro do contexto específico da tarefa de classificação. Alguns exemplos de situações em que a tarefa de classificação pode ser útil no contexto educacional são (ROMERO et al., 2008):

- Predizer a aprovação ou reprovação de um aluno em um componente curricular (SANTOS; CAMARGO; CAMARGO, 2012).
- Predizer o sucesso ou insucesso de um aluno em um curso (CAMARGO; BORIN; FERREIRA, 2014).
- Predizer a evasão de um aluno de um componente ou de um curso.
- Predizer as dificuldades dos alunos em certos componentes com base em seu desempenho nos pré-requisitos.
- Classificar acadêmicos propensos ao desânimo (SANTOS; BERCHT; WIVES, 2015)
- Detectar má conduta acadêmica dos estudantes dentro de ambientes virtuais de aprendizagem
- Detectar o estilo de aprendizagem de um acadêmico com base em seu comportamento no AVA.
- Classificar a qualidade de um determinado recurso de aprendizagem com base nos comentários realizados pelos estudantes (SANTOS; CECHINEL, 2015)
- Classificar postagens em fóruns educacionais como dúvida, resposta ou comentário neutro (ROLIM; MELLO; COSTA, 2017).
- Classificar um determinado problema/questão/exercício de acordo com seu nível de dificuldade.
- Avaliar de maneira automática a qualidade de um recurso educacional digital (CECHINEL et al., 2016)

Ao longo do capítulo abordaremos algumas das características dos dados que devem ser observadas durante o processo de geração de modelos de classificação, além dos cuidados no momento de realizar o treinamento e o teste dos modelos, e algumas medidas para avaliar a qualidade dos mesmos de maneira correta.

2 Observando as características dos dados

Dados são o elemento fundamental para o processo de mineração. Os dados são valores referentes a medições, contagens ou observações relativas a uma amostra ou a um determinado fenômeno. Como exemplo de dados, podemos citar a nota de um aluno em uma avaliação, a quantidade de acessos de um aluno a um material, e a presença ou ausência de um aluno em uma aula. No contexto educacional, os dados podem ser oriundas de diferentes fontes (eg. ambientes virtuais de aprendizagem, questionários, sites de professores, sistemas acadêmicos, sistemas tutores inteligentes, etc) e fornecer uma grande quantidade de informações sobre estudantes, professores e os contextos educacionais em que estão inseridos. De acordo com Romero, Romero e Ventura (2014), os dados educacionais possuem algumas características peculiares e resultantes do contexto específico de onde são extraídos. Por exemplo, é comum que estudantes não concluam todos os exercícios e atividades de uma determinada aula fazendo com que bases de dados sobre essas informações normalmente sejam incompletas e com campos faltantes. Outro situação é a existência de um grande número de atributos sobre os estudantes e de várias instâncias com diferentes níveis de granularidade (e.g. dados relacionados ao curso, a uma disciplina, a uma atividade específica), tornando quase sempre necessário a utilização de técnicas de seleção e filtragem dos atributos mais representativos para um determinado problema.

O conjunto de valores de uma característica particular é chamada de variável. Como exemplo de variável podem ser consideradas as notas de todos os alunos em uma determinada avaliação. As variáveis podem ser quantitativas, representadas por números, ou qualitativas, representadas por categorias ou classes (KAPS; LAMBERSON, 2004).

Variáveis quantitativas têm seus valores expressos em números, com domínio inteiro ou real, e as diferenças entre os valores têm um significado numérico. Como exemplo de uma variável quantitativa pode ser citado o horário em que um aluno ingressou no AVA. As variáveis quantitativas podem ser contínuas ou discretas. Enquanto as variáveis contínuas podem assumir uma quantidade infinita de valores em um determinado intervalo, as variáveis discretas podem assumir uma quantidade finita de valores. Um exemplo de uma variável contínua poderia ser o tempo decorrido entre a entrada e a saída do aluno no AVA, e de uma variável discreta, a quantidade de acessos a uma atividade.

Por outro lado, as **variáveis qualitativas** têm seus valores expressos em categorias. Exemplos de variáveis qualitativas podem ser a situação de um aluno (aprovado ou reprovado), ou seu sexo (masculino ou feminino). Uma variável qualitativa pode ser subclassificada em ordinal ou nominal. Enquanto uma variável ordinal pode ser ordenada, o mesmo não ocorre com as variáveis nominais, onde não há uma relação de ordem entre categorias. Exemplos de variáveis nominais podem ser um indicador de se o aluno ingressou na instituição de ensino através de ações afirmativas ou não. Já um exemplo de variável ordinal pode ser o conceito do aluno (A-Excelente, B-Satisfatório, C-Suficiente, D-Insuficiente). Para a tarefa de classificação, a variável a ser predita deve obrigatoriamente ser qualitativa. Quando a variável a ser predita é quantitativa, os modelos de predição são gerados por meio da tarefa de regressão, que está fora do contexto deste capítulo.

2.1 Sobre o balanceamento dos dados

Há uma grande complexidade inerente ao processo de construção de modelos de classificação sobre conjuntos de dados desbalanceados, ou seja, com ampla disparidade de quantidade de dados em cada classes. Em uma situação ideal, a quantidade de dados de cada classe que se deseja modelar deve ser similar, de forma que possam ser aprendidas as peculiaridades de cada uma das classes, fazendo com que o classificador possa atingir um nível de precisão similar elas.

Na ampla maioria dos algoritmos, o treinamento com dados desbalanceados faz com que os modelos criados também tenham precisão desbalanceada, de forma a atingir um acerto próximo a 100% na classificação da classe majoritária (com mais dados), e acerto próximo a 0% na classe minoritária (com menos dados). Como exemplo, se uma base de dados tem 95% dos dados pertencentes a uma classe A, e 5% pertencentes a uma classe B, o classificador poderá ter a tendência de classificar todos os registros como classe A, tendo uma taxa de acerto geral de 95%. No entanto, esse classificador provavelmente não teria aprendido nenhum dos padrões representativos da classe B.

As técnicas utilizadas para lidar com dados desbalanceados podem ser agrupadas em cinco categorias: métodos de amostragem, métodos sensíveis a custo, métodos de aprendizado baseados em *kernel*, métodos de aprendizado ativo e métodos de aprendizado de classe única (HE; MA, 2013).

Métodos de amostragem têm sido a abordagem mais utilizada. Eles adotam estratégias de buscar um balanceamento dos dados, antes do treinamento do algoritmo de classificação, de forma a reduzirem a quantidade de amostras da classe majoritária, por métodos aleatórios ou baseados em clusters, ou incrementarem a quantidade de amostras da classe minoritária, por reamostragem aleatória ou por geração de dados sintéticos. Ainda é possível a alternativa de combinação destas estratégias.

Os métodos clássicos de treinamento visam minimizar o erro global de classificação, assumindo custos iguais para erros ou acertos de classificação em ambas classes: minoritária e majoritária. Os métodos sensíveis a custo visam incorporar variações de custo no processo de treinamento dos algoritmos, de forma que acertos na classe minoritária sejam melhor avaliados do que acertos na classe majoritária. Outra alternativa é a penalização maior dos erros na classe minoritária em relação a erros na classe majoritária (CASTRO; BRAGA, 2011).

Os métodos de aprendizagem baseados em *kernel* tipicamente envolvem modificações no espaço de características dos dados de entrada visando o deslocamento da superfície de decisão entre as classes ou o aumento da distribuição espacial das amostras minoritárias.

Nos métodos de aprendizado ativo, o próprio classificador exerce um papel ativo na seleção das amostras para treinamento, selecionando as amostras mais informativas dentro do problema a ser tratado. As amostras mais informativas seriam aquelas mais próximas à superfície de decisão que separa as classes analisadas (ATTENBERT; ERTEKIN, 2013).

Enquanto os métodos convencionais de aprendizado buscam aprender as peculiaridades de múltiplas classes, os métodos de classe única, ou abordagem baseada em reconhecimento, visam aprender somente os padrões de uma das classes, visando

predizer se uma nova amostra pertence ou não à classe que ele aprendeu. Assim, pode ser construído um modelo que visa apenas aprender se a amostra pertence ou não à classe minoritária (CASTRO; BRAGA, 2011).

2.2 Sobre a transformação e a dimensionalidade dos dados

Conhecer os tipos de dados é fundamental para a escolha dos algoritmos. Alguns algoritmos tem restrição de só trabalharem com tipos de dados específicos. O uso de **técnicas de transformação de dados** permite superar as restrições de tipo dos dados de entrada para aumentar o conjunto de algoritmos possíveis de serem utilizados, e também pode auxiliar na melhoria do desempenho dos modelos de classificação em alguns casos. Por exemplo, variáveis quantitativas podem ser transformadas em variáveis qualitativas através da **discretização**, onde os intervalos de valores contínuos são mapeados em atributos ordinais categóricos. Dessa maneira, a nota de um acadêmico (variando entre 0 e 10) pode ser discretizada em 2 classes distintas referentes a aprovação (nota 7) e reprovação (nota < 7). Um outro tipo de transformação dos dados bastante utilizada é a **binarização**, onde cada categoria de uma variável qualitativa é transformada em uma nova variável binária, onde o valor 1 para a variável significa a ocorrência da categoria, e o valor 0 a não ocorrência da mesma.

Dados educacionais podem conter centenas de variáveis, sendo que muitas delas podem ser irrelevantes para o processo de classificação em questão (HAN; KAMBER, 2011). Apesar de ser possível o especialista do domínio selecionar as variáveis que ele julga mais informativas, esta tarefa geralmente demanda um grande consumo de tempo, principalmente no caso dos dados não serem plenamente conhecidos.

Neste contexto, a **redução de dimensionalidade dos dados (RDD)** é um processo que visa encontrar uma estrutura mais compacta de representação dos dados através do mapeamento de cada amostra para um vetor de menor dimensão de características. Porém, a RDD não deve resultar em perda de informação relevante em relação aos dados originais, ou pelo menos, os benefícios obtidos com a RDD devem ser maiores que o prejuízo da perda de Informação. Assim, o resultado prático da aplicação de técnicas de RDD é uma redução do espaço de busca de hipóteses, com a consequente melhora do desempenho do processo de criação dos classificadores e simplificação dos resultados do processo de mineração de dados (WANG; XIUJU, 2005).

A RDD é especialmente útil quando há uma grande quantidade de variáveis descrevendo cada exemplo no banco de dados, fato peculiar aos bancos de dados educacionais. Nestes casos, a quantidade de amostras necessária para ajustar um modelo multivariado pode crescer exponencialmente em relação à quantidade de variáveis. Porém, muitas vezes, a obtenção de mais amostras é inviável devido à grande dificuldade ou ao grande custo deste processo. Além disso, o uso de muitas variáveis no modelo preditivo pode dificultar a interpretação da análise e viola o princípio da parcimônia (princípio que recomenda a escolha da explicação mais simples para um determinado fenômeno). Outro fator importante é que muitas variáveis podem mais facilmente conduzir ao sobreajuste (do inglês *overfitting*), do modelo preditivo (LAROSE, 2006). O conceito de sobreajuste será discutido na próxima seção.

Embora os algoritmos de mineração de dados já executem internamente

abordagens de RDD, eles geralmente pecam no quesito escalabilidade (YE, 2003). Desta forma, a aplicação de técnicas específicas de RDD em combinação com os algoritmos de mineração geralmente conduz a melhores resultados. As técnicas de RDD podem ser divididas em três categorias: extração de características, construção de características e seleção de características. Apesar da divisão didática, tanto a extração de características quanto a construção de características geralmente são sucedidas pela seleção. Isto ocorre porque tanto a extração quanto a construção **derivam novos atributos** (novas características) tomando como ponto de partida os atributos existentes na base, suas relações, combinações e também transformações (CAMARGO, 2010). Como exemplos de derivação de novos atributos podemos mencionar (ROMERO; ROMERO; VENTURA, 2014) o percentual de testes corretamente respondidos (calculado pelo número total de testes corretos dividido pelo número total de testes realizados, ou o tempo total de leitura das páginas de uma sessão (calculado pela soma dos tempos gastos em cada página acessada). A derivação de novos atributos pode enriquecer o conjunto de dados existente e melhorar o desempenho dos modelos de classificação em alguns casos.

Considere um exemplo onde os dados de entrada incluem atividades realizadas ou não pelos alunos em um AVA visando predizer a aprovação ou não do aluno na disciplina. Supondo-se a existência de alunos aprovados e reprovados na disciplina, e dois extremos em relação às atividades propostas, de forma que uma atividade x foi realizada por todos os alunos, e uma outra atividade y que não foi realizada por nenhum aluno. Esta situação mostra que a realização ou não das atividades x e y não tem nenhuma influência preditiva na aprovação ou não do aluno. Desta forma, utilizar as atividades x e y como entradas vai aumentar a complexidade do processo de treinamento e tais elementos não serão considerados na construção dos modelos. Logo, a eliminação de tais elementos através de um processo de redução de dimensionalidade, no pré-processamento dos dados, tende a diminuir a complexidade das fases posteriores. Em uma outra situação, busca-se predizer o sucesso do aluno no curso a partir dos dados de entrada com seus históricos escolares. Planeja-se ter uma coluna para cada disciplina com a nota do aluno. No entanto, é uma situação normal que alguns alunos cursem a mesma disciplina mais de uma vez, devido à reprovações por nota ou por frequência. Supondo-se que um determinado aluno tenha cursado cinco vezes uma determinada disciplina, seria necessária a existência de cinco colunas no arquivo de entrada para esta disciplina, sendo que um aluno que tenha conseguido aprovação na primeira matrícula, terá quatro colunas com valores zero. Assim, uma alternativa seria ter no arquivo de entrada duas colunas para cada disciplina: uma contendo a nota de aprovação e outra contendo a quantidade de vezes que a disciplina foi cursada. Esta segunda coluna não existe diretamente no banco de dados, mas poderia ser construída facilmente. Embora esta abordagem de construção citada seja manual, também existem abordagens automáticas de construção de características.

A importância da derivação de atributos - um exemplo prático

No artigo "Modelagem e Predição de Reprovação de estudantes de Cursos de Educação a Distância a partir da Contagem de Interações" (CECHINEL; ARAÚJO; DETONI, 2015), os autores desenvolveram modelos de classificação para prever com antecedência o risco de reprovação de acadêmicos em cursos a distância utilizando dados dos logs das interações no Ambiente Virtual de Aprendizagem. A variável pivô de todo o trabalho é a contagem de interações semanais dos acadêmicos no AVA, porém os autores adotaram a estratégia de derivar alguns novos atributos a partir dessa variável, como por exemplo: 1) a média do total de interações do acadêmico pelo número de semanas, 2) a mediana do conjunto de interações por semana, 3) o número de semanas com zero interações, 4) a média da diferença de interações entre a semana i e a semana i+1, e 5) a razão entre as interações da semana do acadêmico e a média de interações da turma naquela semana. Os resultados dos experimentos apontaram para uma melhora significativa no desempenho dos modelos de classificação que utilizavam atributos derivados, sobretudo nas primeiras semanas das disciplinas. Uma estratégia similar para a derivação de atributos foi utilizada também por Queiroga, Cechinel e Araújo (2017) para a predição de estudantes com risco de evasão em cursos técnicos a distância.

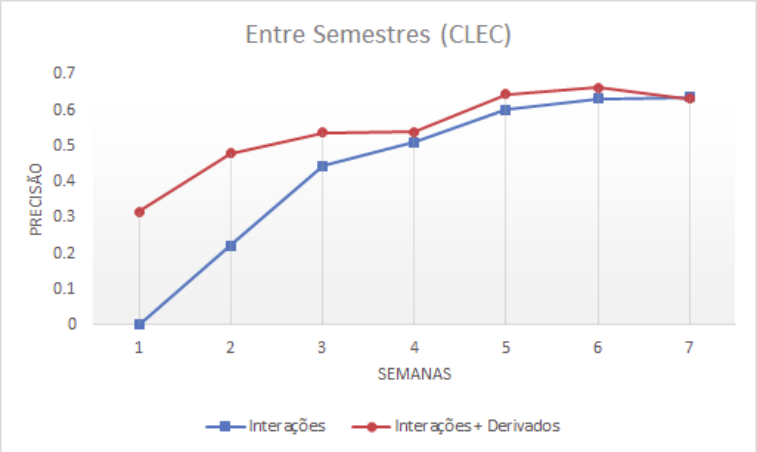

Figura 1. Classificação correta de acadêmicos em risco utilizando contagem de interações e com a utilização de atributos derivados

Fonte: (CECHINEL; ARAÚJO; DETONI, 2015)

3 Cuidados ao realizar a construção e a avaliação dos modelos de classificação

O processo de criação de modelos de classificação envolve duas fases distintas: a construção e a avaliação dos modelos. Os conjuntos de dados utilizados nestas duas fases devem ser disjuntos. Desta forma, a base de dados a ser utilizada deve ser dividida em pelo menos dois conjuntos: o conjunto de treinamento, utilizado na fase de construção, e o conjunto de teste, utilizado na fase de avaliação. Enquanto na fase de construção dos modelos, os algoritmos buscam inferir uma função que permita fazer o melhor mapeamento entre os dados de entrada e de saída, na fase de avaliação busca-se avaliar a capacidade preditiva do modelo sobre dados não vistos previamente pelos algoritmos.

A avaliação do modelo é uma atividade complexa que exige formas sistemáticas de trabalho, sendo que os algoritmos de mineração de dados frequentemente exigem a

configuração de um conjunto de parâmetros que exercem uma influência determinante nos resultados obtidos. Diferentes valores dos parâmetros geram diferentes modelos (CAMARGO, 2010). Alguns dos problemas enfrentados e falhas cometidas no processo de mineração de dados podem ocorrer no momento de realizar o treinamento e o teste dos modelos, assim como também na escolha das métricas mais adequadas para a avaliação dos resultados.

Com relação especificamente às etapas de **treinamento** e **avaliação** dos modelos, é necessária a aplicação de técnicas que permitam avaliar o desempenho preditivo do modelo gerado em dados que não foram previamente vistos (OLSON; DELEN, 2008). A ideia básica é a de garantir que os modelos sejam treinados com um conjunto de dados, e testados em um conjunto distinto desse primeiro. Essa separação garante que os modelos gerados sejam realmente capazes de predizer (classificar) a partir de dados desconhecidos aos modelos; ou seja, o conjunto de teste deve ser formado por instâncias independentes que não tomaram parte na construção do classificador. A qualidade de um modelo de predição está diretamente relacionada com a capacidade que o mesmo possui em generalizar para novos conjuntos de dados as características (padrões) aprendidas a partir do conjunto de dados de treinamento, desde que estes novos dados tenham origem idêntica a dos treinamento. Em outras palavras, um bom modelo de predição deve ser capaz de extrair a função geradora dos dados e não apenas memorizar os dados de treinamento, tornando-o apto a ser aplicado a novos exemplos. É possível que os padrões aprendidos por um determinado modelo em um conjunto de dados de treinamento não sejam necessariamente encontrados no conjunto de dados mais geral. Por esse motivo, a etapa de avaliação dos modelos sempre utiliza também desse conjunto de dados para teste, permitindo assim uma comparação entre a saída real dos modelos com a saída desejada para os mesmos.

Sobreajuste

Chamamos de sobreajuste quando um modelo de predição apresenta um bom desempenho na etapa de treinamento, porém uma baixa capacidade de generalização para outros conjuntos de dados. O sobreajuste ocorre quando o modelo de predição aprende não somente os padrões gerais dos dados, mas também todos os (ou muitos dos) casos específicos e ruídos encontrados no conjunto. Considera-se que o modelo não aprendeu os principais padrões ou tendências dos dados da base, mas simplesmente “decorou” todos os seus casos. Nestas situações, o modelo se encaixa perfeitamente para a base de dados de treinamento, porém não consegue generalizar para novas bases de dados. Na figura 2 o modelo representado pela linha preta é capaz de realizar melhores predições em dados novos do que o modelo da linha verde (*overfitted*).

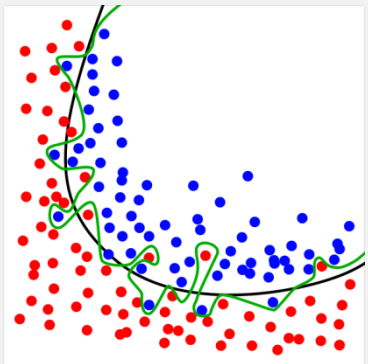

Figura 2. Sobreajuste.

Fonte: ICKE (2008)

Existem diferentes possibilidades para garantir essa distinção entre os conjuntos de dados de treinamento e teste, sendo que as mesmas dependem fundamentalmente das características dos mesmos. Existem situações em que os dados para treinamento e teste já estão naturalmente divididos em conjuntos distintos, não sendo necessário alguma técnica específica de particionamento. Por exemplo, consideremos que desejamos gerar um modelo para classificar alunos entre aprovados e reprovados em uma determinada disciplina utilizando como entrada as interações desses alunos dentro de um Ambiente Virtual de Aprendizagem (CECHINEL; ARAUJO; DETONI, 2015). Sabe-se que a disciplina já foi executada em 2 semestres (A e B) por um mesmo professor e com configurações no AVA e metodologias de ensino idênticas. Nesse tipo de situação, o classificador pode ser treinado com os dados do semestre A e testado com os dados do semestre B, e em seguida treinado com os dados do semestre B e testado com os dados do semestre A.

Quando os dados não estão naturalmente divididos em conjuntos distintos, a alternativa é recorrer a alguma estratégia de **particionamento dos dados**. A seguir são descritas as duas principais formas de particionamento existentes (BISHOP, 1995), sendo elas: o ***holdout*** e a **validação cruzada**. O método de particionamento ***holdout*** é adotado quando existe uma grande quantidade de dados disponível para o processo de mineração. Neste método os dados são divididos aleatoriamente em duas partições independentes e sem sobreposição: uma de treinamento e outra de teste. A partição de treinamento é usada para construir/treinar o modelo, e a partição de teste é utilizada para avaliar a capacidade de generalização do modelo. Não há uma regra universal para definir o tamanho de cada partição, mas é comum a utilização de uma partição de treinamento de aproximadamente 75% dos dados e uma de teste de 25%. Uma variação da técnica holdout é a subamostragem aleatória, onde os conjuntos de treinamento e teste são particionados de maneira aleatória, sendo o procedimento repetido k vezes. A exatidão do método é estimada pela média da exatidão obtida em todas as k repetições. A utilização da técnica de holdout é aconselhada somente para quando se possui uma grande quantidade de dados. Deve-se levar em conta também que a avaliação do modelo pode variar de maneira significativa dependendo dos conjuntos de treinamento e teste gerados. A **validação cruzada** (do inglês *cross-validation*) é utilizada quando o conjunto de dados que possuímos é limitado. Nesses casos, utilizamos todos os casos da base de dados tanto para teste quanto para treinamento, porém não ao mesmo tempo. A estratégia consiste em dividir o conjunto de dados em *n* partições (n-fold) de igual tamanho (ou tamanhos similares), sendo que a partição n é utilizada para teste e as demais partições são utilizadas para treinamento. A divisão dos dados em 10 partições (10-folds) tem se tornado um procedimento padrão visto que, testes em vários bancos de dados e com diferentes técnicas de mineração têm mostrado que 10 seria um número adequado para obtenção de uma boa estimativa de erro (WITTEN; FRANK; HALL, 2011). Quando a quantidade de dados é extremamente pequena, utiliza-se um caso específico de validação cruzada denominado ***leave-one-out*** e que consiste em utilizar partições de apenas 1 único elemento para teste, ou seja, o número de partições gerado é igual ao número de casos da base de dados. A exatidão do modelo é calculada medindo a exatidão na predição da amostra de teste, e a exatidão final do modelo é dada pela média da exatidão de todos os n experimentos. Esse procedimento apresenta grande utilidade para pequenos bancos de dados, porém é computacionalmente custoso.

Existem ainda outras técnicas de validação (ex. *bootstrap*), se o leitor estiver interessado em alguma leitura complementar recomendamos a seção 2.4.3 de Camargo (2010).

As opções de teste na ferramenta Weka

A ferramenta Weka (HALL et al., 2009) é largamente utilizada para realizar tarefas de mineração de dados por conta de sua interface bastante intuitiva. Na aba para a tarefa de Classificação (*Classify*) são apresentadas 4 opções de teste (*Test options*), sendo elas: 1) *Use training set*, 2) *Supplied test set*, 3) *Cross-Validation* e 4) *Percentage Split*. A opção *Use training set* é a primeira a ser apresentada para o usuário e é muitas vezes utilizada de maneira automática por usuários mais iniciantes que não compreendem o significado e funcionamento de cada opção. Esta opção entretanto, não realiza o particionamento dos dados, ou seja, a avaliação (teste) do classificador é realizada a partir da utilização do mesmo conjunto de dados que foi utilizado no treino. Esta opção **somente deve ser utilizada em caráter exploratório**, sendo que é altamente propensa ao sobreajuste e não permite avaliar a capacidade de generalização do modelo (capacidade de predição a partir de dados desconhecidos).

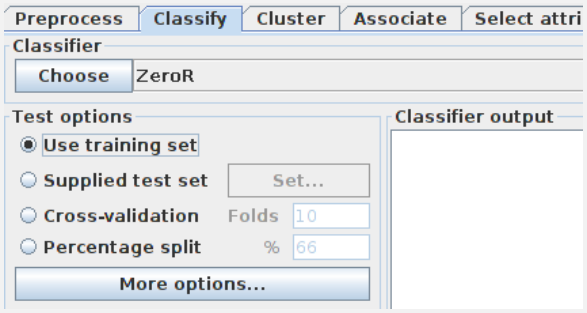

Figura 3. Tarefa de classificação no Weka - Use training set

Fonte: Weka

4 Como avaliar os modelos de classificação treinados

Nesta seção serão apresentados algumas das principais medidas que devem ser utilizadas para avaliar e interpretar os modelos de classificação gerados. As medidas que serão abordadas aqui são: Acurácia geral, matriz de confusão, taxa de verdadeiros positivos, taxa de verdadeiros negativos, valor preditivo positivo, valor preditivo negativo, coeficiente Cohen's Kappa e a curva ROC.

A avaliação dos modelos de classificação pode ser realizada por meio de diferentes métricas que devem ser consideradas em consonância com as características dos dados. Um **erro de classificação** ocorre quando o valor predito pelo classificador é diferente do valor real da variável. O desempenho geral de um modelo de predição pode ser calculado através da sua **exatidão** (também conhecida como **acurácia geral**) que é medida pela quantidade de acertos de classificação cometidos dividido pelo número total de casos na amostra utilizada para o teste (YE, 2003). O cenário de aplicação mais comum quando trabalhamos com a classificação é o de um conjunto de amostras dividido em duas classes e/ou categorias de saída. Nestas situações, o desempenho preditivo do modelo pode ser descrito por meio de uma matriz quadrada de ordem 2 e que é denominada de **matriz de confusão binária** (também conhecida como tabela de

contingência). Na matriz de confusão possuímos os rótulos da classe real observada para as situações de verdadeiro e falso, além dos rótulos para a classe de predição também com as situações de verdadeiro e falso. Conforme pode ser visto na Tabela 1 (HAND; SMYTH; MANNILA, 2001), existem quatro combinações possíveis para os resultados de predição de um classificador binário. Os valores **Verdadeiro Positivo** e **Verdadeiro Negativo** (diagonal principal da tabela) correspondem às respostas corretas do modelo de classificação, e os valores Falso Positivo e Falso Negativo (diagonal secundária da tabela) correspondem às respostas incorretas.

Tabela 1: Matriz de confusão binária

	Total da População	**Dados Reais Observados**	
		Condição = Verdadeiro	**Condição = Falso**
Predição do Modelo	**Predição da condição = Verdadeiro**	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	Predição da condição = Falso	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Considerando a matriz de confusão apresentada, a acurácia geral também pode ser definida pela seguinte fórmula:

$$Acuráciageral = \frac{VP+VN}{VP+FP+FN+VN} \quad (1)$$

A acurácia geral é uma medida importante para observarmos a qualidade de um modelo, mas deve ser analisada com bastante cuidado e nunca de maneira isolada. É possível, por exemplo, que em bases de dados com grande quantidade de casos concentrados em uma determinada classe, o modelo alcance uma acurácia geral alta simplesmente classificando todos os dados como sendo daquela classe predominante. Para evitar esta armadilha, devemos utilizar também outras métricas de avaliação que podem ser calculadas a partir das informações contidas na matriz de confusão. Algumas dessas métricas são: Taxa de Verdadeiros Positivos, Taxa de Verdadeiros Negativos, e o Valor Preditivo Positivo.

A Taxa de Verdadeiros Positivos (TVP) representa a proporção entre a quantidade de casos que foram corretamente classificados como positivos (VP) e a quantidade total de casos positivos (VP + FN). A TVP também é denominada de Sensibilidade (na área de diagnóstico médico) e de Recall ou Revocação (na área de Recuperação da Informação).

$$TVP = \frac{VP}{VP+FN} \quad (2)$$

A Taxa de Verdadeiros Negativos (TVN) representa a proporção entre a quantidade de casos que foram corretamente classificados como negativos (VN) e a quantidade total de casos negativos (VN + FP). A TVN também é conhecida como Especificidade (na área de diagnóstico médico.

$$TVN = \frac{VN}{VN+FP} \quad (3)$$

O Valor Preditivo Positivo (VPP) representa a proporção entre a quantidade de casos que foram corretamente classificados como positivos e a quantidade de exemplos classificados como positivos, sejam eles corretos ou não. O VPP também é conhecido como Precisão (na área de Recuperação da Informação).

$$VPP = \frac{VP}{VP+FP} \quad (4)$$

O Valor Preditivo Negativo (VPN) representa a proporção entre a quantidade de casos que foram corretamente classificados como negativos e a quantidade de exemplos classificados como negativos, sejam eles corretos ou não.

$$VPN = \frac{VN}{VN+FN} \quad (5)$$

Outras métricas que também podem auxiliar bastante o pesquisador no momento de avaliar a qualidade de seus modelos são: o coeficiente Kappa, o espaço ROC (Receiver Operating Characteristic) e a AUC (Area Under the Curve).

O coeficiente estatístico Cohen's Kappa (K) é uma medida popular usada para estimar a concordância entre dados categóricos. A métrica compara a acurácia geral do modelo de classificação com a acurácia esperada para o mesmo caso a classificação fosse realizada ao acaso, indicando assim a intensidade de concordância entre as mesmas. O K varia de 0 a 1, sendo que 0 (zero) indica que não existe nenhuma concordância entre a classificação realizada pelo modelo e os dados que foram observados e 1 indica total concordância (menor que 0 - sem concordância; 0 a 0,20 - concordância muito pobre; 0,21 a 0,40 - fraca; 0,41 a 0,6 - moderada; 0,61 a 0,80 - substancial; e de 0,81 a 1 - quase perfeita). O coeficiente K auxilia a avaliar a possibilidade do modelo de classificação estar em acordo com os dados observados por mero acaso ou não. O cálculo do coeficiente Kappa é realizado a partir da seguinte fórmula:

$$K = \frac{(Acuráciageral - AcuráciaEsperada)}{(1 - AcuráciaEsperada)} \quad (6)$$

sendo que,

$$AcuráciaEsperada(Verdadeiro) = \frac{(VP+FN)*(VP+FP)}{VP+FN+VN+FP} \quad (7)$$

$$AcuráciaEsperada(Falso) = \frac{(VN+FP)*(VN+FN)}{VP+FN+VN+FP} \quad (8)$$

$$AcuráciaEsperada = \frac{AcuráciaEsperada(Verdadeiro)+AcuráciaEsperada(Falso)}{VP+FN+VN+FP} \quad (9)$$

Uma última métrica bastante importante e que deve ser considerada na avaliação dos modelos é a **ROC** (*Receiver Operating Characteristic*) que é comumente utilizada a partir da leitura da **AUC** (*Area Under the Curve*) (ver figura 4). Um espaço ROC é uma representação bidimensional do desempenho de um classificador binário. Este espaço bidimensional é projetado em dois eixos, ambos com intervalo entre [0-1], sendo no eixo x representada a taxa de falsos positivos, ou sensibilidade, e no eixo y a taxa de verdadeiros positivos, ou complemento da especificidade, calculado por 1-especificidade (BRADLEY, 1997). Assim, um espaço ROC representa a relação entre os benefícios, ou verdadeiros positivos, e os custos, ou falsos positivos, de um conjunto de amostras classificadas por determinado modelo (FAWCETT, 2006). Neste contexto, o ponto nas coordenadas (0,1) representaria um classificador perfeito, que não aponta falsos positivos. Já um classificador binário aleatório, estaria posicionado sobre uma linha diagonal neste gráfico, para todo y=x. Esta linha liga os pontos (0,0) e (1,1).

No entanto, nem sempre um classificador binário produz um único ponto no espaço ROC. Alguns classificadores, tais como redes neurais ou classificadores probabilísticos, geram, como saída, duas probabilidades ou escores, que representam o nível ou probabilidade de pertinência de uma amostra para cada uma das duas classes possíveis. Nestes casos, há a necessidade de definição de um limiar que permita ao classificador transformar estes dois valores contínuos em uma saída binária. Podem ser testados diferentes valores reais para este limiar, o que irá produzir diferentes pontos em um espaço ROC. Quando interligados, estes pontos formam uma curva no espaço ROC. O modelo com a maior Área sobre a Curva pode ser considerado o mais efetivo, e o limiar ótimo para o modelo é aquele que estiver mais próximo ao canto superior esquerdo do gráfico, que representaria o classificador perfeito no ponto (0,1) (KUHN; JOHNSON, 2013).

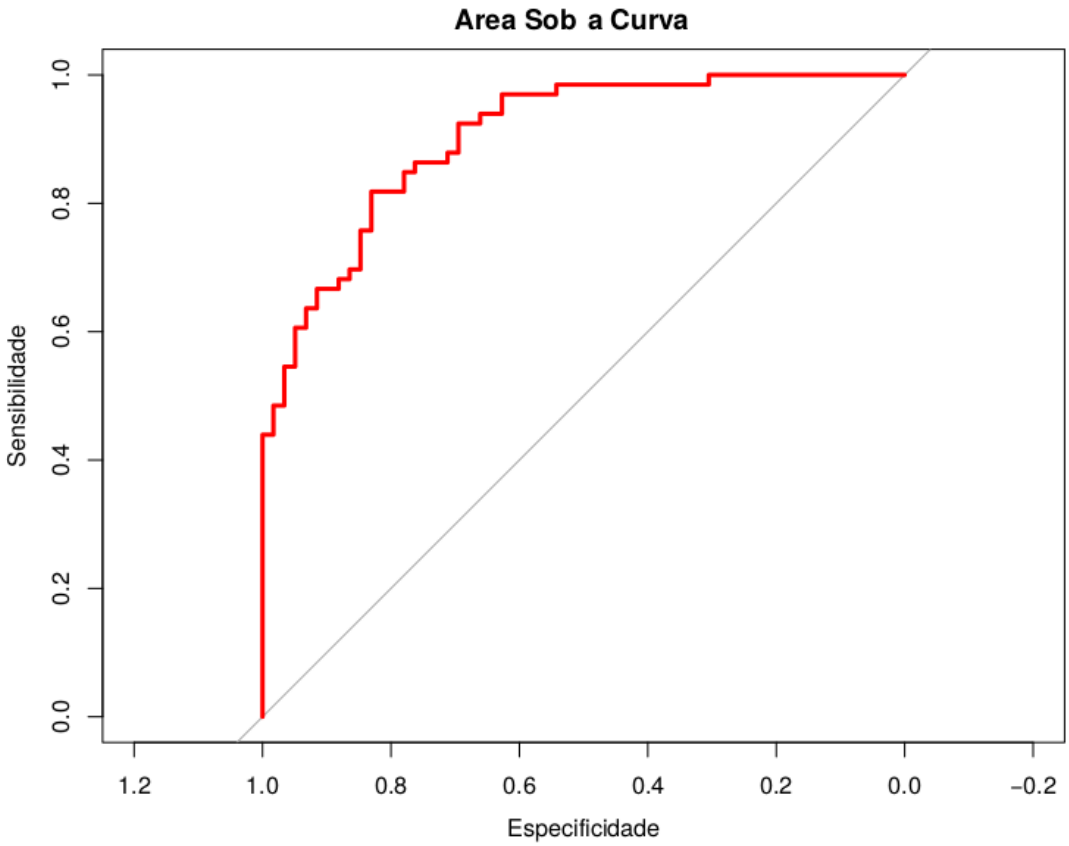

Figura 4. Área sob a curva (AUC)

5 Exemplos ilustrativos

Agora que conhecemos alguns dos principais aspectos que devem ser observados durante a geração e avaliação de modelos de classificação, vamos explorar alguns cenários do contexto educacional em que esses conceitos podem ser aplicados.

5.1 Transformando variáveis: de numérico para nominal

Consideremos um conjunto de dados contendo informações sobre a evasão de acadêmicos em um curso superior. Utilizando modelos de classificação (por exemplo, baseados em árvores de decisão) é possível avaliar a existência de padrões na evasão dos acadêmicos (CAMARGO; SANTOS; CAMARGO, 2012). Consideremos que a base de dados em questão possui as seguintes variáveis:

- Sexo (Qualitativa nominal): Masculino (M) ou Feminino(F)
- Forma_Ingresso (Qualitativa nominal): ENEM, Vestibular, Reopção de Curso, Transferência Ex-oficio, Transferência Externa, Transferência Interna e Portador de Diploma
- Ano_Ingresso (Qualitativa nominal): Ano em que o acadêmico ingressou no curso
- Idade_Ingresso (Quantitativa discreta): Idade do acadêmico quando ingressou no curso
- Forma_Evasao (Qualitativa nominal): Cancelamento, Abandono, Transferência Interna, Reopção de Curso, Desligamento, Aluno regular.

Após abrir a base na ferramenta WEKA, observamos que a variável **Ano_Ingresso** está sendo tratada como variável quantitativa, uma vez que os valores armazenados nas tabelas são numéricos (ver figura 5). Entretanto, os valores referentes aos anos de ingresso não representam quantidades, mas sim categorias. Para essa variável, faz mais sentido utilizar as quantidades de ingressantes de cada um dos anos (categorias da variável) do que calcular uma média dos diferentes anos em que os acadêmicos ingressaram no curso, por exemplo. Na etapa de pré-processamento, a variável **Ano_Ingresso** pode ser transformada de Numérica para Nominal por meio da aplicação de um filtro. A simples transformação dessa variável de numérica para categórica reflete em um aumento imediato (ainda que pequeno) na acurácia geral de um modelo de classificação gerado por meio do algoritmo J48 (árvores de decisão) e para um treinamento utilizando a validação cruzada com 10 partições (aumento de 62% de acurácia geral para 64.4%).

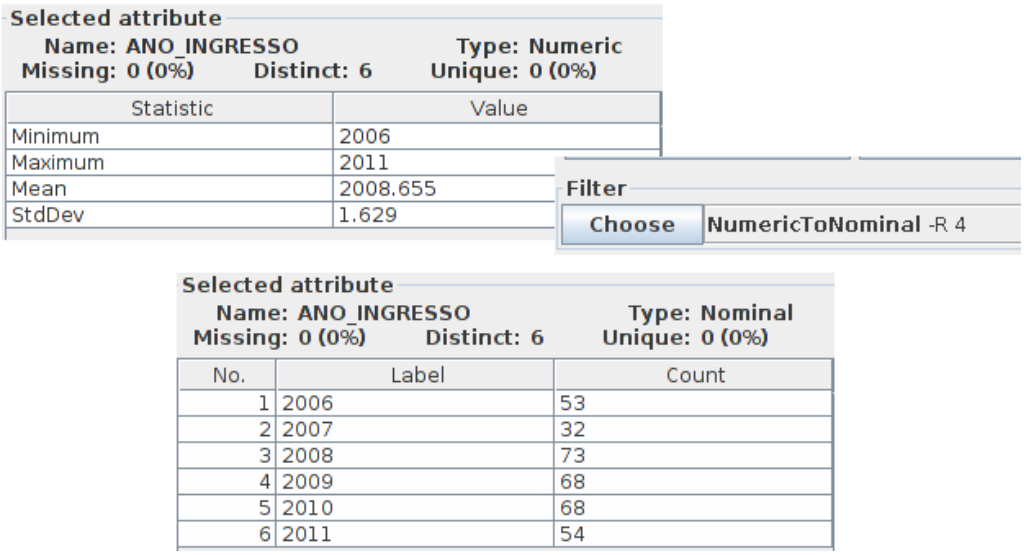

Figura 5. Transformação de atributo: de numérico para nominal

Esse caso ilustrativo utiliza a base de dados **dados_evasao02_numericaParaNominal.zip** disponível em https://github.com/cristiancechinel/bookchapter.

5.2 Transformando variáveis: binarização

Quando olhamos as características da variável que está sendo classificada pelo modelo (**Forma_Evasao**), percebemos que as diferentes categorias existentes estão bastante desbalanceadas (ver figura 6). Na variável, 55% dos acadêmicos estão concentrados na classe **Aluno_regular**, 34% na classe Abandono, 4% na classe Cancelamento e os demais casos nas diferentes classes restantes. Esse desbalanceamento dos dados na variável de saída interfere diretamente nos desempenhos dos modelos que normalmente encontram dificuldades em classificar dados pertencentes às classes minoritárias e tendem a favorecer a classificação nas classes mais comuns. A matriz de confusão da figura 6 a seguir oferece uma visualização clara da dificuldade que o modelo possui em classificar os casos das classes minoritárias.

Selected attribute

Name: FORMA_EVASA0 **Type: Nominal**
Missing: 0 (0%) **Distinct: 8** **Unique: 0 (0%)**

No.	Label	Count
1	Cancelamento	17
2	Aluno_Regular	194
3	Abandono	120
4	Transferencia_Interna	4
5	Reopcao_de_Curso	4
6	Transferencia	4
7	Desligamento	3
8	Classificado_e_Nao_Matriculado	2

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   <-- classified as
   0  13   4   0   0   0   0   0 |   a = Cancelamento
   0 166  28   0   0   0   0   0 |   b = Aluno_Regular
   0  59  58   1   0   0   0   2 |   c = Abandono
   0   3   1   0   0   0   0   0 |   d = Transferencia_Interna
   0   1   3   0   0   0   0   0 |   e = Reopcao_de_Curso
   0   3   1   0   0   0   0   0 |   f = Transferencia
   0   2   1   0   0   0   0   0 |   g = Desligamento
   0   0   2   0   0   0   0   0 |   h = Classificado_e_Nao_Matriculado
```

Figura 6. Transformação de atributo: binarização

Uma segunda transformação que podemos fazer nessa base de dados é binarizar a variável de saída, ou seja, transformar cada uma das categorias em uma nova variável binária que pode assumir os valores 0 ou 1, sendo que 1 significa que a situação daquela categoria está ocorrendo, e 0 significa que não está ocorrendo. Considerando que possuímos 348 instâncias e que 194 delas pertencem a categoria **Aluno_Regular,** uma binarização da variável **Forma_Evasao** geraria uma variável **Forma_Evasao=Aluno_Regular** contendo 194 instâncias pertencentes a categoria 1 (aluno regular) e 154 pertencentes a categoria 0 (aluno não regular ou evadido). Essa nova variável apresenta um balanceamento entre as classes bem melhor do que a anterior e pode ser agora adotada como a variável de saída a ser classificada pelo modelo.

Uma nova rodada de treinamento e teste de um modelo de classificação por meio do algoritmo J48 (utilizando validação cruzada com 10 partições) irá apresentar uma melhoria na acurácia geral em comparação com o modelo da seção anterior (aumentando de 64,4% para 69,5% a acurácia geral), assim como gerando taxas de verdadeiros positivos e verdadeiros negativos mais próximas entre as categorias classificadas (que agora são somente 2). É importante considerar que essa última transformação acarreta em uma perda na capacidade de predição do modelo no que se refere ao tipo de evasão do acadêmico, uma vez que agora o modelo é capaz de predizer apenas se o acadêmico é um aluno regular ou se ele evadiu (sem precisar de que maneira essa evasão ocorreu).

Esse caso ilustrativo utiliza a base de dados **dados_evasao03_NominalParaBinaria.zip** disponível em https://github.com/cristiancechinel/bookchapter.

5.3 Interpretando uma árvore de decisão para predição da evasão de estudantes

Consideremos um conjunto de dados contendo a matrícula dos estudantes, as notas finais de três disciplinas do primeiro semestre (Algoritmos, Cálculo I e Geometria Analítica) e a situação final do estudante (evadido ou formado). Gostaríamos de gerar um modelo capaz de predizer se um estudante conseguirá se formar ou irá evadir do curso em que está matriculado, mas também desejamos interpretar o modelo de maneira a compreender melhor os fatores que estão relacionados com a evasão ou sucesso nesse cenário específico. O modelo de classificação é gerado por meio do algoritmo J48 utilizando validação cruzada com 10 partições e a variável referente ao número da matrícula é desconsiderada na mineração.

O modelo é capaz de classificar corretamente 92.46% dos casos (acurácia geral) e apresenta um coeficiente Kappa de 0.85. Ainda, as acurácias para classificação das classes evadido e formado estão bem equilibradas sendo de 93.8% e 90.8% respectivamente. É possível dizer que a árvore de decisão gerada é capaz de predizer a evasão e o sucesso do estudante com um ótimo desempenho.

```
Algoritmos <= 5.3
|   Algoritmos <= 4: Evadido (83.0/1.0)
|   Algoritmos > 4
|   |   CalculoI <= 5.2: Evadido (16.0)
|   |   CalculoI > 5.2: Formado (6.0/1.0)
Algoritmos > 5.3
|   Algoritmos <= 7.2
|   |   CalculoI <= 5
|   |   |   Algoritmos <= 6.9: Evadido (11.0/1.0)
|   |   |   Algoritmos > 6.9: Formado (4.0/1.0)
|   |   CalculoI > 5: Formado (14.0/1.0)
|   Algoritmos > 7.2: Formado (65.0/1.0)

Number of Leaves  :     7

Size of the tree :      13
```

Figura 7. Árvore de decisão para predição da evasão

Ao observar a árvore gerada (figura 7), é possível perceber que a variável mais importante na classificação é a nota da disciplina de Algoritmos. De modo geral, e pela leitura da árvore de decisão, estudantes que alcançam nota superior a 7.2 na disciplina de Algoritmos se formam com sucesso, e estudante que tiram notas inferiores a 4 nessa disciplina evadem. Nas situações em que as notas de Algoritmos estão entre esses limiares, a nota da disciplina de Cálculo I passa a ter importância para a predição da evasão. A nota da disciplina de Geometria Analítica não foi utilizada pelo modelo.

Esse caso ilustrativo utiliza a base de dados **dados-notas-versus-disciplina-versus-evasao.zip** disponível em https://github.com/cristiancechinel/bookchapter.

6 Resumo

Nesse capítulo foram apresentadas algumas das principais características que devem ser observadas nos dados durante o processo de geração de modelos de classificação no contexto educacional (origens possíveis dos dados, tipos de variáveis e atributos), juntamente com algumas possibilidades de transformação nos dados e de sua dimensionalidade. Foram vistos também os principais cuidados que o pesquisador deve tomar durante a construção e avaliação desses modelos, sobretudo com respeito a divisão dos dados de treinamento e teste em conjuntos distintos e as possibilidades de estratégias de particionamento de dados existentes. Por último, foram apresentadas algumas medidas para a avaliar o desempenho dos modelos de classificação construídos.

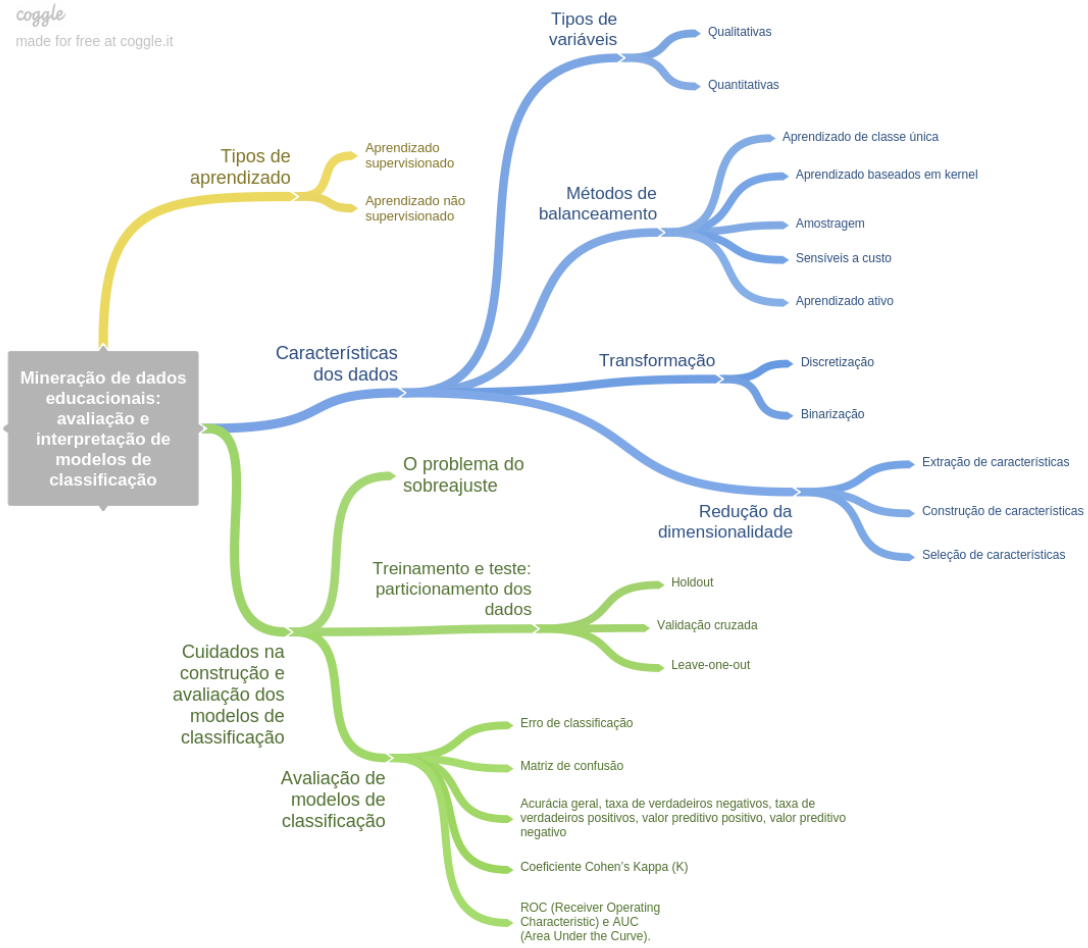

Figura 8. Mapa mental da avaliação e interpretação dos modelos de classificação.

7 Leituras recomendadas

Mineração de dados educacionais: conceitos, técnicas, ferramentas e aplicações. (COSTA et al., 2013). Neste capítulo você irá encontrar uma breve explicação sobre as diferentes tarefas existentes na mineração de dados (Classificação e regressão, agrupamento e associação) e sobre o funcionamento de alguns de seus algoritmos. O capítulo também traz bons exemplos da área de mineração de dados educacionais.

***A Survey on Pre-Processing Educational Data* (ROMERO; ROMERO; VENTURA, 2014).** Este capítulo apresenta um apanhado geral sobre diferentes estratégias de pré-processamento especificamente voltadas para o contexto educacional. Os autores abordam as características mais comumente encontradas em bases de dados educacionais e apresentam alternativas para o pré-processamento dessas bases.

8 Base de dados exemplo

Base de dados utilizada nos cenários ilustrativos do capítulo 5 e relacionadas a evasão de estudantes (CAMARGO; SANTOS; CAMARGO, 2012): https://github.com/cristiancechinel/bookchapter

9 Checklist

De maneira geral, o processo de mineração e avaliação de modelos de classificação deve seguir os seguintes passos específicos:

- Observação das características dos dados coletados e verificação da necessidade de transformação dos dados, ou de geração de novos atributos derivados
- Verificar o balanceamento dos dados e realizar operações de balanceamento caso sejam necessárias. Levar em consideração que o desbalanceamento dos dados interfere no desempenho dos modelos de classificação.
- Observar a dimensionalidade dos dados e avaliar a necessidade de selecionar as características (atributos) mais relevantes para serem utilizados no processo de mineração.
- Selecionar o método de particionamento mais adequado para ser utilizado na etapa de treinamento e teste
- Definir os algoritmos de mineração e rodar os experimentos.
- Avaliar o desempenho dos modelos utilizando diferentes medidas e considerando as características específicas dos dados em questão.

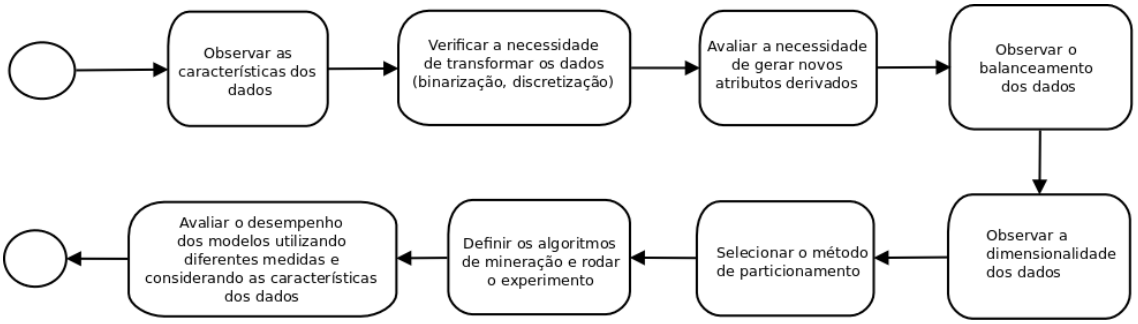

Figura 9. Fluxograma para o processo de mineração, avaliação e interpretação dos modelos de classificação

10 Exercícios

1. **Exercício 1 - Abordagens de aprendizado de máquina.** Quais as principais diferenças entre o aprendizado de máquina supervisionado e não supervisionado, e quando devemos utilizar cada um deles?
2. **Exercício 2 - Importância da Acurácia Geral.** Qual a importância da medida de avaliação Acurácia Geral e quais os cuidados que devem ser tomados ao utilizar essa medida como referência para medição do desempenho de um modelo de classificação?
3. **Exercício 3 - Uso da base de dados.** Faça o download da base de dados disponibilizada na seção 8 e realize a mineração da mesma utilizando o algoritmo J48 com um particionamento de validação cruzada com 10 partições. Use como variável alvo a **Forma_Evasao** e observe o desempenho do modelo de predição gerado. Em seguida, reproduza as transformações propostas nos exemplos ilustrativos das seções 5.1 (Transformando variáveis: de numérico para nominal) e 5.2 (Transformando variáveis: binarização) e compare os resultados com os do modelo inicial.
4. **Exercício 4 - O problema da acurácia geral alta.** Imaginemos uma situação problema em que se deseja desenvolver um modelo para predizer com antecedência se um determinado estudante irá evadir de um determinado curso ou não. A variável **Evasão** pode então assumir os valores Verdadeiro (o acadêmico evadiu do curso) ou Falso (o acadêmico não evadiu do curso). Para um conjunto de 200 estudantes (190 de concluintes e 10 evadidos) foi gerado um modelo de classificação com a seguinte matriz de confusão:

Tabela 2: Matriz de confusão - Modelo de classificação de estudantes evadidos - exemplo 1

	Total da População	**Dados Reais Observados**		**Total**
		Condição = Verdadeiro	**Condição = Falso**	

Predição do Modelo	**Predição da condição = Verdadeiro**	0 (VP)	0 (FP)	0
	Predição da condição = Falso	10 (FN)	190 (VN)	200
		10	190	200

Quando calculamos a acurácia geral deste modelo, temos um percentual bastante alto de casos que foram classificados corretamente (190/200 = 95%). Entretanto, o modelo apresenta um sério problema, uma vez que ele é incapaz de classificar corretamente os estudantes evadidos (VP = 0). Na verdade, o modelo simplesmente classifica todos os casos da base de dados como concluintes, gerando assim uma **alta taxa de acurácia geral que esconde a sua má qualidade de desempenho**. As demais métricas de avaliação nos ajudam a perceber que o modelo apresenta uma má qualidade de predição. Veja que a Taxa de Verdadeiro Positivo e o Valor Preditivo Positivo são iguais a zero.

- Acurácia geral = 190/200 = 95%
- Taxa de Verdadeiro Positivo = 0/(0+10) = 0%
- Taxa de Verdadeiro Negativo = 190/(190+0) = 100%
- Valor Preditivo Positivo = 0/(0+0) = (não existe) 0%
- Valor Preditivo Negativo = 190/200 = 95%

Observe a diferença da matriz de confusão do modelo de classificação anterior para a matriz de confusão desse novo modelo apresentado a seguir. Calcule os valores da Acurácia geral, Taxa de Verdadeiro Positivo, Taxa de Verdadeiro Negativo, Valor Preditivo Positivo e Valor Preditivo Negativo e compare com as medidas de avaliação de desempenho do modelo anterior.

Tabela 3: Matriz de confusão - Modelo de classificação de estudantes evadidos - exemplo 2

	Total da População	**Dados Reais Observados**		**Total**
		Condição = Verdadeiro	**Condição = Falso**	
Predição do Modelo	**Predição da condição = Verdadeiro**	8 (VP)	8 (FP)	16
	Predição da condição = Falso	2 (FN)	182 (VN)	184
		10	190	200

- **Exercício 5 - Calculando o Coeficiente Kappa.** A melhoria no desempenho dos modelos de predição comentados no exercício anterior também pode ser observada quando calculamos o coeficiente Kappa para os modelos. Realize o cálculo dos coeficientes Kappa para as matrizes de confusão das tabelas 2 e 3 do exercício anterior. Observe as diferenças dos valores dos coeficientes e como os mesmos permitem realizar a avaliação de qual modelo apresenta um melhor desempenho para o problema em questão.
- **Exercício 6 – Interpretando um modelo de predição da evasão.** Reproduza o modelo de predição do exemplo ilustrativo das seção 5.3 utilizando a base de dados **dados-notas-versus-disciplina-versus-evasao.zip** disponível em https://github.com/cristiancechinel/bookchapter.

11 Referências

ATTENBERG, J.; ERTKIN, S. Class imbalance and active learning. In: **Imbalanced learning: foundations, algorithms, and applications**. Hoboken, New Jersey: John Wiley & Sons, Inc., 2013. p. 101–150. ISBN 9781118074626.

BISHOP, C. M. **Neural Networks for Pattern Recognition**. New York, NY, USA: Oxford University Press, Inc., 1995. ISBN 0198538642.

BRADLEY, A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. **PATTERN RECOGNITION**, v. 30, n. 7, p. 1145–1159, 1997.

CAMARGO, F. N. P.; SANTOS, H. L. dos; CAMARGO, S. da S. Aplicação de técnicas de modelagem computacional para predição de desempenho de estudantes. In: Anais da V Conferência Sul em Modelagem Computacional. Rio Grande: Editora da FURG, 2012. v. 1, p. 155–160.

CAMARGO, S. da S. **Um modelo neural de aprimoramento progressivo para redução de dimensionalidade**. Tese (Doutorado em Ciência da Computação) — Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Junho 2010.

CAMARGO, S. da S.; BORIN, J. M.; FERREIRA, A. P. L. Identifying dropout patterns by using data mining techniques: A study case. In: **Proceedings of the 2014 Latin American Computing Conference (CLEI)**. Montevideo: Curran Associates, Inc., 2014. v. 1, p. 690–698.

CASTRO, C. L. de; BRAGA, A. A. P. A. Aprendizado supervisionado com conjuntos de dados desbalanceados. **SBA: Controle & Automação**. Campinas: Sociedade Brasileira de Automatica, v. 22, n. 5, p. 441–466, out 2011. ISSN 0103-1759.

CECHINEL, C.; ARAUJO, R. M.; DETONI, D. Modelagem e predição de reprovação de estudantes de cursos de educação a distância a partir da contagem de interações. **Revista Brasileira de Informática na Educação**, v. 23, n. 3, p. 1–11, dez 2015.

CECHINEL, C. et al. Mining models for automated quality assessment of learning objects. **Journal of Universal Computer Science**, v. 22, n. 1, p. 94–113, jan 2016.

COSTA, E. et al. Mineração de dados educacionais: Conceitos, técnicas, ferramentas e aplicações. In: **Anais da Primeira Jornada de Atualização em Informática na Educação (JAIE 2012)**. Rio de Janeiro: Sociedade Brasileira de Computação, 2012. v. 1, p. 1–29.

FAWCETT, T. An introduction to roc analysis. **Pattern Recognition Letters**, Elsevier Science Inc.: New York, NY, USA, v. 27, n. 8, p. 861–874, jun 2006. ISSN 0167-8655.

HALL, M. et al. The weka data mining software: An update. **SIGKDD Explor. Newsl.** New York, NY, USA: ACM, v. 11, n. 1, p. 10–18, nov 2009. ISSN 1931-0145.

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

HAND, D. J.; SMYTH, P.; MANNILA, H. **Principles of Data Mining**. Cambridge, MA, USA: MIT Press, 2001. ISBN 9780262082907.

HE, H.; MA, Y. **Imbalanced Learning: Foundations, Algorithms, and Applications**. 1st. ed. Hoboken,NJ: Wiley-IEEE Press, 2013. ISBN 9781118074626.

KUHN, M.; JOHNSON, K. **Applied Predictive Modeling**. New York, NY, USA: Springer, 2013. ISBN 978-1-4614-6849-3.

ICKE, I. Wikimedia Commons. 2008. Disponível em <https://commons.wikimedia.org/wiki/File%3AOverfitting.svg>. Acesso em: 30 de Julho de 2018.

KAPS, M.; LAMBERSON, W.R. **Biostatistics for Animal Science**. Columbia, USA: CABI Publishing, 2004.

LAROSE, D. T. **Data Mining Methods and Models**. New York, NY, USA: John Wiley & Sons, Inc., 2006. ISBN 9780470227732.

OLSON, D. L.; DELEN, D. **Advanced Data Mining Techniques**. 1st. ed. Berlin: Springer Publishing Company, Incorporated, 2008. ISBN 9783540769163.

QUEIROGA, E.; CECHINEL, C.; ARAúJO, R. Predição de estudantes com risco de evasão em cursos técnicos a distância. In: **Anais do XXVIII Simpósio Brasileiro de**

Informática na Educação (SBIE 2017). Recife: Sociedade Brasileira de Computação, 2017. v. 1, p. 1547–1556.

ROLIM, V. B.; MELLO, R. F. L. de; COSTA, E. de B. Monitoring educational forums using machine learning (utilização de técnicas de aprendizado de máquina para acompanhamento de fóruns educacionais). **Brazilian Journal of Computers in Education (Revista Brasileira de Informática na Educação - RBIE)**, v. 25, n. 3, p. 112–130, 2017. ISSN 2317-6121.

ROMERO, C.; ROMERO, J.; VENTURA, S. A survey on pre-processing educational data. **Studies in Computational Intelligence**, v. 524, p. 29–64, 2014.

ROMERO, C. et al. Data mining algorithms to classify students. In: **Proceedings of The 1st International Conference on Educational Data Mining**. [S.l.: s.n.]. p. 8–17, 2008.

SANTOS, F. D.; BERCHT, M.; WIVES, L. Classificação de alunos desanimados em um AVEA: uma proposta a partir da mineração de dados educacionais. In: **Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015)**. Maceió: Sociedade Brasileira de Computação, 2015. p. 1052–1061.

SANTOS, H. dos; CECHINEL, C. Geração automática de avaliações de objetos de aprendizagem por meio de mineração de textos. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. Maceió: Sociedade Brasileira de Computação, 2015. v. 4, n. 1, p. 1007–1015.

SANTOS, H. L. dos; CAMARGO, F. N. P.; CAMARGO, S. da S. Predizendo o sucesso de estudantes através do uso avaliações formativas em avas. In: **Anais dos Workshops do CBIE 2012**. Curitiba: Sociedade Brasileira de Computação, 2012.

WANG, L.; XIUJU, F. **Data Mining with Computational Intelligence**. Berlin, Heidelberg: Springer-Verlag, 2009. ISBN 9783642063879.

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

YE, N. **The Handbook of Data Mining**. London: Taylor & Francis, 2003. ISBN 9780805855630.

Sobre os autores

Cristian Cechinel

http://lattes.cnpq.br/2782164252734586

Doutor em Ingeniería de la Información y del Conocimiento pela Universidad de Alcalá (Espanha). É professor Adjunto da Universidade Federal de Santa Catarina. Tem experiência nas áreas mineração de dados educacionais e Learning Analytics, na construção de ferramentas para o apoio ao ensino, e na busca de métricas para a avaliação da qualidade de recursos educacionais dentro de repositórios. Participa ativamente da Comunidade Latino-Americana de Tecnologias de Aprendizagem (LACLO). Na área de mineração de dados educacionais, possui projetos aprovados junto a agências de financiamento como o CNPq (2017-2019) e a Agência Nacional de Investigación e Innovación (ANII - Uruguai) (2017-2018).

Sandro da Silva Camargo

http://lattes.cnpq.br/8826344853104147

Doutor em Computação pela Universidade Federal do Rio Grande do Sul. É professor Adjunto da Universidade Federal do Pampa. Atua na área de informática na educação há mais de 17 anos. Tem experiência nas áreas de aprendizado de máquina e learning analytics, e na construção de ferramentas para o apoio ao ensino. Entre 2014 e 2015, atuou como assessor da reitoria da Universidade Federal do Pampa com foco em gestão acadêmica baseada em dados.