
**UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO**
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

ESRON DTAMAR DA SILVA

# USO DA TEORIA DA DISTÂNCIA TRANSACIONAL NA PREDIÇÃO DA EVASÃO NA EAD: O CASO DA UNIVASF.

**JUAZEIRO - BA**
**2019**

**UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO**
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

ESRON DTAMAR DA SILVA

# USO DA TEORIA DA DISTÂNCIA TRANSACIONAL NA PREDIÇÃO DA EVASÃO NA EAD: O CASO DA UNIVASF.

Trabalho apresentado à Universidade Federal do Vale do São Francisco - Univasf, *Campus* Juazeiro, como requisito da obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Jorge Luis Cavalcanti Ramos

**JUAZEIRO - BA**
**2019**

# UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO
# CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

## FOLHA DE APROVAÇÃO

## ESRON DTAMAR DA SILVA

## USO DA TEORIA DA DISTÂNCIA TRANSACIONAL NA PREDIÇÃO DA EVASÃO NA EAD: O CASO DA UNIVASF.

Trabalho apresentado à Universidade Federal do Vale do São Francisco - Univasf, ***Campus*** Juazeiro, como requisito da obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Jorge Luis Cavalcanti Ramos

**Aprovado em: _____ de _____ de 2019**

**Banca Examinadora**

---

**João Carlos Sedraz Silva, Doutor,**
**Universidade Federal do Vale do São Francisco**

---

**Jorge Luis Cavalcanti Ramos, Doutor,**
**Universidade Federal do vale do São Francisco**

---

**Romulo Calado Pantaleão Câmara, Doutor,**
**Universidade Federal do Vale do São Francisco**

*"É tudo computador..."*
**(Julinho da Van, 2018)**

## AGRADECIMENTOS

Agradeço a todos que tiveram paciência comigo durante todos esses anos. Em especial, aos meus companheiros da graduação e meus colegas de trabalho. Agradeço a minha mãe, Dona Tânia, que me apoiou desde o começo e nunca me abandonou. Agradeço aos professores, que deram lições importantes e abriram meus olhos para um mundo que eu jamais descobriria sozinho. Agradeço a paciência do meu orientador professor Jorge, que me deu motivação para continuar quando tudo parecia perdido.

Muito obrigado Gabriel, Johnathan, Gustavo, Leonardo, Matheus Brian, Vitão e todos os amigos que tive a oportunidade de compartilhar experiências, dificuldades e mesas de bar durante a graduação.

Muito obrigado a Laury, seu apoio e seu companheirismo foram muito importantes nos últimos momentos da realização deste trabalho.

“A educação é a arma mais poderosa que você pode usar para mudar o mundo...”

**Nelson Mandela**

**RESUMO**

Os avanços do acesso às tecnologias da informação criou um ambiente fértil para a pesquisa na área de educação a distância (EAD). Porém, apesar da grande disponibilidade e flexibilidade, os cursos da modalidade EAD, no Brasil, ainda sofrem com o problema da evasão de estudantes. Acompanhando o crescimento da EAD, se desenvolve também a área de Mineração de Dados Educacionais (MDE). Este trabalho propõe a utilização de uma metodologia fundamentada em técnicas de MDE com o objetivo de construir e avaliar modelos de classificação da situação final de estudantes da EAD entre duas classes, evadidos e não evadidos, usando os algoritmos de aprendizagem de máquina: KNN, Regressão Logística e Árvore de Decisisão. Na construção dos modelos, foram utilizados dados obtidos de duas turmas no contexto da Universidade Federal do Vale do São Francisco (UNIVASF), e, além disso, as variáveis preditoras foram concebidas a partir dos construtos da Teoria da Distância Transacional (TDT). Os resultados apontam que essas variáveis podem ser usadas como parâmetros dos classificadores, alcançando valores similares aos verificados em trabalhos relacionados com a predição de evasão de estudantes na EAD.

**Palavras-chave**: Ciência de Dados. Aprendizagem Supervisionada. Descoberta de Conhecimentos em Bases de Dados

**ABSTRACT**

The advances in information technologies created a very fertile research environment in the distance education field. However, despite of the great disponibility and flexibility, the brazilian DE course genre still suffers from the problem of student dropout. Along with the growing in DE, the Educational Data Mining (EDM) is developing as well. This paper proposes the use of a methodology based on EDM techniques with the purpose of constructing and evaluating classification models of situation of distance learning students between two classes, evaded and not evaded using machine learning algorithms: KNN, Logistic Regression and Decision Tree. In the construction of the models, we used data obtained from two classes in the context of the Federal University of Vale do Francisco (UNIVASF), and, moreover, the predictor variables were conceived from the Transactional Distance Theory (TDT) constructs. The results indicate that these variables can be used as parameters from the classifiersm reaching values similar to those found in works related to predicting student dropout in DE.

**Key-words**: Data Science. Supervised Learning. Knowledge Discovery in Databases

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE QUADROS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| ABED | Associação Brasileira de Educação a Distância |
| AVA | Ambiente Virtual de Aprendizagem |
| CFA | Análise Fatorial Confirmatória |
| CSV | *Comma Separeted Value* |
| DM | *Data Mining* |
| DT | Distância Transacional |
| EAD | Educação a Distância |
| EDM | *Educational Data Mining* |
| IDE | *Integrated Development Environment* |
| IES | Instituição de ensino superior |
| INEP | Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira |
| KDD | *Knowledge Discovery in Databases* |
| KNN | *K-nearest Neighbors* |
| LA | *Learning Analytics* |
| LMS | *Learning Management System* |
| ML | *Machine Learning* |
| RL | Regressão Logística |
| SEAD | Secretaria de Educaçao a Distância |
| SGBD | Sistema de Gerenciamento de Banco de Dados |
| SQL | *Structured Query Language* |
| STI | Secretaria de Tecnologia da Informação |
| SVM | *Support Vector Machine* |
| TFT | Taxa de Falsos Positivos |
| TFN | Taxa de Falsos Negativos |

| | |
|---|---|
| TDT | Teoria da Distância Transacional |
| UNIVASF | Universidade Federal do Vale do São Francisco |

**SUMÁRIO**

# 1 INTRODUÇÃO

O desafio de levar educação e formação profissional a lugares remotos, onde, dificilmente, a formação presencial tradicional conseguiria alcançar de maneira efetiva, é uma das principais bandeiras da Educação a Distância (EAD). Entretanto, outros desafios surgem em decorrência da expansão da modalidade: como garantir a qualidade dessa formação? Como transpor modelos de educação presencial para a distância? Como atuar de maneira a prevenir e reduzir os altos índices de evasão, ainda verificados na modalidade? São questões como essas que devem ser respondidas a partir do desenvolvimento de pesquisas nessa modalidade. O uso de novas tecnologias e de novos processos podem contribuir com essas pesquisas.

Diversas iniciativas reforçam o crescimento da EAD e exigem uma atenção maior nos aspectos importantes para a consolidação e manutenção das atividades dessa modalidade nas instituições. Dentre esses aspectos, está a necessidade de pesquisas, como forma de se agregar procedimentos validados cientificamente, ferramentas de gestão mais eficientes e metodologias inovadoras, capazes de superar grandes desafios impostos pela EAD.

O avanço da modalidade de EAD requer o desenvolvimento de recursos que permitam o acompanhamento de cursos oferecidos em um ambiente virtual de aprendizagem. Esses recursos podem ser obtidos a partir de metodologias de análise que envolvem o conhecimento das estratégias pedagógicas dos cursos EAD, o levantamento das necessidades apontadas pelos profissionais que atuam na área e na elicitação dos requisitos para implementação de ferramentas de visualização de dados de diversas atividades dentro de um contexto educacional. (RAMOS, 2016)

Com a expansão da EAD de maneira responsável e planejada, com infraestrutura compatível e recursos humanos qualificados, será possível a oferta de novos cursos pelas instituições, disseminando conhecimento e possibilitando mais oportunidades para o desenvolvimento regional.

Para Ramos (2016), os altos índices de evasão dos alunos em cursos EAD representam um grande desafio para todos os que atuam na modalidade. Além desses índices estarem em níveis elevados, observou-se também que estão em crescimento. Com isso, há uma necessidade contínua de desenvolvimento de pesquisas que apontem caminhos, métodos e ferramentas que os auxiliem a enfrentar melhor esse problema. O uso de técnicas estatísticas e de mineração de dados, em conjunto com teorias consolidadas na modalidade, pode fundamentar modelos eficientes de detecção precoce do risco de evasão pelos alunos.

No estudo apresentado por Ramos (2016), foram desenvolvidos, testados e validados, modelos preditivos da evasão de estudantes de graduação em cursos ofertados na modalidade

EAD, tomando como base as variáveis que compõem cada um dos construtos da Teoria da Distância Transacional (MOORE, 2008). Essa pesquisa ocorreu a partir dos dados de cursos de licenciatura em Biologia e Pedagogia, ambos ofertados por EAD, na Universidade de Pernambuco (UPE).

A citada pesquisa testou cinco algoritmos de classificação para definição dos modelos preditivos: Árvore de Decisão, Máquina de Vetor de Suporte (SVM, do inglês, *Support Vector Machine*), Rede Neural Artificial, K-Vizinhos Mais Próximos (KNN, do inglês, *K-nearest Neighbors*) e Regressão Logística, sendo este último o que apresentou resultados mais relevantes, embora os demais não ficaram muito distantes, nas métricas analisadas.

A partir dessa referência, este estudo foi desenvolvido no sentido de verificar se o mesmo conjunto de variáveis usadas e os algoritmos de classificação podem, também, ser replicados e validados em outro cenário educacional. Desta vez nos cursos de graduação em Administração Pública e na Licenciatura em Biologia, ofertados também por EAD, mas pela Universidade Federal do Vale do São Francisco (UNIVASF).

Algumas adpatações no processo de replicação do estudo foram necessários, tais como: mudança de tecnologia, ajuste nos scripts de coleta de dados e redução de cinco para três algoritmos de classificação. Essas alterações não alteraram os objetivos do trabalho, apenas forneceram novas e adequadas condições para o seu desenvolvimento.

Assim, a principal questão a ser esclarecida neste trabalho é se um conjunto de variáveis representativas da Teoria da Distância Transacional (TDT) e os alguns dos algoritmos classificadores, também, podem ser usados em modelos preditivos de evasão na EAD, em um cenário diferente do apresentado por Ramos (2016).

Espera-se com este trabalho contribuir para o fortalecimento da EAD, além de fomentar a linha de pesquisa voltada para o estudo das tecnologias educacionais, tão evidenciadas e diversificadas, a partir do uso cada vez maior das tecnologias de informação e comunicação no processo de ensino e aprendizagem, particularmente aquelas destinadas a reduzir os atuais índices de evasão verificados na modalidade.

## 1.1 OBJETIVOS

Esta pesquisa será desenvolvida com o propósito de atingir os seguintes objetivos geral e específicos:

### 1.1.1 Objetivo geral

Avaliar se um conjunto de variáveis obtidas a partir da TDT, pode ser usado para prever evasão de estudantes de cursos na modalidade EAD ofertados pela UNIVASF.

### 1.1.2 Objetivos específicos

- Adaptar os modelos preditivos já desenvolvidos para uma outra ferramenta tecnológica;
- Aplicar os classificadores em bases de dados de cursos EAD da UNIVASF;
- Avaliar os resultados dos classificadores segundo métricas consolidadas na literatura.

## 1.2 ORGANIZAÇÃO DO TEXTO

Esse trabalho está organizado em 6 capítulos. No primeiro capítulo apresenta-se o projeto, uma contextualização sobre o problema abordado, assim como os objetivos gerais e específicos.

No segundo capítulo, é realiada uma revisão sobre a TDT, MDE e Aprendizagem Supervisionada, com objetivo de promover um maior detalhamento sobre os conceitos utilizados ao longo do texto. Também nesse capítulo, são apresentados resumos de trabalhos relacionados com esta pesquisa.

O terceiro capítulo explora os detalhes da caracterização da pesquisa e a metodologia aplicada, Descoberta de Conhecimento em Bases de Dados (KDD, do inglês *Knowledge Discovery in Databases*).

O quarto capítulo traz os resultados obtidos com este estudo, as comparações entre as métricas dos algoritmos, as estatísticas descritivas das variáveis obtidas e as visualizações geradas.

O quinto capítulo conclui discute sobre os objetivos alcançados e sugere trabalhos futuros.

# 2 FUNDAMENTAÇÃO TEÓRICA

A necessidade de apoiar o desenvolvimento da EAD tem feito surgir novas teorias, métodos, abordagens de ensino e tratamento das informações, geradas nas diversas tecnologias usadas nessa modalidade. Nas seções seguintes, serão abordadas as principais temáticas envolvidas neste estudo, o que oferecerá as bases teóricas necessárias para a fundamentação da pesquisa.

## 2.1 TEORIA DA DISTÂNCIA TRANSACIONAL

Em 1972, Michael Grahame Moore propôs uma teoria pioneira para a EAD. Essa teoria seria, posteriormente, denominada de Teoria da Distância Transacional (TDT). Ao longo de mais de 40 anos, desde da proposição da TDT, o próprio autor e outros pesquisadores trataram de atualizá-la, principalmente, em razão da evolução tecnológica. Os textos originais do autor estão nas suas obras de 1973, 1993 e 2013 (MOORE, 1973, 1993, 2013).

Em seus estudos, Moore (2013) afirmou que na EAD não existe apenas uma distância física entre professores e alunos, mas, também, uma distância psicológica. Na TDT, as interações do estudante, com o professor, com o conteúdo e com os estudantes, podem ser estudadas com base em construtos elementares, sendo eles, a estrutura dos programas ou cursos, o diálogo entre alunos e professores e o grau de autonomia do discente. De acordo com a TDT a EAD tem a sua própria identidade e características pedagógicas distintivas. Como outras teorias, a TDT pode ser usada no estabelecimento de uma heurística, para a tomada de decisões em projetos de cursos EAD (MOORE, 2013).

Dewey e Bentley (1960) elaboraram o conceito de transação, que, conforme foi exposto posteriormente por Boyd e Apps (1980), denota a interação entre o ambiente, os indivíduos e os padrões de comportamento numa dada situação. Uma transação, em EAD, é a interação entre professores e alunos que estão espacialmente separados. Como foi definido por Moore (2013), essa separação cria padrões especiais de comportamento que afetam tanto o ensino quanto o aprendizado. Derivado da separação, surge um espaço psicológico e comunicacional propício a mal-entendidos nas interações instrutor-aluno. A esta separação é dado o nome de Distância Transacional (DT).

Faz-se necessário lembrar que, segundo Moore (2013) a distância transacional não é um valor fixo ou dicotômico, na verdade, é um valor relativo e contínuo. Além disso, essa distância é diferente para cada estudante, mesmo entre os que compartilham o mesmo curso. Foi apontado por Rumble (1986) que existe uma DT mesmo em cursos presenciais. Com base nisso, pode-se dizer que a EAD é um subconjunto da educação e os estudos

realizados em EAD podem auxiliar a teoria e a prática da educação tradicional. Porém, em uma situação classificada como EAD, a distância entre os participantes — professores e alunos — é grande o suficiente para justificar a investigação de técnicas próprias de ensino-aprendizagem.

Os procedimentos de ensino se dividem em dois grupos, e acontece também um terceiro grupo de variáveis que descreve o comportamento dos alunos. A DT é uma função desses três grupos de variáveis. Na TDT, estes grupos de variáveis recebem o nome de Diálogo, Estrutura e Autonomia do Aluno (MOORE, 2013).

### 2.1.1 Diálogo

O diálogo foi originalmente definido por Moore (2013) como sendo interações focadas, positivas e propositais entre o professor e os alunos. Ainda segundo Moore, o diálogo ocorre entre professores e alunos quando alguém ensina e os demais reagem. O diálogo deve ser direcionado para o aperfeiçoamento da compreensão por parte do aluno.

"A extensão e natureza do diálogo são determinadas pela filosofia educacional da instituição responsável pelo projeto do curso, pelas personalidades do professor e do aluno, pelo tema do curso e por fatores ambientais." (CABAU; COSTA, 2018, p. 438).

Moore (2013) cita meios de comunicação como um importante fator ambiental na EAD, no entanto, relata ser importante que outras variáveis sejam atendidas à medida que a EAD amadurece, as variáveis destacadas por Moore foram: projeto de curso, seleção e treinamento de instrutores e o estilo de aprendizagem dos alunos.

O diálogo é o mediador central da DT e referenciado como medida de aprendizado ao passo que a DT seria uma medida de não-aprendizado. No entanto, já que o diálogo não se limita apenas à interação professor-aluno, especialmente com os avanços da EAD provendo novas formas de interações entre estudantes, diversos pesquisadores vêm propondo a inclusão de interações entre alunos no conceito de diálogo (BENSON; SAMARAWICKREMA, 2009; CHEN; WILLITS, 1999; HUANG *et al.*, 2016).

### 2.1.2 Estrutura do curso

A estrutura do curso diz respeito aos elementos do projeto, bem como, divisão do curso em unidades, objetivos, estratégias institucionais e métodos de avaliação. A estrutura transmite a flexibilidade ou rigidez dos elementos do curso. É, também, responsável pela facilitação ou não-facilitação do diálogo (MOORE, 2013).

Como o diálogo, a estrutura do curso é uma variável qualitativa, e a medida da estrutura em um programa EAD é, normalmente, determinada pela natureza dos meios de comunicação empregados, e também pela filosofia e personalidade dos professores,

pelas personalidades dos alunos e pelas restrições impostas pelas instituições educacionais (MOORE, 2013).

Embora Moore atribua como qualitativa o tipo de variável relacionada ao diálogo e à estrutura, diversos estudos recentes mostraram que é possível quantificar e mensurar esses componentes da TDT (ZHANG, 2003; HORZUM, 2011; PAUL *et al.*, 2015; RAMOS, 2016).

Em cursos gravados em fitas, discos, ou mesmo cursos televisionados a estrutura é rígida e o diálogo não existe, pois não é possível reorganizar o conteúdo para levar em consideração as interações de um aluno. Em contrapartida cursos por teleconferências, permitem ampla variedade de respostas alternativas do instrutor às perguntas dos participantes. Um curso altamente estruturado não possibilita o diálogo professor-aluno, consequentemente, a DT entre alunos e professores aumenta. No entanto, o contrário não pode ser generalizado. "...a extensão do diálogo e a flexibilidade da estrutura variam de programa para programa. É essa variação que dá a um programa maior ou menor distância transacional que outro" (MOORE, 2013).

Em um programa com pequena DT os alunos recebem instruções e orientações por meio do diálogo com o instrutor, nesse caso é possível ter uma estrutura aberta, que dê respaldo para tais interações. Em programas com maior DT é necessário uma estrutura robusta, materiais didáticos que forneçam todas as orientações, instruções e aconselhamentos que o instrutor puder prever, mas sem a possibilidade de alterações por meio de diálogo aluno-professor (MOORE, 2013).

Temos então que, em programas com maior DT, os alunos precisam se responsabilizar em escolher quais atividades e avaliações serão feitas e quando serão feitas. Mesmo que o curso seja bem estruturado, o estudante, na falta de diálogo, decidirá quais atividades serão realizadas, quando, e qual a importância de cada uma. Sendo assim, quanto maior a DT mais é exigido uma autonomia do aluno (MOORE, 2013).

### 2.1.3 Autonomia do aluno

No período do surgimento da TDT, na década de 1970, ela representava a fusão de duas tradições pedagógicas que pareciam contraditórias. Uma, a tradição humanística, que valorizava o diálogo aberto, não-estruturado e interpessoal, tanto na educação quanto no aconselhamento. A outra, a tradição behaviorista, que valorizava o projeto sistemático da instrução, baseado em objetivos comportamentais com o máximo de controle do processo de aprendizagem por parte do professor. No início dos anos 1970, a EAD era dominada pela tradição behaviorista, tanto que, o título do primeiro trabalho sobre a TDT de Moore (1972) foi: "A autonomia do aluno — a segunda dimensão da aprendizagem independente". Nesse trabalho Moore afirmou que: "educadores por correspondência limitavam o potencial do seu método ao negligenciar a habilidade dos alunos em compartilharem a responsabilidade

por seus próprios processos de aprendizagem" (MOORE, 2013).

O termo "autonomia do aluno" foi escolhido para descrever os padrões de comportamento de alunos que usavam materiais didáticos e programas de ensino para atingir seus próprios objetivos, à sua maneira e sob seu próprio controle (MOORE, 2013).

Autonomia do aluno se refere a capacidade de se auto-direcionar. Moore (2013) definiu o estudante autônomo ideal como "a pessoa emocionalmente independente de um professor" e quem "tem capacidade de abordar o assunto estudado diretamente sem a ajuda de um instrutor". Diferente da estrutura do curso e do diálogo, é um fator que depende apenas do aluno. Um aprendiz pouco autônomo pode precisar de um direcionamento maior e uma estrutura mais rígida (HUANG *et al.*, 2016).

## 2.2 EVASÃO DE ALUNOS NA EAD

O censo realizado pela Associação Brasileira de Educação a Distância (ABED), com dados de 2016, consultou 340 instituições em todo o país, formadoras e fornecedoras de de produtos e serviços para EAD (ABED, 2017).

De acordo com o Censo EAD.BR 2016, as taxas de evasão informadas pelos respondentes recaíram, principalmente, entre 11% e 25%. O censo, também, revelou que, entre os respondentes, cursos semipresenciais tem taxa de evasão menor que cursos totalmente a distância. A Tabela 1 compara os índices dos censos realizados pela ABED entre 2014 e 2017 (ABED, 2014; ABED, 2015; ABED, 2016; ABED, 2017).

**Tabela 1** – Taxas de evasão ao longo dos anos segundo o censo realizado pela ABED

| **Taxas de evasão declaradas** | **Percentuais de instituições declarantes, por faixa** | | | |
|---|---|---|---|---|
| | **2013** | **2014** | **2015** | **2016** |
| Até 25% | 65% | 50% | 53% | 58% |
| Entre 26 e 50% | 24% | 38% | 40% | 19% |
| Acima de 50% | 2% | 2% | 7% | 1% |
| Não declararam | 9% | 10% | - | 22% |

**Fonte:** ABED (2014), ABED (2015), ABED (2016), ABED (2017).

Entre os motivos para a evasão investigados e declarados no censo, questões financeiras e falta de tempo foram os citados como os que geram maior evasão. Houve uma parcela considerável de respondentes que acredita que a evasão não é um problema em cursos totalmente a distância, pois os participantes podem sempre retornar.

Em cursos livres, o motivo mais citado foi a falta de tempo, e, também, grande parte dos respondentes acredita que os alunos desses cursos sempre podem retornar.

O Censo EAD.BR 2016 apontou que cursos presenciais, semipresenciais e corporativos possuem mecanismos que vão além do conteúdo e da interação online com professores para manter seus alunos engajados. Já os cursos totalmente a distância e cursos livres não corporativos dependem apenas da experiência do aluno com o conteúdo e com seus professores e tutores.

A Tabela 2 apresenta dados dos indicadores da evasão, em cursos superiores a distância, segundo o Mapa do Ensino Superior no Brasil Edições 2015 e 2016, que foram publicados pelo Sindicato das Empresas Mantenedoras do Ensino Superior (SEMESP) feito com base nos dados do INEP dos anos 2013 e 2014.

**Tabela 2** – Taxas de evasão em cursos superiores presenciais e a distância

| **Ano** | **Cursos presenciais** | | **Cursos a Distância** | |
|---|---|---|---|---|
| | **IES públicas** | **IES privadas** | **IES públicas** | **IES privadas** |
| 2013 | 17,8% | 27,4% | 25,6% | 29,2% |
| 2014 | 18,3% | 27,9% | 26,8% | 32,5% |

**Fonte:** SEMESP (2015), SEMESP (2016)

Segundo o trabalho de Paz e Cazella (2017) a evasão em instituições de ensino superior (IES) é um tema complexo na gestão universitária no Brasil. Um grave problemas das universidades brasileiras é o aumento das taxas de evasão escolar.

Manhães *et al.* (2012) identificaram que a descoberta precoce de grupos de estudantes com risco de evasão é condição importante para reduzir tal problema, pois possibilita proporcionar algum tipo de atendimento personalizado para a situação de cada aluno. Ainda segundo Manhães *et al.* (2012), os processos de identificação desses grupos à época eram manuais e sujeitos a falhas e dependiam, primordialmente, da experiência do docente.

## 2.3 RELAÇÃO ENTRE A DISTÂNCIA TRANSACIONAL E A EVASÃO EM CURSOS A DISTÂNCIA

Pela sua definição, a Distância Transacional é um dos fatores que pode gerar maior dificuldade no engajamento e na comunicação do estudante no ambiente de aprendizagem (GOEL *et al.*, 2012). Além de Moore (2013), outros autores afirmaram que quanto maior for a DT, maior a possibilidade de ocorrência de problemas como atritos, insatisfações e abandono de cursos (ZHANG, 2003; STEINMAN, 2007; HORZUM, 2011; MBWESA, 2014; PAUL *et al.*, 2015).

Zhang (2003), em sua tese de doutorado, demonstrou a existência de uma correlação negativa entre a distância transacional e o envolvimento dos alunos com a sua aprendizagem,

assim como com a sensação de satisfação e a intenção do aluno em persistir no seu curso *on-line*.

Para Steinman (2007), as percepções dos alunos de cursos *on-line* podem ser negativas se eles experimentam grande DT com o instrutor e com outros alunos, podendo ainda influenciar sua decisão de permanecer ou abandonar o curso. Assim, uma vez que a DT afeta a satisfação e retenção dos alunos, esse conceito é visto como um importante tópico de discussão sobre evasão em cursos *on-line*.

A obtenção dos construtos da DT pode refletir uma condição ou um estado de um curso no tempo de sua execução, permitindo, por exemplo, que professores e tutores notem um distanciamento exagerado de determinados alunos e consigam intervir no sentido de prevenir ou reverter situações de evasão de alunos do curso (HORZUM, 2011).

## 2.4 DESCOBERTA DE CONHECIMENTO

De acordo com Costa *et al.* (2012), Mineração de Dados (DM, do inglês, *Data Mining*) pode ser interpretada como uma etapa de um processo mais amplo denominado como Descoberta de Conhecimento em Bases de Dados (KDD, do inglês, *Knowledge Discovery in Databases*). No KDD são identificadas duas grandes etapas: a de pré-processamento de dados, na qual os dados são captados, tratados e organizados, e a de pós-processamento dos resultados obtidos da etapa de mineração (FAYYAD *et al.*, 1996).

Para a obtenção de conhecimentos relevantes, no KDD, é necessário estabelecer metas bem definidas. No estudo de Fayyad *et al.* (1996), as metas são definidas em função do objetivo na utilização da metodologia, sendo dois tipos básicos de metas: verificação e descoberta. No caso de verificação, o sistema está limitado a testar hipóteses definidas pelo usuário, enquanto que em descoberta o sistema encontra novos padrões de forma autônoma. Quando a meta é do tipo descoberta, em geral, o objetivo está relacionado com as seguintes tarefas de mineração de dados: predição e descrição.

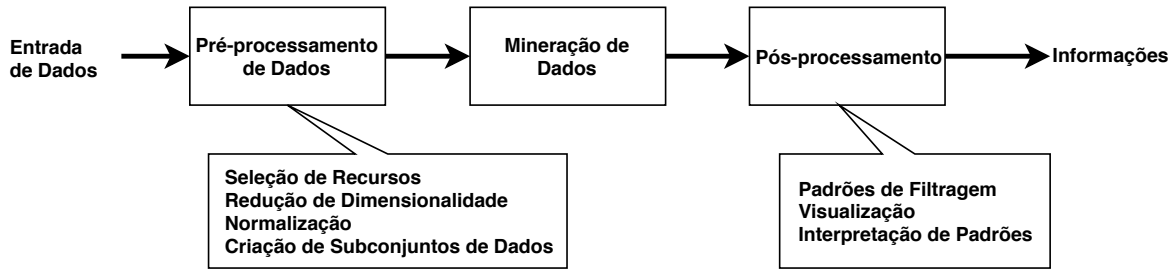
As tarefas preditivas buscam descobrir o valor de um determinado atributo com base nos valores de outros atributos. O atributo a ser predito pode ser chamado de variável preditiva, dependente ou alvo, já os atributos utilizados na predição são chamados de variáveis preditoras, independentes ou explicativas. Sendo generalista, a predição utiliza um conjunto de variáveis para predizer o valor de outras (FAYYAD *et al.*, 1996).

Tarefas descritivas objetivam encontrar padrões — correlações, tendências, grupos, trajetórias e anomalias — que representem os dados (FAYYAD *et al.*, 1996).

Para realizar tarefas de predição e descrição são utilizados alguma das seguintes tarefas e métodos de mineração de dados: classificação, regressão, agrupamento, sumarização, modelagem de dependência e identificação de mudanças e desvios.

Conforme Tan *et al.* (2009), DM é uma parte do KDD, um processo geral de conversão de dados brutos em informações úteis, sendo este composto de uma séries de passos de transformação, do pré-processamentos dos dados até o pós-processamento dos resultados da mineração de dados. A Figura 1 ilustra uma visão geral do KDD segundo Tan *et al.* (2009).

**Figura 1** – Processo de descoberta de conhecimento em bases de dados




**Fonte:** Tan *et al.* (2009).

Ainda de acordo com Tan *et al.* (2009), os dados de entrada podem estar armazenados nos mais diversos formatos (tabelas eletrônicas, bases de dados estruturadas, arquivos simples), e podem estar em um único repositório ou distribuídos por diversas fontes. A etapa de pré-processamento é responsável por transformar os dados brutos em dados apropriados para as análises seguintes. Fusão de dados de múltiplas fontes, limpeza para remoção de ruídos, e seleção de características relevantes à DM, são passos importantes realizados na etapa de pré-processamento. Como existem diversas formas de se coletar e armazenar os dados, o pré-processamento se torna, muitas vezes, a etapa mais demorada e trabalhosa do KDD.

De acordo com Tan *et al.* (2009), o pós-processamento é a etapa do KDD na qual os dados válidos e úteis gerados na etapa de mineração são integrados a ferramentas de auxílio na tomada de decisões. Um exemplo de pós-processamento é a visualização de dados, que permite por meio de gráficos, auxiliar na interpretação de comportamentos e características dos dados. Também podem ser utilizados testes estatísticos para eliminar resultados não legítimos da mineração de dados.

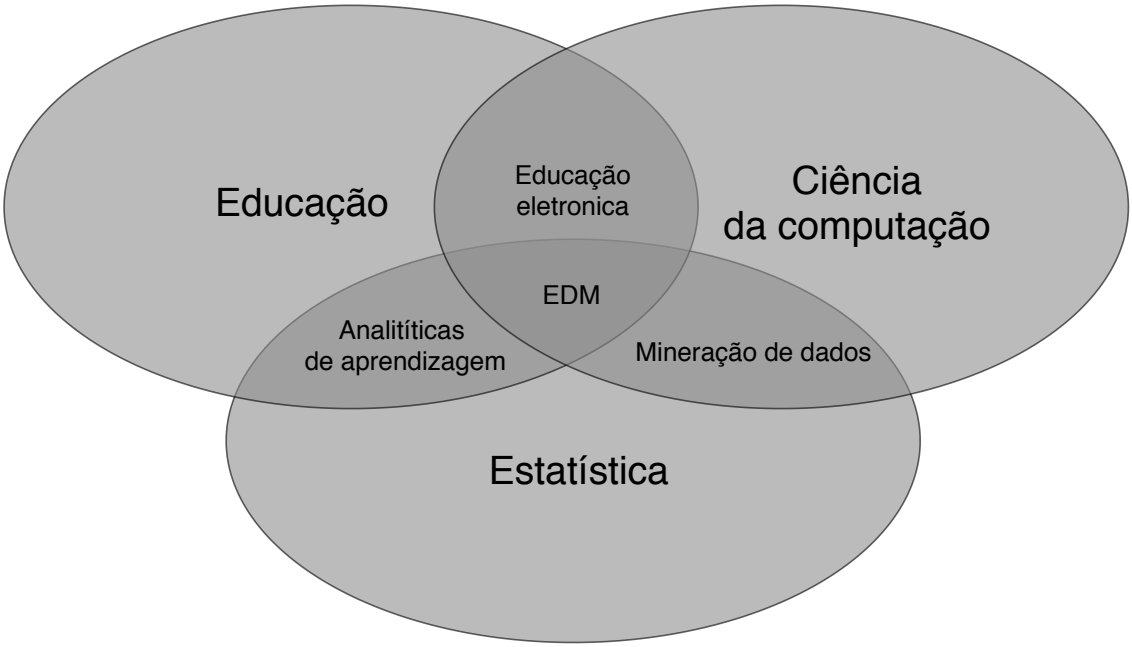
### 2.4.1 Mineração de Dados Educacionais

Segundo Costa *et al.* (2012), a área emergente de Mineração de Dados Educacionais (EDM, do inglês, *Educational Data Mining*) procura desenvolver ou adaptar métodos e algoritmos de mineração existentes, de tal modo que se prestem a compreender melhor os dados em contextos educacionais, produzidos principalmente por estudantes e professores, considerando os ambientes nos quais eles interagem, tais como Ambientes Virtuais de Aprendizagem (AVA), Sistemas Tutores Inteligentes, entre outros.

Muitos métodos utilizados em EDM são, originalmente, da área de mineração de dados. No entanto, no trabalho de Baker *et al.* (2010), muitas vezes estes métodos devem ser modificados, por se fazer necessário considerar a hierarquia da informação. Existe também, falta de independência estatística nos tipos de dados encontrados ao coletar informações em contextos educacionais. Logo, diversos algoritmos e ferramentas utilizadas na área de DM não podem ser aplicados para análise de dados educacionais sem sofrerem os devidos ajustes (BAKER *et al.*, 2011; COSTA *et al.*, 2012).

A EDM pode ser descrita como a combinação de três áreas principais (Figura 2): ciência da computação, educação e estatística. As interseções dessas três áreas forma subáreas próximas da EDM, como sendo analíticas de aprendizagem (LA, do inglês, *Learning Analytics*), ambientes de aprendizado baseados em computador e aprendizado de máquina (ROMERO; VENTURA, 2013).

**Figura 2** – Principais áreas relacionadas com EDM




**Fonte:** Romero e Ventura (2013).

## 2.5 APRENDIZAGEM SUPERVISIONADA

O campo do Aprendizado de Máquina (ML, do inglês, *Machine Learning*) fornece uma ampla área para cientistas explorarem modelos e algoritmos de aprendizado que podem ajudar "máquinas" (computadores) a aprender sobre um sistema com base em dados. Em outras palavras, o objetivo do ML é construir sistemas inteligentes. Algoritmos de aprendizado são ferramentas de reconhecimento de padrão. A seguir é apresentado, de uma forma geral, a descrição de um problema de ML. Suponha que são dados um

conjunto de dados e sua respectiva resposta para um sistema. Então, o problema de ML pode ser definido como ajustar um modelo entre eles, os dados e sua resposta, e como treinar e validar o modelo para aprender as características do sistema por meio dos dados (SUTHAHARAN, 2016).

A tarefa de Aprendizagem Supervisionada é a seguinte:

Dados um conjunto de treinamento de $N$ exemplos de pares entrada/saída

$$(x_1, y_1), (x_2, y_2) \dots (x_n, y_n),$$

onde cada $y_j$ foi gerado por uma função desconhecida $y = f(x)$, descobrir uma função $h$ que aproxime a verdadeira função $f$ (RUSSELL; NORVIG, 2011).

Na definição anterior, $x$ e $y$ podem ser qualquer valor, não necessariamente numérico. A função $h$ é uma hipótese. Aprender é procurar em um espaço de hipóteses possíveis por uma que tenha alto desempenho, mesmo em exemplos não contidos no conjunto de treinamento. Para mensurar a acurácia de uma hipótese se utiliza um conjunto de teste, exemplos que são distintos do conjunto de treinamento. É dito que uma hipótese generaliza bem se prediz corretamente os valores $y$ para exemplos novos (RUSSELL; NORVIG, 2011).

Quando a saída $y$ é uma em um conjunto finito de valores, o problema de aprendizado é denominado classificação, e é chamado classificação booleana ou classificação binária quando existem apenas dois valores possíveis(RUSSELL; NORVIG, 2011).

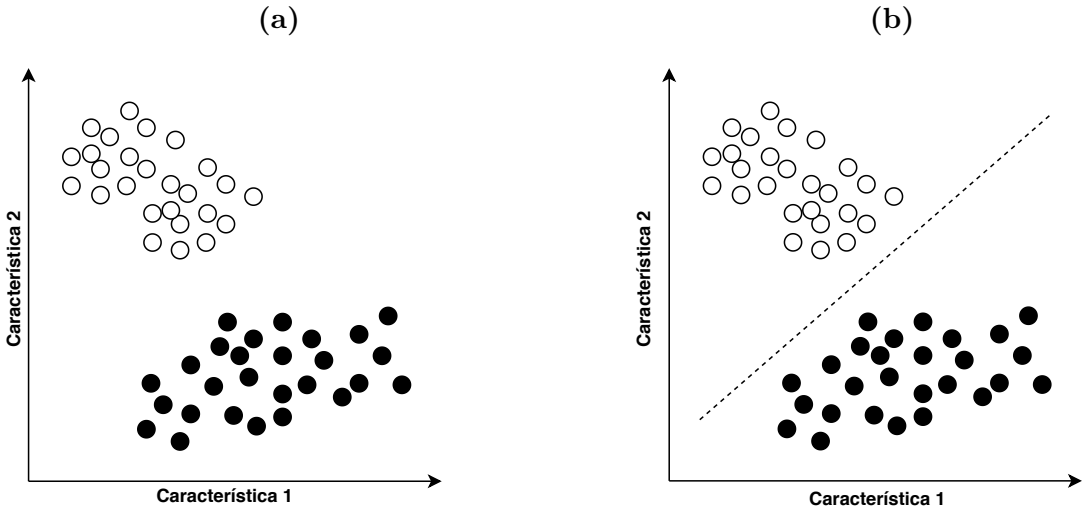
### 2.5.1 Classificação

De acordo com Tan *et al.* (2009) a Classificação é a tarefa de aprender uma função alvo $f$ que mapeie cada conjunto de atributos $x$ para um dos rótulos de classes $y$ pré-determinados. A função alvo também pode ser chamada de modelo de classificação.

Em problemas de classificação, assumimos que são disponibilizados dados etiquetados (classes) para gerar regras que podem ajudar a atribuir uma etiqueta a novos dados que não possuem classes. Nesse caso, podemos derivar uma regra exata pela disponibilidade das classes. A Figura 3 ilustra um exemplo com duas classes, etiquetadas com pontos brancos e pontos pretos, e uma reta (Figura 3b) representado a regra que nos ajuda a estabelecer uma classe para cada novo ponto (SUTHAHARAN, 2016).

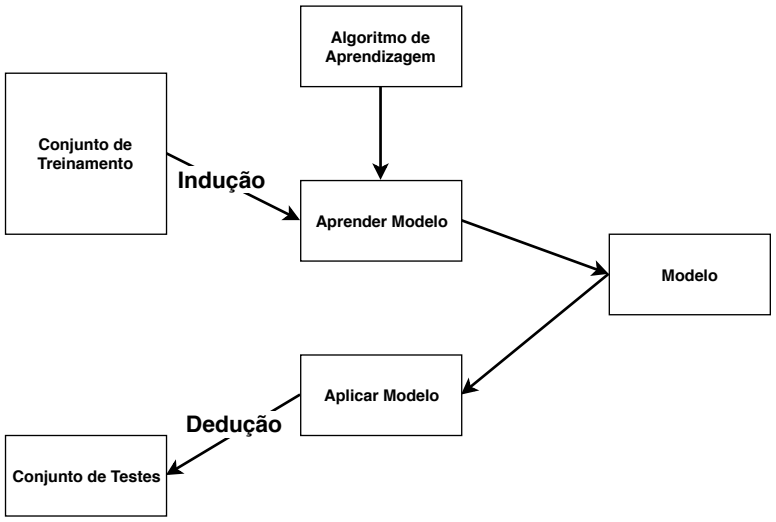
Em uma abordagem geral, para a construção de um modelo de classificação, primeiro, um conjunto de treinamento consistindo de registros com rótulos de classe conhecido é fornecido. O conjunto de treinamento é usado para construir um modelo de classificação, que é então aplicado a um conjunto de testes, constituído por registros com rótulos desconhecidos para o modelo. A Figura 4 ilustra essa abordagem geral (TAN *et al.*, 2009).

**Figura 3** – Exemplo de classificação




**Fonte:** Suthaharan (2016).

**Figura 4** – Abordagem geral para a construção de um modelo de classificação




**Fonte:** Tan *et al.* (2009).

Conforme Tan *et al.* (2009) a avaliação do desempenho de um modelo de classificação é baseada na contagem de registros do conjunto de teste que foram classificados correta e incorretamente. Estas contagens são organizadas em uma tabela denominada matriz de confusão. O Quadro 1 apresenta uma matriz de confusão para um problema de classificação binária. A partir das entradas da matriz de confusão, o número de previsões corretas realizadas pelo modelo é $(f_{11} + f_{00})$ e o número de previsões incorretas é $(f_{10} + f_{01})$.

A matriz de confusão mostra informações importantes para determinar o desem-

**Quadro 1** – Exemplo de matriz de confusão

| | | **Classe prevista** | |
|---|---|---|---|
| | | Classe = 1 | Classe = 0 |
| **Classe real** | Classe = 1 | $f_{11}$ | $f_{10}$ |
| | Classe = 0 | $f_{01}$ | $f_{00}$ |

**Fonte:** Elaborado pelo autor.

penho do modelo, no entanto, resumir essas informações em um único número é mais conveniente quando queremos comparar o desempenho entre diferentes modelos. Isso pode ser feito usando uma métrica de desempenho como a acurácia, que é definida da seguinte maneira (TAN *et al.*, 2009):

$$\text{Acurácia} = \frac{\text{Número de previsões corretas}}{\text{Número total de previsões}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

A acurácia representa a taxa de acertos do algoritmo, e é, geralmente, a primeira a ser analisada para medir performance.

Neste trabalho, além da acurácia, também serão avaliadas as seguintes métricas:

$$\text{Precisão} = \frac{f_{11}}{f_{11} + f_{01}},$$

que representa a preditividade positiva, que é o percentual de acertos de verdadeiros positivos dentre todos os exemplos classificados como positivos,

$$\text{Sensibilidade} = \frac{f_{11}}{f_{11} + f_{00}},$$

que indica a taxa de verdadeiros positivos, ou seja, o percentual de verdadeiros positivos previstos corretamente pelo classificador,

$$\text{Especificidade} = \frac{f_{10}}{f_{11} + f_{01}},$$

que fornece a taxa de verdadeiros negativos, ou seja, o percentual de instâncias previstos corretamente como verdadeiros negativos,

$$\text{Taxa de Falsos Positivos (TFP)} = \frac{f_{01}}{f_{10} + f_{01}},$$

que indica o percentual de instâncias negativas previstas incorretamente como verdadeiros positivos e a

$$\text{Taxa de Falsos Negativos (TFN)} = \frac{f_{00}}{f_{00} + f_{11}},$$

que indica o percentual de instâncias positivas previstas incorretamente como verdadeiros negativos.

#### 2.5.1.1 Árvore de decisão

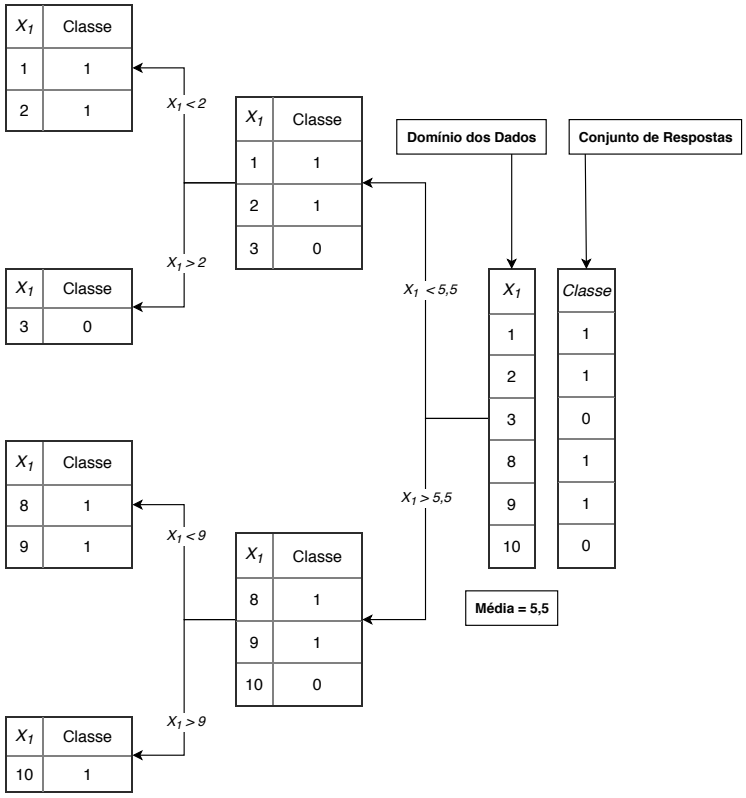
Em ML, existem dois tipos de árvores de decisão: árvores de regressão e árvores de classificação. Uma árvore de decisão utiliza uma abordagem baseada em regras para dividir o domínio dos dados em múltiplos espaços lineares e prever respostas. Se as respostas previstas forem contínuas, então a árvore de decisão é chamada de árvore de regressão, e se as previsões são discretas, ou seja, pertencem a uma classe, então a árvore de decisão é chamada de árvore de classificação (SUTHAHARAN, 2016).

De acordo com Suthaharan (2016), árvores de decisão são modelos de aprendizado supervisionado, que mapeiam o domínio dos dados hierarquicamente em um conjunto de respostas. Dividindo o domínio dos dados (também chamado de nó), recursivamente, em dois subdomínios, de forma que os subdomínios tenham um maior ganho de informação que o nó que foi dividido. Já que o objetivo do aprendizado supervisionado é a classificação dos dados, portanto, o aumento do ganho de informação influencia na eficiência da classificação nos subdomínios criados pela divisão. Encontrar a divisão que traga o máximo de ganho de informação, ou seja, eficiência na classificação é o objetivo do dos algoritmos de otimização no aprendizado supervisionado baseado em árvores de decisão. Na Figura 5 vemos um exemplo de árvore de decisão em termos de divisão de domínios focado no ganho de informação.

Suthaharan (2016) traz um exemplo de classificação usando árvore de decisão. Suponha que temos um sistema que produz eventos (observações) que podem pertencer a uma de duas classes, 0 ou 1, e estes eventos dependem apenas de uma variável. Consequentemente, definimos o domínio como: $D = \{e_1, e_2, \ldots, e_n\}$ (assumimos que isto é um conjunto ordenado), e seus rótulos de classe correspondentes $L = r_1, r_2, r_3, \ldots, r_n$, onde $r_i$ pertence $\{0, 1\}$, e $i = 1 \ldots n$. A propagação dos rótulos das classes sobre o domínio dos dados determina a facilidade na classificação. Representamos o ganho de informação do domínio $D$ em relação a $L$ por $I_i$ e dividimos o conjunto ordenado na localização $m$ para formar dois subdomínios $D_1 = \{e_1, e_2, \ldots, e_m\}$ e $D_2 = \{e_{m+1}, e_{m+2}, \ldots, e_n\}$ com os conjuntos de respostas correspondentes $L_1 = \{r_1, r_2, \ldots, r_m\}$ e $L_2 = \{r_{m+1}, r_{m+2}, \ldots, r_n\}$. Se os respectivos ganhos de informação são $I_{i1}$ e $I_{i2}$, então $m$ será considerado a melhor divisão se a média$(I_{i1}, I_{i2}) > I_i$. Não obstante, precisamos de uma boa medida quantitativa para mensurar o ganho de informação obtido após a divisão dos dados.

Vamos supor que $p_0$ e $p_1$ representem as probabilidades de que as classes 0 e 1 possam ser extraídas do domínio $D$, respectivamente. Se, por exemplo, $|p_0 - p_1| \to 1$; então podemos observar que uma classe em particular tem grande predominância neste domínio, portanto, não é mais necessário dividir os dados. Similarmente, se $|p_0 - p_1| \to 0$, então as classes tem predominância igual no domínio; logo, uma divisão é necessária. Neste caso geramos dois subdomínios $D_1$ e $D_2$. Digamos que, $q_0$ e $q_1$ são as probabilidades de que a classe 0 e a classe 1 sejam derivadas do subdomínio $D_1$, respectivamente. Se a divisão for

**Figura 5** – Exemplo de árvore de decisão usada para classificação construída com um domínio de dados uni-dimensional




**Fonte:** Suthaharan (2016).

eficiente, $q_0 > p_0$ ou $q_1 > p_1$. Assumindo $q_0 > p_0$, então $q_0 = p_0 + \epsilon$, onde $\epsilon > 0$.

$$|q_0 - q_1| = |2q_0 - 1| = |2(p_0 + \epsilon) - 1| = |2p_0 + 2\epsilon - 1|$$

$$|q_0 - q_1| = |p_0 + 1 - p_1 + 2\epsilon - 1| = |p_0 - p_1 + 2\epsilon|$$

Esta equação enfatiza a seguinte inequação, (quando $q_0 > p_0$):

$$|q_0 - q_1| > |p_0 - p_1|$$

As diferenças absolutas na inequação acima são as medidas quantitativas de proporcionalidade entre as classes em seus respectivos subdomínios. Essas medida probabilística é uma boa métrica para abordagem de otimização de árvores de decisão.

#### 2.5.1.2 K-ésimo vizinho mais próximo

Fix e Hodges (1951) introduziram um método não paramétrico de reconhecimento de padrões que ficou conhecido como Regra do K-ésimo Vizinho Mais Próximo (KNN, do inglês, *K-nearest-neighbor*). O KNN é um dos algoritmos de classificação mais simples e

mais fundamentais e deveria ser a primeira escolha para um estudo de classificação quando se tem pouco ou nenhum conhecimento sobre a distribuição dos dados. A classificação com KNN foi desenvolvida a partir da necessidade de realizar análises discriminatórias quando estimativas confiáveis de densidade de probabilidade dos dados não são conhecidas ou difíceis de determinar.

Cover *et al.* (1967) descreveram as propriedades formais do KNN, por exemplo, foi demonstrado que para $k = 1$ e $n \to \infty$ o erro de classificação do KNN é limitado pelo dobro da taxa de erro de Bayes. Desde que essas propriedades formais foram estabelecidas, seguiu-se uma longa linha de investigações, incluindo uma abordagem de rejeição, melhoramento em relação com a taxa de erro de Bayes, abordagem com pesos nas distâncias (PETERSON, 2009).

Segundo Tan *et al.* (2009) um classificador que utiliza KNN representa cada exemplo, de treinamento ou de teste, como um ponto de dado em um espaço $d$-dimensional, onde $d$ é a quantidade de atributos. Dado um exemplo de teste, calcula-se a sua proximidade com o resto dos pontos de dados do conjunto de treinamento, usando alguma medida de distância, geralmente a distância euclidiana. Os $k$ vizinhos mais próximos de um determinado ponto de teste $z$ se referem aos $k$ pontos com menor distância de $z$. Então, $z$ é classificado com base nos rótulos de classe do seus vizinhos mais próximos e lhe é atribuída a classe majoritária dos seus vizinhos mais próximos. No exemplo da Figura 6, no qual o símbolo **-** representa a classe negativo, o símbolo $+$ representa a classe positivo e $\lambda$ representa um ponto dado a ser classificado. Na Figura 6a, onde $k = 1$, foi atribuído ao ponto dado a classe negativo. Na Figura 6b, com $k = 2$, os dois vizinhos mais próximos do ponto dado tem classes distintas, portanto, podemos atribuir aleatoriamente qualquer uma das duas classes. Na Figura 6c, com $k = 3$, dois dos vizinhos mais próximos do ponto dado são da classe positivo e apenas um é da classe negativo, logo, atribuímos ao ponto dado a classe positivo.

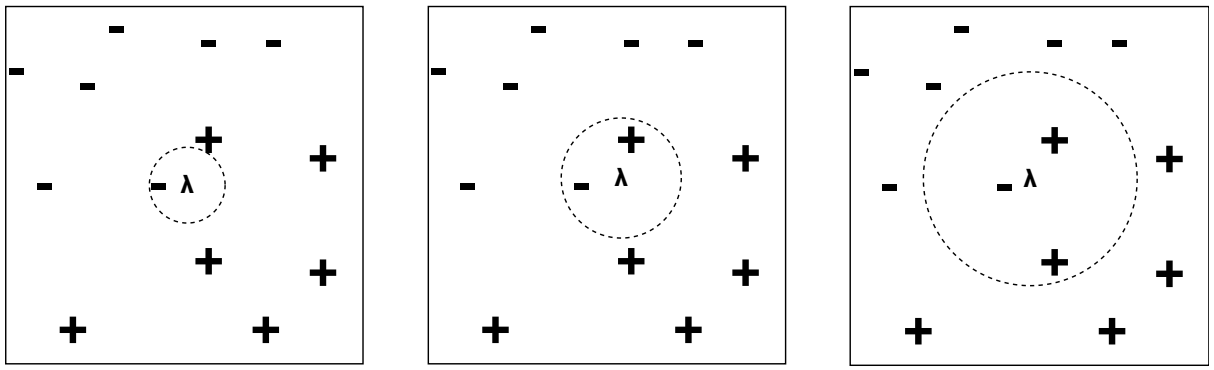
O desempenho de um classificador usando KNN pode ser melhorado quando os atributos são transformados antes da análise de classificação. A forma mais comum de transformação é a normalização ou padronização. A normalização remove efeitos provocados por atributos com escalas diferentes, como, o atributo peso de um paciente que pode ser baseado na unidade quilograma enquanto os valores de proteína no sangue são baseados em nanograma por decilitro variando entre $-3$ e 3, logo, o peso do paciente teria maior influência no cálculo das distâncias entre os pontos de exemplo e, por consequência, na classificação (PETERSON, 2009).

#### 2.5.1.3 Regressão logística

A Regressão Logística (RL) é uma generalização da regressão linear. É usada, principalmente, para prever variáveis dependentes binárias ou de múltiplas classes. Como

**Figura 6** – Os 1, 2 e 3 vizinhos mais próximos de um ponto dado

**(a)** 1-vizinho mais próximo **(b)** 2-vizinhos mais próximo **(c)** 3-vizinhos mais próximo



**Fonte:** Tan *et al.* (2009).

a variável de resposta é discreta, ela não pode ser modelada diretamente por regressão linear. Portanto, em vez de prever uma estimativa de ponto do evento em si, o modelo baseia-se para prever a probabilidade de sua ocorrência (ŞEN *et al.*, 2012).

O modelo de RL surge do desejo de modelar as probabilidades posteriores de $K$ classes através de funções lineares em $x$, ao mesmo tempo, garantir que a soma dessas probabilidades seja um (1) e elas permaneçam no intervalo entre 0 e 1 (JAMES *et al.*, 2013).

Uma vantagem da RL no processo de classificação de uma variável dependente binária (binomial), é que, nela, pode ser usado um conjunto de variáveis independentes numéricas ou categóricas (KLEINBAUM; KLEIN, 2002).

Dada uma variável ou conjunto de variáveis $X$, podemos utilizar um modelo de RL para calcular a probabilidade de pertencimento à classe $y$. Para cada valor ou valores de $X$ pode ser feita uma previsão para a classe $y$. Por exemplo, pode-se dizer que o item sendo testado pertence a classe $y$ sempre que o modelo RL retornar uma probabilidade maior que 50%, sendo que este limiar pode ser ajustado de acordo com a necessidade do problema abordado (JAMES *et al.*, 2013).

Os modelos de RL para diversas variáveis é descrito pela seguinte fórmula:

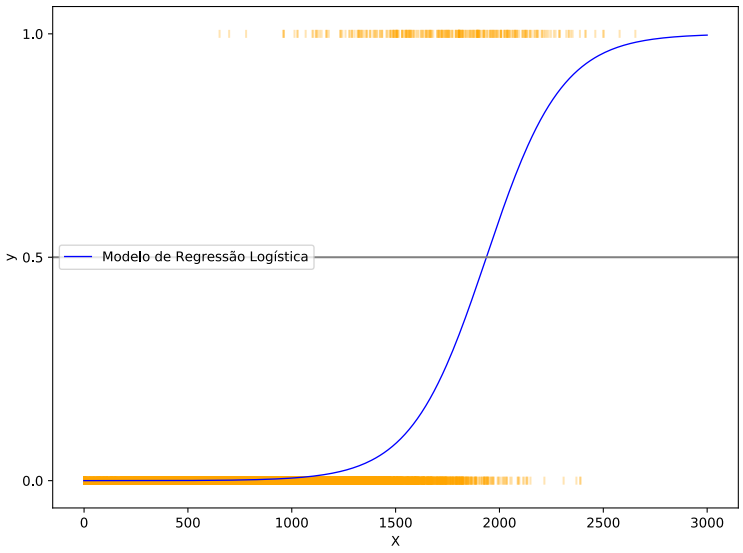
$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_{1,i} + \ldots + \beta_k X_{k,i}$$

Onde, $\beta_0$, $\beta_1$, ..., $\beta_k$ são os coeficientes das variáveis que explicam a ocorrência de determinado evento e $p_i$ é a probabilidade de um evento ocorrer dado o conjunto de variáveis $i$.

O resultado do modelo *logit* é uma curva em forma de S (Figura 7). Para estimar

um modelo de regressão logística, essa curva de valores previstos é ajustada aos dados reais, analogamente, como é feito com uma relação linear em regressão múltipla.

**Figura 7** – Previsões utilizando regressão logística. As probabilidades se encontram no intervalo entre 0 e 1




**Fonte:** James *et al.* (2013).

Em níveis muito baixos da variável independente, a probabilidade se aproxima de 0%, mas nunca alcança tal valor. Da mesma forma, quando o valor da variável independente aumenta, os valores previstos crescem para acima da curva, mas, em seguida, a inclinação começa a diminuir, aproximando a probabilidade de 100%, sem, entretanto, exceder esse valor (HAIR *et al.*, 2009).

## 2.6 TRABALHOS RELACIONADOS

Ramos *et al.* (2016), propuseram o mapeamento do comportamento de usuários de um *Learning Management System* (LMS), em variáveis que representam os contrutos da TDT. O objetivo foi descrever e validar um conjunto de variáveis com as quais esses construtos podem ser medidos, permitindo o desenvolvimento de pesquisas na área, assim como a obtenção destas medidas a qualquer momento do curso e sem a necessidade de questionários. A criação e validação de um conjunto final de variáveis foi feita a partir da Análise Fatorial Confirmatória (CFA), que apontou como cada construto pode ser representado por um conjunto de atributos obtidos a partir do banco de dados do LMS.

Ramos *et al.* (2018), analisaram a performance de diferentes algoritmos na previsão da evasão em alunos EAD. No trabalho foram utilizados dados de turmas de dois cursos de graduação na Universidade de Pernambuco (UPE). Os algoritmos testados foram: Árvore

de Decisão, Máquina de Vetor de Suporte (SVM), Rede Neural Artificial, K-Vizinhos Mais Próximos (KNN) e Regressão Logística. As variáveis foram construídas com base na TDT. O algoritmo com maior acurácia foi o KNN, com maior precisão foi o SVM e a Regressão Logística teve os maiores valores de Recall e Área Sob a Curva ROC (AUC).

Queiroga *et al.* (2018), elaboraram um modelo de predição da evasão de estudantes em cursos técnicos a distância, por meio de mineração de dados, utilizando dados de turmas EAD do Instituto Federal Sul-rio-grandense. Os algoritmos utilizados para gerar os modelos testados foram: *Bayes Net*, *Simple Logistic*, *Multilayer Perceptron*, *Random Forest* e J48, implementados na biblioteca WEKA. Todos os algoritmos selecionados previram com exatidão de 95% a evasão de um aluno antes do final do primeiro ano. O algoritmo que mais se destacou no quesito acurácia foi o Random Forest, com 85%.

Manhães *et al.* (2011), utilizaram mineração de dados para identificar antecipadamente alunos com risco de evasão. Foram utilizados dados de cursos de graduação da Universidade Federal do Rio de Janeiro (UFRJ). Os resultados mostraram que utilizando as primeiras notas semestrais dos calouros é possível identificar com precisão de 80% a situação final do aluno no curso.

Paz e Cazella (2017), alicaram KDD em dados coletados em uma instituição de esnino superior (IES), e, através da tarefa de classificação, utilizando a técnica de árvores de decisão, atingiram acurácia de 90% na identificação de alunos evasores.

Ramos (2016) desenvolveu e testou modelos preditivos com base nos algoritmos Árvore de Decisão, SVM, Rede Neural Artificial, KNN e Regressão Logística, usando com base as variáveis representativas dos construtos da distância transacional, obtidas em trabalho anterior (RAMOS *et al.*, 2016). Esse trabalho serviu como principal referência para o desenvolvimento deste estudo, a fim de verificar a aplicabilidade do método em um outro cenário educacional.

## 2.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram apresentados os principais conceitos da EAD, dados demográficos sobre a evasão, os métodos e algoritmos de ML que foram utilizados neste trabalho, e os trabalhos relacionados na literatura.

No próximo capítulo é detalhado o método KDD e como foi aplicado neste estudo.

# 3 PROCEDIMENTOS METODOLÓGICOS

O percurso metodológico desenvolvido neste trabalho é descrito nas seções seguintes deste capítulo. Como principal método, adotou-se um modelo clássico de descoberta de conhecimento em bases de dados, a partir do qual todas as as etapas da pesquisa foram deenvolvidas.

## 3.1 CARACTERIZAÇÃO DA PESQUISA

Segundo Marconi e Lakatos (2003), a pesquisa é um procedimento formal, com método de pensamento reflexivo, que requer um tratamento científico e que se constitui no caminho para conhecer a realidade ou para descobrir verdades parciais. A pesquisa é um procedimento sistemático e crítico, que permite descobrir novos fatos, relações ou leis acerca de qualquer campo do conhecimento.

Uma pesquisa pode ser caracterizada segundo os seguintes critérios (GIL, 2008):

a) Quanto à natureza: básica ou aplicada;

b) Quanto aos objetivos: exploratória, descritiva ou explicativa;

c) Quanto à abordagem: qualitativa ou quantitativa;

d) Quanto aos procedimentos: documental, bibliográfica, experimental, levantamento, estudo de caso, entre outros.

Este trabalho pode ser classificado como de natureza aplicada, já que será adotada uma metodologia para busca de conhecimentos em bancos de dados e métodos de classificação para prever a evasão de cursos EAD.

Em relação aos objetivos, podemos classificar este trabalho como pesquisa exploratória e descritiva. Tendo como base Gil (2002), a pesquisa exploratória busca ampliar o conhecimento sobre o problema, procurando torná-lo mais explícito ou a construção de hipóteses, tendo como objetivo central o aperfeiçoamento de ideias ou a revelação de intuições. E a pesquisa descritiva objetiva descrever características de determinado fenômeno ou população. Este trabalho utiliza uma metodologia de exploração de conhecimento para tentar prever um comportamento em um conjunto de uma população.

Quanto à abordagem, este trabalho é classificado como quantitativo, em razão da utilização de algorítmos de DM, a partir dos quais serão extraídas as características dos estudantes de EAD e aplicados algoritmos de classificação que farão a devida categorização.

No quesito procedimentos, classificamos este trabalho como pesquisa experimental. De acordo com Gil (2002), a pesquisa experimental consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto.

No caso deste trabalho, o objeto de estudo é a evasão na EAD da UNIVASF e as variáveis foram definidas com base na TDT.

## 3.2 MÉTODO

Para tratamento e preparação dos dados para os diferentes algoritmos de classificação que serão avaliados, foi utilizado KDD como descrito por Tan *et al.* (2009) e ilustrado na Figura 8.

**Figura 8** – Fluxo básico do processo KDD.


**Fonte:** Tan *et al.* (2009).

As subseções seguintes descrevem como o processo KDD foi aplicado neste trabalho.

### 3.2.1 Entrada de Dados

A fase de entrada de dados foi desenvolvida, baseando-se no trabalho de Ramos (2016), coletando as variáveis mais relevantes que poderiam representar cada um dos três construtos da TDT. Os dados das quais foram retiradas as variáveis estão armazenados nas bases de dados do Sistema de Gestão de Aprendizado Moodle[1], atualmente, em uso pelos cursos de graduação oferecidos na modalidade EAD pela UNIVASF. Os dados foram cedidos pela Secretaria de Educaçao a Distância (SEAD) da UNIVASF, por meio da Secretaria de Tecnologia da Informação (STI), responsável pelo suporte e manutenção das plataformas de EAD na instituição. Os dados foram então armazenados em um computador pessoal utilizando o Sistema de Gerenciamento de Banco de Dados (SGBD) MySQL.

O MySQL[2] é o SGBD mais popular no mundo. Provê performance, confiabilidade e facilidade de uso, MySQL vem liderando a escolha de aplicações *web*, usado por grandes empresas na internet como: Facebook, Twitter, YouTube, Yahoo! e muitas outras.

O MySQL utiliza Linguagem de Busca Estruturada (SQL, do inglês, *Structured Query Language*). Entre as suas vantagens podemos listar: portabilidade, compatibilidade,

[1] <https://moodle.org/> Acesso em: 06 de mar. 2019
[2] <https://www.mysql.com/> Acesso em: 06 de mar. 2019

excelente desempenho e estabilidade, facilidade de manuseio e é um *software* livre sob a licença GPL.

O Moodle é uma plataforma de ensino projetada para oferecer a educadores, administradores e estudantes, com uma sistema integrado, simple e robusto, a criação de ambientes de aprendizado personalizados. É apoiado por uma rede de mais de 80 empresas ao redor do mundo.

O banco de dados Moodle é grande e complexo, retendo informações sobre os diversos componentes de uma sala de aula virtual como *chats*, questionários *online* e fóruns de discussão, além de manter um registro de todas as ações do usuário nos seus componentes.

A depender da versão do Moodle, a quantidade de tabelas na base de dados pode variar significativamente. A versão utilizada neste trabalho possuía cerca de 430 tabelas. O Quadro 2 apresenta as tabelas essenciais para a coleta dos dados utilizados neste trabalho.

**Quadro 2** – Descrição das principais tabelas do BD Moodle, onde foram coletados dados desse trabalho.

| Tabela | Descrição |
|---|---|
| *mdl_assign* | Guarda informações sobre as atividades avaliativas relacionadas com a produção de material pelos alunos em cada disciplina. |
| *mdl_context* | Registra os níveis (contextos) de acesso de cada usuário, de acordo com o seu perfil. |
| *mdl_course* | Tabela principal dos cursos, onde as disciplinas de cada curso são registradas e configuradas. |
| *mdl_course_categories* | Tabela auxiliar da *mdl_course*, onde são criadas as categorias que podem representar cursos distintos (Biologia, Pedagogia entre outros) |
| *mdl_forum* | Possui informações gerais de cada fórum criado nas disciplinas. |
| *mdl_forum_discussions* | Registra os tópicos criados em cada um dos fóruns. |
| *mdl_forum_posts* | Guarda as postagens dos alunos que são associadas aos respectivos fóruns/tópicos. |
| *mdl_log* | Registra todas as ações dos usuários no ambiente. É a tabela com maior número de registros. |
| *mdl_message_read* | Armazena as mensagens que foram lidas pelos destinatários, assim como o emissor e o receptor. |
| *mdl_role_assignments* | Registros da atribuição de funções do usuário em contextos diferentes. |
| *mdl_user* | Cadastro geral de usuários. |

**Fonte:** Elaborado pelo autor.

Foram selecionados dois cursos dos quais foram extraídos os dados. O curso de

Bacharelado em Administração Pública com início no período letivo 2013.2 e termino no período letivo 2017.1, contando com 285 estudantes e 41 disciplinas. E o curso de Licenciatura em Pedagogia que ocorreu entre os períodos letivos de 2014.2 e 2018.1, com 160 estudantes e 39 disciplinas. A Tabela 3 apresenta um resumo dessas informações.

**Tabela 3** – Resumo das informações dos cursos selecionados

| Curso | Alunos | Disciplinas | Período Inicial | Período Final |
|---|---|---|---|---|
| Bacharelado em Administração Pública | 285 | 41 | 2013.2 | 2017.1 |
| Licenciatura em Pedagogia | 160 | 39 | 2014.2 | 2018.1 |

**Fonte:** Elaborado pelo autor.

### 3.2.2 Pré-processamento

Devido ao grande volume de dados foram elaborados *scripts* em SQL, que geram tabelas auxiliares apenas com dados dos das disciplinas e alunos matrículados nos cursos selecionados para este trabalho. Essas tabelas serão apresentadas no capítulo de resultados.

As variáveis utilizadas neste trabalho foram baseadas na pesquisa de Ramos (2016), que também extraiu dados de uma instância do Moodle. No entanto, em vez de utilizar todas as variáveis, foram utilizadas apenas as resultantes da etapa de seleção de variáveis realizada por Ramos em sua tese de doutorado. Essas variáveis também serão descritas no capítulo de resultados.

Foram elaborados *scripts* SQL que, a partir das tabelas mencionadas nos Quadros 2 e 3, foram construídas tabelas com os dados brutos, nas quais cada linha gerada representa um aluno em uma disciplina e as variáveis mapeadas para os construtos da TDT. As tabelas geradas serviram como base para as etapas seguintes do processo KDD.

Para que as variáveis que dependiam de períodos de tempo fossem coletadas corretamente, foi necessário inserir a data final de disciplinas que não possuíam esse campo em suas linhas da tabela *mdl_course*. Nesses casos foi inserida a data de fechamento do semestre. A data de início de algumas disciplinas também teve que ser corrigida, pois, elas foram criadas nos primeiros períodos, mas ocorreram posteriormente.

Foram necessárias várias iterações de ajustes nos *scripts* e conferência dos resultados antes de avançar para a próxima fase, como já havia sido previsto, esta foi a etapa que consumiu a maior parcela do tempo de elaboração desse trabalho. A conferência dos dados se deu a partir de acessos com perfil de professor no Moodle, com acesso a várias disciplinas, nas quais eram feitas as contagem de eventos realcionados às variáveis coletadas.

As tabelas geradas foram convertidas para planilhas eletrônicas, para facilitar o processo de filtragem e remoção de erros de implementação ou de configurações feitas pelos gestores dos cursos. Foram removidos professores cadastrados como alunos, disciplinas ofertadas para alunos repetentes e disciplinas ofertadas fora do período (já que essas disciplinas poderiam enviesar os algoritmos de classificação). Também, foram eliminadas colunas que poderiam ser utilizadas para identificar os alunos, com o objetivo de anonimizar os dados.

Disciplinas como Estágio Supervisionado, Trabalho de Conclusão de Curso, entre outras, também, foram eliminadas, pois não seguem a mesma estrutura de disciplinas tradicionais, podendo causar vieses nos algoritmos.

Todos os *scripts* gerados nessa etapa estão disponíveis no Apêndice A

### 3.2.3 Mineração de dados

A análise exploratória dos dados foi realizada utilizando a linguagem de programação Python, na distribuição Anaconda.

Python[3] é uma linguagem de programação de código aberto classificada como linguagem de alto nível de abstração. Considerada de fácil manuseio mesmo por usuários iniciantes. É mantida e desenvolvida pela Python *Software Foundation*.

Graças a sua enorme comunidade, existem diversos pacotes e bibliotecas desenvolvidas em Python para as mais variadas tarefas, desde servidores HTTP, desenvolvimento de aplicações desktop até mineração de dados, inteligência artificial e estatística.

A distribuição de código aberto Anaconda[4] é uma maneira fácil de realizar tarefas de mineração de dados e aprendizado de máquina em ambientes Linux, Windows ou Mac OS X. Anaconda é um gerenciador de pacotes e ambientes e uma distribuição Python especializada em data science com mais de 1500 pacotes de código aberto.

O ambiente de desenvolvimento selecionado foi o Jupyter Notebook, [5] que é uma aplicação *web* de código aberto que permite a criação e compartilhamento de documentos que contém código em tempo de execução, equações, visualizações e textos narrativos. Funciona como uma IDE (do inglês, *Integrated Development Environment*) e foi desenvolvido para tarefas de limpeza e transformação de dados, simulações numéricas, modelagem estatística, visualização de dados, aprendizado de máquina e mais.

Jupyter Notebook suporta mais de 40 linguagens de programação incluindo Python e já vem pré configurado na distribuição Anaconda.

[3] <https://www.python.org/> Acesso em: 06 de mar. 2019
[4] <https://www.anaconda.com/> Acesso em: 06 de mar. 2019
[5] <https://jupyter.org/> Acesso em: 06 de mar. 2019

As planilhas foram carregadas no ambiente utilizando a Python Data Analysis Library (pandas) em estruturas de dados denominadas *dataframes*.

A Python Data Analysis Library[6], ou simplesmente pandas, é uma biblioteca de código aberto sob a licença BSD que provê estruturas de dados e ferramentas de análise de dados de alta performance e fácil uso para a linguagem de programação Python. Pandas proporciona estruturas de dados rápidas, flexíveis e expressivas desenvolvidas para uso com dados relacionais ou etiquetados.

Foram analisadas as variáveis dos alunos do último período, após essa analise, percebeu-se que nenhum aluno que estava listado havia evadido. Todas as ocorrências desses alunos foram marcadas como não evadidos.

Percebeu-se que os administradores dos cursos removeram os alunos que evadiram do quarto para o quinto semestre, então, foi necessário recolocar esses alunos nas disciplinas que aconteceram após o quarto período. Todos os alunos que tiveram de ser adicionados novamente foram marcados como evadidos.

Os *dataframes* de cada curso foram divididos em dois conjuntos de dados, conjunto de teste e conjunto de treino. Para tal, foi utilizada a função *train_test_split* da bilbioteca pandas. Essa função garante que os dados sejam divididos de forma aleatória, o que mantém a mesma distribuição de variáveis entre os conjuntos.

Cada par de conjunto de dados passou pelas seguintes etapas para cada algoritmo de classificação.

1 - Treinamento: utilizando o conjunto de treinamento e por meio das funções e classes disponibilizados pela biblioteca pandas, os algoritmos de classificação são treinados ou "aprendem" os padrões dos dados.

Os algoritmos de classificação selecionados (KNN, Árvore de Decisão e Regressão Logística) são disponibilizados pela biblioteca de ML em Python Scikit-learn.

Scikit-learn[7] é um módulo Python para aprendizado de máquina de código aberto sob a licença BSD. Além das principais tarefas de mineração, como: classificação, regressão e clusterização a biblioteca proporciona as visualizações mais básicas para análise exploratória.

2 - Avaliação: utilizando o modelo resultante da etapa de treinamento e o conjunto de teste avaliamos o classificador segundo as seguintes métricas: acurácia, precisão, sensibilidade, especificidade.

3 - Importância de variáveis: de posse do modelo treinado, avaliamos quais variáveis tiveram mais influência na performance do modelo por meio da classe *SelectFromModel* da

[6] <https://pandas.pydata.org/> Acesso em: 06 de mar. 2019
[7] <https://scikit-learn.org/> Acesso em: 06 de mar. 2019

biblioteca *feature_selection* no scikit-learn.

### 3.2.4 Pós-processamento

Na última etapa, pós-processamento, foram avaliados e interpretados os padrões extraídos na etapa de mineração, ocorreram retornos a etapa anterior para mais iterações.

## 3.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foi detalhado como o método KDD foi aplicado nos dados disponibilizados e quais conhecimentos se pretendia extrair.

Os resultados desse processo serão detalhados no capítulo 4.

# 4 RESULTADOS

Neste capítulo são detalhados os resultados obtidos em cada fase do processo de KDD aplicado neste trabalho. Ele está nas mesmas seções do capítulo anterior, pois cada fase do KDD contribui para o resultado final do processo e gera seus próprios conhecimentos.

## 4.1 ENTRADA DE DADOS

Para a etapa de entrada de dados, foram desenvolvidos *scripts* em SQL que criavam tabelas na base de dados em MySQL com subconjuntos de dados relacionados com os cursos selecionados para esse estudo. Esses *scripts* estão disponíveis no Apêndice A. O Quadro 3 apresenta e resume essas tabelas auxiliares.

**Quadro 3** – Descrição das tabelas criadas apenas com dados dos cursos selecionados

| Tabela | Descrição |
|---|---|
| *adm* | Tabela base agrupando dados de cada aluno do curso Bacharelado em Administração Pública em cada disciplina em que ele cursou. |
| *adm_base_log_reduzido* | Tabela com registros de log dos alunos e cursos do curso Bacharelado em Administração Pública. |
| *adm_disciplinas* | Registra o identificador das disciplinas do curso de Bacharelado em Administração Pública, data de início e data de fim. |
| *adm_id_alunos* | Registra o identificador dos alunos matriculados no curso de Bacharelado em Administração Pública. |
| *lic_ped* | Tabela base agrupando dados de cada aluno do curso Licenciatura em Pedagogia em cada disciplina em que ele cursou. |
| *lic_ped_base_log_reduzido* | Tabela com registros de log dos alunos e cursos do curso Licenciatura em Pedagogia. |
| *lic_ped_disciplinas* | Registra o identificador das disciplinas do curso de Licenciatura em Pedagogia, data de início e data de fim. |
| *lic_ped_id_alunos* | Registra o identificador dos alunos matriculados no curso de Licenciatura em Pedagogia. |

**Fonte:** Elaborado pelo autor.

Essas tabelas se tornaram necessárias, pois, executar *scripts* que buscassem dados em toda a base do Moodle demandava horas de processamento. Em contrapartida, utilizando apenas essas tabelas o tempo de execução foi reduzido para poucos minutos.

## 4.2 PRÉ-PROCESSAMENTO

Utilizando as tabelas do Quadro 3 e os *scripts* disponíveis no Apêndice A, as variáveis, mapeadas para construtos da TDT por Ramos (2016) foram salvas como valores separados por vígulas (CSV) em uma tabela cujas colunas eram os identificadores listados no Quadro 4.

**Quadro 4** – Lista das variáveis e respectivos construtos e seus identificadores

| **Identificador** | **Variáveis** | **Construto** |
|---|---|---|
| VAR01 | Quantidade geral de postagens do aluno em fóruns, por disciplina. | Diálogo |
| VAR02 | Quantidade geral de mensagens enviadas pelo aluno dentro do ambiente, por semestre. | Diálogo |
| VAR03 | Quantidade geral de mensagens recebidas pelo aluno dentro do ambiente, por semestre. | Diálogo |
| VAR04 | Quantidade geral de recursos disponibilizados pelo professor (página web, vídeo, pdfs, entre outros) por disciplina. | Estrutura |
| VAR05a | Quantidade de acessos do aluno ao ambiente por turno (Manhã), por semestre. | Autonomia |
| VAR05b | Quantidade de acessos do aluno ao ambiente por turno (Tarde), por semestre. | Autonomia |
| VAR05c | Quantidade de acessos do aluno ao ambiente por turno (Noite), por semestre. | Autonomia |
| VAR06 | Quantidade de colegas diferentes para quem o aluno enviou mensagens no ambiente, por semestre. | Diálogo |
| VAR07 | Quantidade de acessos do aluno ao ambiente no semestre. | Autonomia |
| VAR08 | Quantidade de mensagens enviadas pelo aluno aos professores pelo ambiente, por semestre. | Diálogo |
| VAR09 | Quantidade de mensagens dos professores recebidas pelo aluno no ambiente, por semestre. | Diálogo |
| VAR10 | Quantidade de mensagens de colegas recebidas pelo aluno no ambiente, por semestre. | Diálogo |
| VAR11 | Quantidade de mensagens enviadas pelo aluno para outros colegas no ambiente, por semestre. | Diálogo |
| VAR12 | Quantidade de atividades com prazos de resposta ou envio definidos por professor, por disciplina. | Estrutura |

**Quadro 4 continuação da página anterior**

| Identificador | Variáveis | Construto |
|---|---|---|
| VAR13 | Quantidade de acessos do aluno aos diferentes tipos de atividades disponibilizadas (webquest, fórum, quiz, entre outros), por disciplina. | Autonomia |
| VAR14 | Quantidade de fóruns de discussão disponibilizados sobre os conteúdos por disciplina. | Estrutura |
| VAR15 | Quantidade de acessos do aluno aos fóruns, por disciplina. | Autonomia |

**Fonte:** Ramos (2016).

Além das variáveis, cada linha possui o identificador de um aluno e de uma disciplina. Dessa forma, cada linha representa um aluno em uma disciplina do curso.

Essas tabelas foram convertidas para planilhas eletrônicas para facilitar o processo de correção e validação dos resultados. As planilhas serviram como base para os processos de mineração de dados.

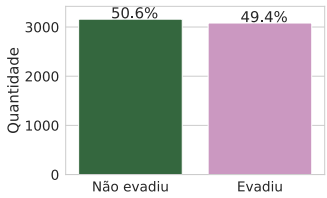
## 4.3 MINERAÇÃO DE DADOS

As planilhas eletrônicas foram carregadas em *dataframes*, como foi descrito no capítulo 3.

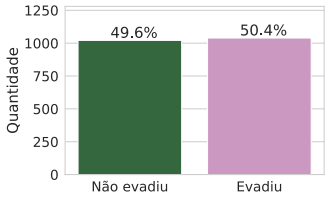
Os dados do curso de Licenciatura em Pedagogia somaram 6240 entradas. Sendo que, $50,6\%$ foram marcados como não evadidos e $49,4\%$ foram marcados como evadido, como ilustra a Figura 9a. A proporção das classes após a divisão dos dados em conjunto de testes e conjunto de treinamento é mostrada nas Figuras 9b e 9c, respectivamente.

**Figura 9** – Distribuição de classes para os dados do curso de Licenciatura em Pedagogia.

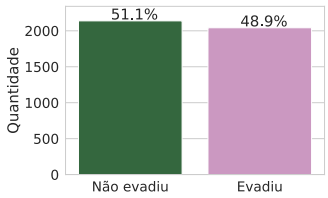
**(a)** Distribuição em todos os dados.




**(b)** Distribuição para os dados de teste.




**(c)** Distribuição para os dados de treinamento.




**Fonte:** Elaborado pelo autor.

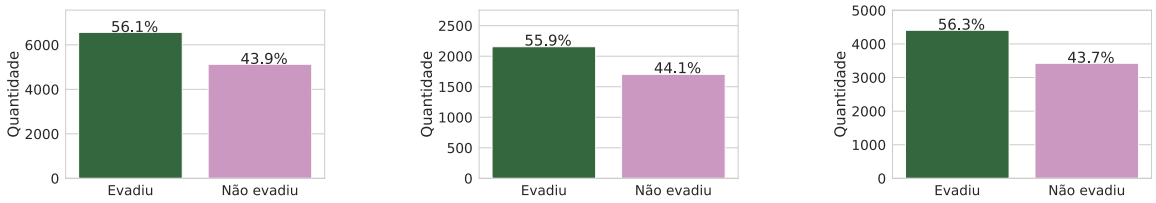
Os dados do curso de Bacharelado em Administração Pública somaram 11685 entradas. Sendo que, $43,9\%$ foram marcados como não evadidos e $56,1\%$ foram marcados como evadido, como ilustra a Figura 10a. A proporção das classes após a divisão dos dados em conjunto de testes e conjunto de treinamento é mostrada nas Figuras 10b e 10c, respectivamente.

**Figura 10** – Distribuição de classes para os dados do curso de Bacharelado em Administração Pública.

**(a)** Distribuição em todos os dados.

**(b)** Distribuição para os dados de teste.

**(c)** Distribuição para os dados de treinamento.




**Fonte:** Elaborado pelo autor.

Já que os conjuntos de dados mantiveram proporções semelhantes, os conjuntos de teste servirão como conjuntos de validação, as matrizes de confusão e as métricas serão calculadas utilizando-os.

As Tabelas 4 e 5 apresentam as estatísticas descritivas das variáveis coletadas no curso de Licenciatura em Pedagogia e Bacharelado em Administração Pública, respectivamente.

Sabendo-se dessas informações, principiaram-se os experimentos com os algoritmos de classificação. Destes, o primeiro foi o KNN. Iniciou-se pela escolha do parâmetro $k$, a quantidade de vizinhos. Para tal, foram testados valores entre 3 e 201, mantendo-se sempre uma quantidade impar de vizinhos. Esse experimento foi repetido, porém, com os dados normalizados, na tentativa de remover a influência da diferença de escalas entre as variáveis.

A acurácia do algoritmo foi escolhida como métrica nesses experimentos por ser uma métrica clássica e utilizada como padrão na biblioteca *Scikt-learn*.

A partir da análise dos gráficos gerados, foi escolhido o valor 3 para o parâmetro $k$ na aplicação do KNN sobre os dados do curso de Licenciatura em Pedagogia e o valor 1 para o curso de Bacharelado em Administração Pública.

Os algoritmos Árvore de Decisão e Regressão Logística não possuem parâmetros, portanto, não tiveram que passar pelo mesmo processo que o KNN.

Notou-se que a variável *VAR07* (Quantidade de acessos do aluno ao ambiente

**Tabela 4** – Estatísticas descritivas para as variáveis do curso de Licenciatura em Pedagogia

| Variável | Min | Média | Mediana | Max |
|---|---|---|---|---|
| VAR01 | 0,00 | 2,19 | 0,00 | 49,00 |
| VAR02 | 0,00 | 20,45 | 4,00 | 956,00 |
| VAR03 | 0,00 | 43,46 | 28,00 | 356,00 |
| VAR04 | 0,00 | 12,89 | 14,00 | 36,00 |
| VAR05a | 0,00 | 15,91 | 7,00 | 157,00 |
| VAR05b | 0,00 | 25,77 | 15,00 | 215,00 |
| VAR05c | 0,00 | 36,70 | 19,00 | 314,00 |
| VAR06 | 0,00 | 2,89 | 0,00 | 115,00 |
| VAR07 | 0,00 | 80,87 | 57,00 | 604,00 |
| VAR08 | 0,00 | 9,82 | 2,00 | 241,00 |
| VAR09 | 0,00 | 30,39 | 21,00 | 194,00 |
| VAR10 | 0,00 | 6,79 | 1,00 | 102,00 |
| VAR11 | 0,00 | 6,78 | 0,00 | 539,00 |
| VAR12 | 0,00 | 0,29 | 0,00 | 5,00 |
| VAR12 | 0,00 | 3,97 | 2,67 | 117,17 |
| VAR14 | 0,00 | 4,71 | 4,00 | 28,00 |
| VAR15 | 0,00 | 17,74 | 3,00 | 684,00 |

**Tabela 5** – Estatísticas descritivas para as variáveis do curso de Bacharelado em Administração Pública

| Variável | Min | Média | Mediana | Max |
|---|---|---|---|---|
| VAR01 | 0,00 | 1,42 | 0,00 | 59,00 |
| VAR02 | 0,00 | 21,66 | 2,00 | 1797,00 |
| VAR03 | 0,00 | 61,59 | 43,00 | 1114,00 |
| VAR04 | 0,00 | 17,70 | 17,00 | 81,00 |
| VAR05a | 0,00 | 20,60 | 7,00 | 402,00 |
| VAR05b | 0,00 | 27,46 | 14,00 | 342,00 |
| VAR05c | 0,00 | 29,96 | 15,00 | 272,00 |
| VAR06 | 0,00 | 1,09 | 0,00 | 96,00 |
| VAR07 | 0,00 | 80,40 | 50,00 | 1000,00 |
| VAR08 | 0,00 | 13,22 | 1,00 | 346,00 |
| VAR09 | 0,00 | 48,89 | 32,00 | 471,00 |
| VAR10 | 0,00 | 5,22 | 0,00 | 626,00 |
| VAR11 | 0,00 | 5,08 | 0,00 | 1598,00 |
| VAR12 | 0,00 | 0,13 | 0,00 | 6,00 |
| VAR12 | 0,00 | 2,65 | 1,67 | 55,10 |
| VAR14 | 0,00 | 4,66 | 5,00 | 20,00 |
| VAR15 | 0,00 | 9,55 | 0,00 | 533,00 |

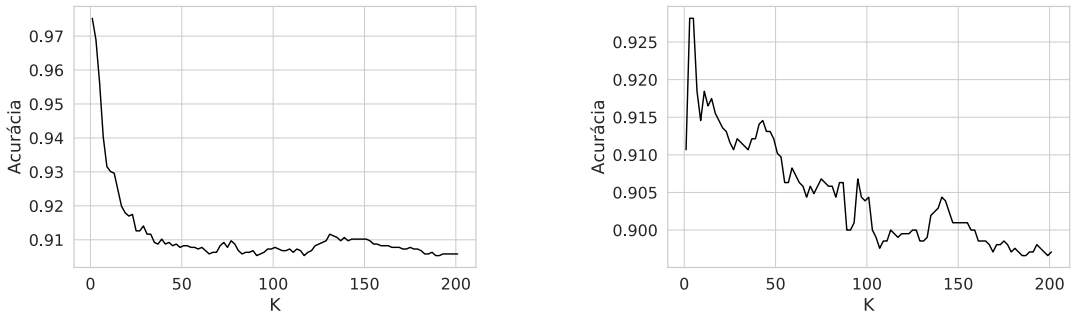
no semestre) apareceu como a mais relevante. Porém, como todas as outras variáveis só são contabilizadas depois que o aluno acessa o ambiente, essa variável se torna bastante enviesada. Devido a isso, a etapa de mineração foi executada novamente com a exclusão

da mesma.

Para o algoritmo KNN, o valor de $k$ foi ajustado para os dados sem a variável *VAR07*. As Figuras 11 e 12 ilustram os gráficos utilizados para a escolha do parâmetro $k$.

**Figura 11** – Gráficos de valores de acurácia para valores de K, com K variando entre 3 e 201 com incremento de 2 para o curso de Licenciatura em Pedagogia após a exclusão da variável *VAR07*.

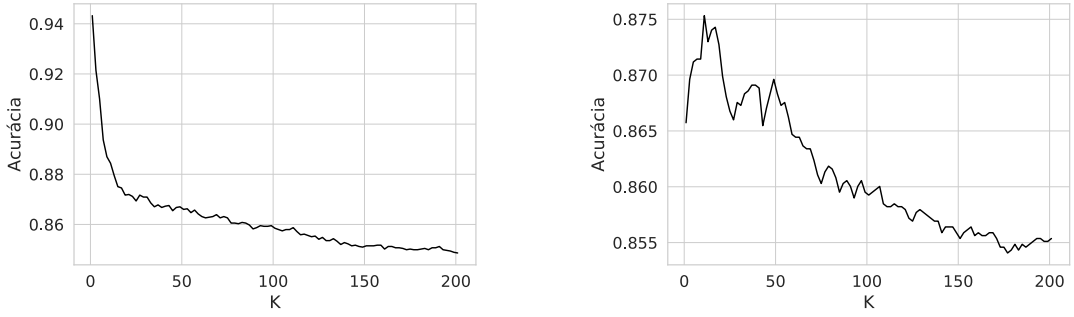
**(a)** Dados não normalizados. **(b)** Dados normalizados.




**Fonte:** Elaborado pelo autor.

**Figura 12** – Gráficos de valores de acurácia para valores de K, com K variando entre 3 e 201 com incremento de 2 para o cuso de Bacharelado em Administração Pública após a exclusão da variável *VAR07*.

**(a)** Dados não normalizados. **(b)** Dados normalizados.




**Fonte:** Elaborado pelo autor.

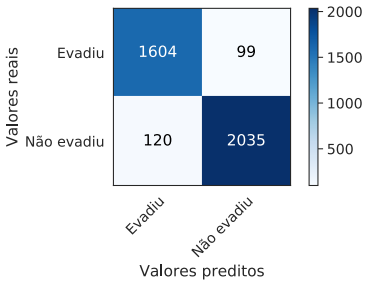
Após análise dos gráficos contidos nas Figuras 11a e 12a, optou-se para valores de $k = 1$ e os dados não normalizados em ambas as bases.

Em seguida, foram geradas novas matrizes de confusão que estão ilustradas na Figura 13.

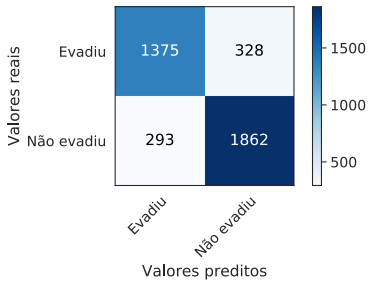
As matrizes de confusão serviram como base para o cálculo das métricas dos algoritmos de classificação, as Tabelas 6 e 7 apresentam os resultados para cada algoritmo.

**Figura 13** – Matrizes de confusão após a exclusão da *VAR07*.

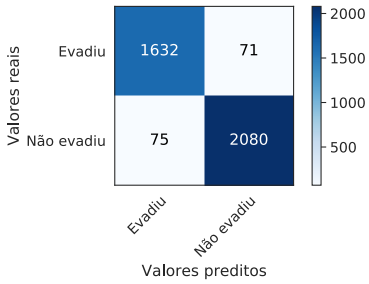
**(a)** KNN Administração.



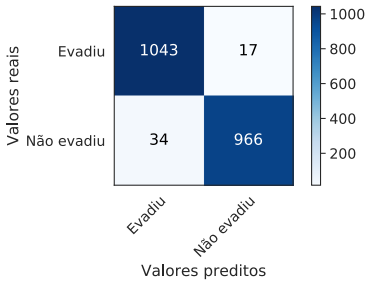

**(b)** LR Administração.



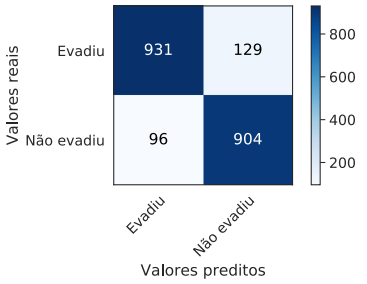

**(c)** Árvore de Decisçao Administração.



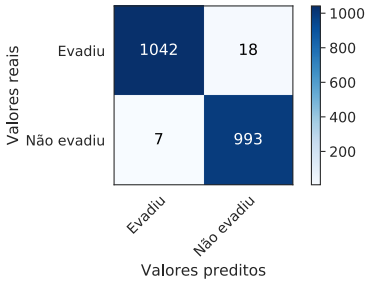

**(d)** KNN Pedagogia.




**(e)** LR Pedagogia.




**(f)** Árvore de Decisçao Pedagogia.




**Fonte:** Elaborado pelo autor.

**Tabela 6** – Métricas dos algoritmos aplicados no curso de Licenciatura em Pedagogia

| Algoritmo | Acurácia | Precisão | Sensibilidade | Especificidade | TFP | TFN |
|---|---|---|---|---|---|---|
| KNN | 0,9752 | 0,9827 | 0,9660 | 0,9840 | 0,0160 | 0,0340 |
| Árvore de Decisão | 0,9879 | 0,9841 | 0,9910 | 0,9849 | 0,0151 | 0,0090 |
| Regressão Logística | 0,8908 | 0,8751 | 0,9040 | 0,8783 | 0,1217 | 0,0960 |

**Fonte:** Elaborado pelo autor.

O algoritmo de Regressão Logística, embora tenha obtido boas métricas, teve o pior desempenho, diferente do que aconteceu no trabalho de Ramos (2016), em que o algoritmo obteve os melhores resultados. Supõe-se que isso se deve as características diferentes entre as bases de dados.

**Tabela 7** – Métricas dos algoritmos aplicados no curso de Bacharelado em Administração Pública

| Algoritmo | Acurácia | Precisão | Sensibilidade | Especificidade | TFP | TFN |
|---|---|---|---|---|---|---|
| KNN | 0,9432 | 0,9536 | 0,9443 | 0,9419 | 0,0581 | 0,0557 |
| Árvore de Decisão | 0,9614 | 0,9674 | 0,9633 | 0,9589 | 0,0411 | 0,0367 |
| Regressão Logística | 0,8390 | 0,8502 | 0,8640 | 0,8074 | 0,1926 | 0,1360 |

**Fonte:** Elaborado pelo autor.

Também foi realizada a análise da árvore de decisão gerada depois da remoção da variável *VAR07*. A representação gráfica das árvores para o curso de Licenciatura em Pedagogia e Bacharelado em Administração Pública estão ilustradas nas Figuras 14 e 15, respectivamente.

Nas novas árvores, a variável com maior importância foi a *VAR05c*, que representa a quantidade de acessos do aluno ao ambiente no período da noite, por semestre.

## 4.4 PÓS-PROCESSAMENTO

Os dados foram convertidos para formato CSV de forma que podem ser utilizados no futuro para treinamento ou validação de outros algoritmos de classificação.

Os scripts e visualizações geradas foram disponibilizadas publicamente em um repositório público.

## 4.5 CONHECIMENTOS OBTIDOS

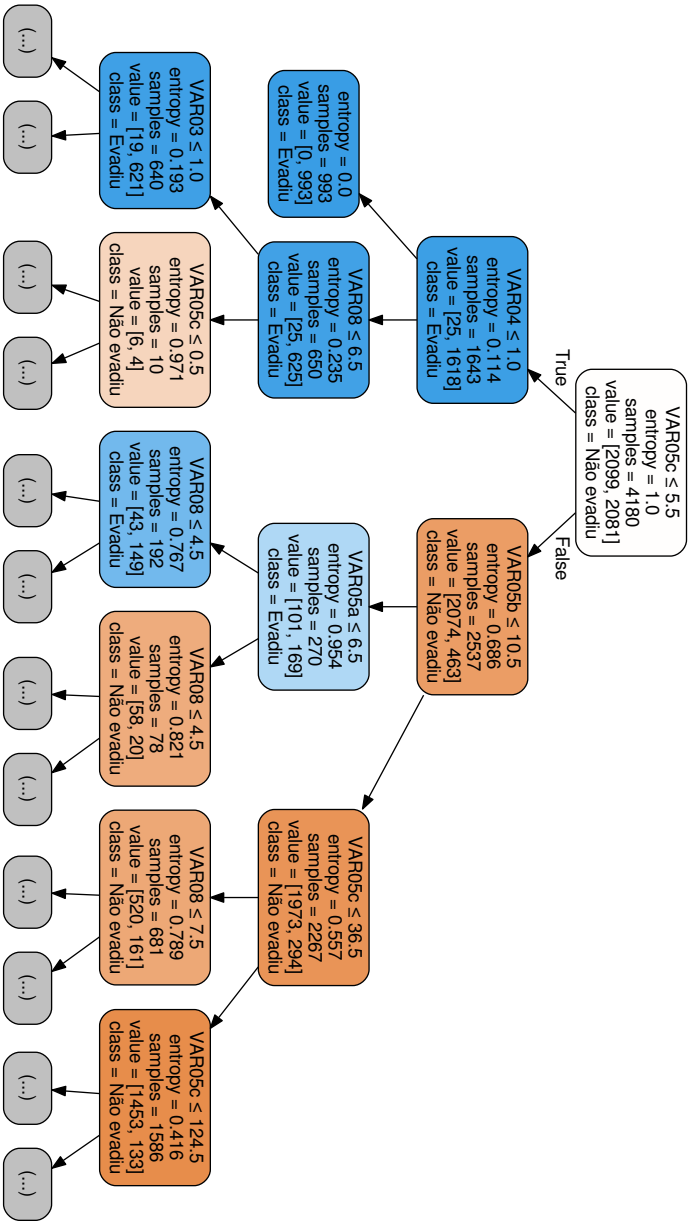
A análise dos dados permitiu perceber que os alunos da EAD na UNIVASF tendem a evadir entre o quarto e quinto semestre do curso.

Os algoritmos de classificação apresentaram métricas compatíveis com as encontradas na literatura.

Para o curso de Licenciatura em Pedagogia, o algoritmo Árvore de Decisão foi o que obteve as melhores métricas, com exceção da sensibilidade, onde ele ficou abaixo do KNN. A variável com maior importância, segundo o gráfico da árvore, foi a *VAR07* seguida pelas *VAR04*, *VAR08* e *VAR14*.

O mesmo comportamento foi observado para o curso de Bacharelado em Administração Pública. O fato da variável *VAR07* (Quantidade de acessos do aluno ao ambiente no semestre — Autonomia), ter sido destacada como mais importante em ambos os cursos é explicado pelo fato da mesma a variável que define a principal característica de um aluno evadir-se ou não, afinal sem acessar o ambiente virtual, nenhum recurso pode ser visualizado e nenhuma atividade pode ser executada, nenhum comportamento do aluno

**Figura 14** – Árvore de Decisão gerada a partir dos dados do curso de Licenciatura em Pedagogia.

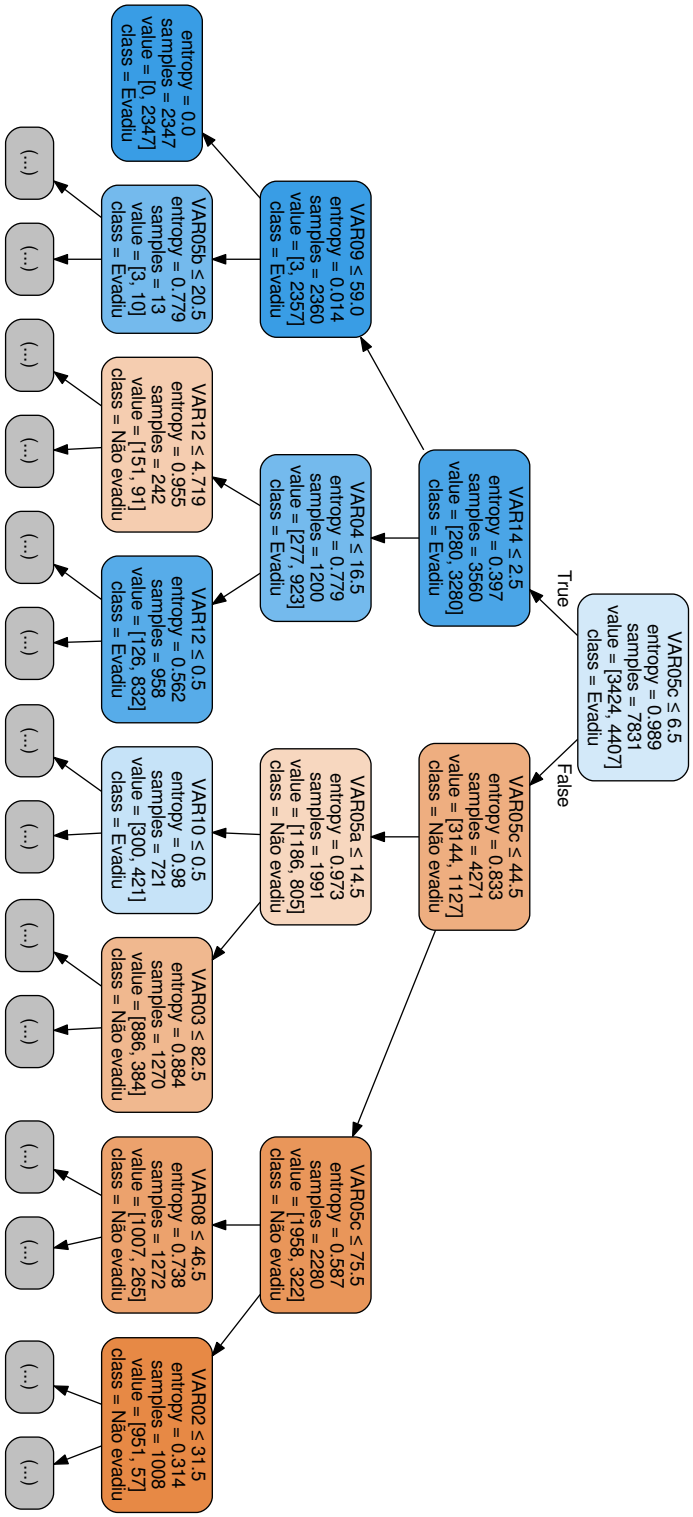



**Fonte:** Elaborado pelo autor.

pode ser verificado, além do baixo ou nenhum acesso aos cursos. Além disso, o fato da mesma apresentar uma grande variabilidade em seus valores, pode também ter influenciado nessa importância em relação às demais.

Após a segunda análise, sem a variável *VAR007*, a que obteve maior importância, para a árvore de decisão, foi a *VAR05c*, apesar de ainda ser uma variável relativa a quantidade de acessos, ela revela que os alunos que acessam o ambiente durante a noite

**Figura 15** – Árvore de Decisão gerada a partir dos dados do curso de Bacharelado em Administração Pública.




**Fonte:** Elaborado pelo autor.

são os que possuem menores chances de evasão.

## 4.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram apresentados os resultados obtidos em cada etapa do KDD aplicada neste trabalho, foi verificado a distribuição das classes nos conjuntos de dados, as estatísticas descritivas das variáveis, as métricas dos algoritmos de classificação e as visualizações das árvores de decisão.

No proximo capítulo, serão apresentadas as considerações finais desta pesquisa e sugestões de trabalhos futuros.

# 5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A modalidade EAD ajuda a democratizar o ensino, levando-o às regiões de difícil acesso aos professores ou dando a oportunidade ao estudante de criar sua própria rotina de estudos. A evasão desta modalidade de ensino ainda é um grande problema a ser resolvido, logo, existe a necessidade de pesquisa científica nesta área.

Com o uso crescente de ferramentas de tecnologia da informação em EAD fica evidente que o uso de aprendizagem de máquina pode ser utilizado para modelar e prever os fenômenos que causam a evasão.

O fluxo de descoberta de conhecimento em bases de dados descrito na metodologia deste trabalho foi utilizado como arcabouço para a aplicação dos algoritmos de classificação que foram comparados segundo métricas consolidadas na literatura. Os resultados obtidos foram satisfatórios, aproximando-se aos encontrados na literatura.

Os resultados obtidos neste trabalho, apontam que o uso das variáveis obtidas a partir dos contrutos da Teoria da Distância Transacional, podem ser usadas em ferramentas ou modelos preditivos da evasão dos alunos na EAD na UNIVASF, contemplando o objetivo geral do projeto.

As ferramentas selecionadas atenderam aos requisitos e apresentaram resultados semelhantes aos encontrados na literatura, o que valida seu uso em trabalhos futuros.

Como trabalhos futuros, propõe-se a elaboração de um sistema de controle para professores e gestores, que, integrado a um modelo de classificação, ajude a executar alguma ação no curso antes que a evasão do aluno ocorra. Sinalizações automáticas de alunos com risco de evasão podem ser implementadas, para alertar professores e tutores sobre essa situação. Além disso, um estudo aprofundado sobre relevância das variáveis preditoras, pode levar a uma redução da dimensionalidade das variáveis, sem perda do poder preditivo dos modelos. Isso pode impactar na perfomance do processo, pois com menos variáveis, uma menor quantidade de consultas ao banco de dados e um menor tempo de processamento dos modelos podem ser alcançados.

Propõe-se também, a elaboração desse estudo utilizando os dados dos cursos seccionados por semestre, com o objetivo de realizar a previsão da evasão em tempo hábil para que o gestor tome alguma decisão sobre o aluno com chance de evadir-se.

# REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE EDUCAÇÃO A DISTÂNCIA. **CENSO EAD.BR 2013**. 2014. Disponível em: <http://www.abed.org.br/censoead2013/CENSO_EAD_2013_PORTUGUES.pdf>. Acesso em: 13 jan. 2019. Citado na página 21.

_____. **CENSO EAD.BR 2014**: Relatório analítico da aprendizagem a distância no brasil. 2015. Disponível em: <http://www.abed.org.br/censoead2014/CensoEAD2014_portugues.pdf>. Acesso em: 13 jan. 2019. Citado na página 21.

_____. **CENSO EAD.BR 2015**: Relatório analítico da aprendizagem a distância no brasil. 2016. Disponível em: <http://abed.org.br/arquivos/Censo_EAD_2015_POR.pdf>. Acesso em: 13 jan. 2019. Citado na página 21.

_____. **CENSO EAD.BR 2016**: Relatório analítico da aprendizagem a distância no brasil. 2017. Disponível em: <http://abed.org.br/censoead2016/Censo_EAD_2016_portugues.pdf>. Acesso em: 13 jan. 2019. Citado na página 21.

BAKER, R.; ISOTANI, S.; CARVALHO, A. Mineraçao de dados educacionais: Oportunidades para o brasil. **Brazilian Journal of Computers in Education**, v. 19, n. 02, p. 03, 2011. Citado na página 25.

BAKER, R. *et al.* Data mining for education. **International encyclopedia of education**, Elsevier Oxford, UK, v. 7, n. 3, p. 112–118, 2010. Citado na página 25.

BENSON, R.; SAMARAWICKREMA, G. Addressing the context of e-learning: using transactional distance theory to inform design. **Distance Education**, Taylor & Francis, v. 30, n. 1, p. 5–21, 2009. Citado na página 19.

BOYD, R. D.; APPS, J. W. Redefining the discipline of adult education. **The AEA Handbook Series in Adult Education**, Jossey-Bass„ 1980. Citado na página 18.

CABAU, N. C. F.; COSTA, M. L. F. A teoria da distância transacional: um mapeamento de teses e dissertações brasileiras (the theory of transactional distance: a mapping of brazilian theses and dissertations). **Revista Eletrônica de Educação**, v. 12, n. 2, p. 431–447, 2018. Citado na página 19.

CHEN, Y.-J.; WILLITS, F. K. Dimensions of educational transactions in a videoconferencing learning environment. **American Journal of Distance Education**, Taylor & Francis, v. 13, n. 1, p. 45–59, 1999. Citado na página 19.

COSTA, E. *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. Citado 3 vezes nas páginas 23, 24 e 25.

COVER, T. M.; HART, P. E. *et al.* Nearest neighbor pattern classification. **IEEE transactions on information theory**, Menlo Park, v. 13, n. 1, p. 21–27, 1967. Citado na página 31.

DEWEY, J.; BENTLEY, A. F. **Knowing and the known**. [S.l.]: Beacon Press Boston, 1960. Citado na página 18.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37, 1996. Citado na página 23.

FIX, E.; HODGES, J. L. **Discriminatory analysis-nonparametric discrimination: consistency properties**. [S.l.], 1951. Citado na página 30.

GIL, A. C. Como elaborar projetos de pesquisa. **São Paulo**, v. 5, n. 61, p. 16–17, 2002. Citado 2 vezes nas páginas 35 e 36.

_____. **Métodos e técnicas de pesquisa social**. 6. ed. [S.l.]: Editora Atlas SA SA, 2008. Citado na página 35.

GOEL, L.; ZHANG, P.; TEMPLETON, M. Transactional distance revisited: Bridging face and empirical validity. **Computers in Human Behavior**, Elsevier, v. 28, n. 4, p. 1122–1129, 2012. Citado na página 22.

HAIR, J. F. *et al.* **Análise multivariada de dados**. [S.l.]: Bookman Editora, 2009. Citado na página 33.

HORZUM, M. B. Developing transactional distance scale and examining transactional distance perception of blended learning students in terms of different variables. **Educational sciences: Theory and practice**, ERIC, v. 11, n. 3, p. 1582–1587, 2011. Citado 3 vezes nas páginas 20, 22 e 23.

HUANG, X. *et al.* Understanding transactional distance in web-based learning environments: An empirical study. **British Journal of Educational Technology**, Wiley Online Library, v. 47, n. 4, p. 734–747, 2016. Citado 2 vezes nas páginas 19 e 21.

JAMES, G. *et al.* **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112. Citado 2 vezes nas páginas 32 e 33.

KLEINBAUM, D. G.; KLEIN, M. Analysis of matched data using logistic regression. **Logistic regression: A self-learning text**, Springer, p. 227–265, 2002. Citado na página 32.

MANHÃES, L. *et al.* Identificação dos fatores que influenciam a evasão em cursos de graduação através de sistemas baseados em mineração de dados: Uma abordagem quantitativa. **Anais do VIII Simpósio Brasileiro de Sistemas de Informação, São Paulo**, 2012. Citado na página 22.

MANHÃES, L. M. B. *et al.* Previsão de estudantes com risco de evasão utilizando técnicas de mineração de dados. **Simpósio Brasileiro de Informática na Educação-SBIE**, 2011. Citado na página 34.

MARCONI, M. d. A.; LAKATOS, E. M. **Fundamentos de Metodologia Científica**. 5. ed. São Paulo: Editora Atlas SA, 2003. Citado na página 35.

MBWESA, J. K. Transactional distance as a predictor of perceived learner satisfaction in distance learning courses: A case study of bachelor of education arts program, university of nairobi, kenya. **Journal of Education and Training Studies**, v. 2, n. 2, p. 176–188, 2014. Citado na página 22.

MOORE, M. G. Learner autonomy: The second dimension of independent learning. **Convergence**, International Council for Adult Education, v. 5, n. 2, p. 76, 1972. Citado na página 20.

_____. The theory of transactional distance. In: **Handbook of distance education**. [S.l.]: Routledge, 1973, 1993, 2013. Citado 5 vezes nas páginas 18, 19, 20, 21 e 22.

_____. Teoria da distância transacional. **Revista Brasileira de Aprendizagem Aberta e a Distância**, v. 1, n. 0, 2008. ISSN 1806-1362. Disponível em: <http://seer.abed.net.br/index.php/RBAAD/article/view/111>. Citado na página 16.

PAUL, R. C. *et al.* Revisiting zhang's scale of transactional distance: Refinement and validation using structural equation modeling. **Distance Education**, Taylor & Francis, v. 36, n. 3, p. 364–382, 2015. Citado 2 vezes nas páginas 20 e 22.

PAZ, F.; CAZELLA, S. Identificando o perfil de evasão de alunos de graduação através da mineração de dados educacionais: um estudo de caso de uma universidade comunitária. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2017. v. 6, n. 1, p. 624. Citado 2 vezes nas páginas 22 e 34.

PETERSON, L. E. K-nearest neighbor. **Scholarpedia**, v. 4, n. 2, p. 1883, 2009. Revision #137311. Citado na página 31.

QUEIROGA, E. M. *et al.* Modelo de predição da evasão de estudantes em cursos técnicos a distância a partir da contagem de interações. **Revista Thema**, v. 15, n. 2, p. 425–438, 2018. Citado na página 34.

RAMOS, J. L. C. **Uma abordagem preditiva da evasão na educação a distância a partir dos construtos da distância transacional**. Tese (Doutorado) — Universidade Federal de Pernambuco, 2016. Citado 9 vezes nas páginas 15, 16, 20, 34, 36, 38, 43, 44 e 48.

RAMOS, J. L. C. *et al.* Um estudo comparativo de classificadores na previsão da evasão de alunos em ead. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2018. v. 29, n. 1, p. 1463. Citado na página 33.

_____. Mapeamento de dados de um lms para medida de construtos da distância transacional. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2016. v. 27, n. 1, p. 1056. Citado 2 vezes nas páginas 33 e 34.

ROMERO, C.; VENTURA, S. Data mining in education. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 3, n. 1, p. 12–27, 2013. Citado na página 25.

RUMBLE, G. **The planning and management of distance education**. [S.l.]: Croom Helm, 1986. Citado na página 18.

RUSSELL, S.; NORVIG, P. Artificial intelligence a modern approach 3rd edition pdf. **Hong Kong: Pearson Education Asia**, 2011. Citado na página 26.

ŞEN, B.; UÇAR, E.; DELEN, D. Predicting and analyzing secondary education placement-test scores: A data mining approach. **Expert Systems with Applications**, Elsevier, v. 39, n. 10, p. 9468–9476, 2012. Citado na página 32.

SINDICATO DAS MANTENEDORAS DO ENSINO SUPERIOR. **Mapa do Ensino Superior no Brasil — 2015**. 2015. Disponível em: <http://convergenciacom.net/pdf/mapa-ensino-superior-brasil-2015.pdf>. Acesso em: 21 jan. 2019. Citado na página 22.

_____. **Mapa do Ensino Superior no Brasil — 2016**. 2016. Disponível em: <http://convergenciacom.net/pdf/mapa_ensino_superior_2016.pdf>. Acesso em: 21 jan. 2019. Citado na página 22.

STEINMAN, D. Educational experiences and the online student. **TechTrends**, Springer, v. 51, n. 5, p. 46–52, 2007. Citado 2 vezes nas páginas 22 e 23.

SUTHAHARAN, S. Machine learning models and algorithms for big data classification. In: **Integrated Series in Information Systems**. [S.l.]: Springer, 2016. v. 36. Citado 4 vezes nas páginas 26, 27, 29 e 30.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introdução ao datamining: mineração de dados**. [S.l.]: Ciência Moderna, 2009. Citado 7 vezes nas páginas 24, 26, 27, 28, 31, 32 e 36.

ZHANG, A. M. **Transactional distance in web-based college learning environments: Toward measurement and theory construction**. [S.l.]: Virginia Commonwealth University, 2003. Citado 2 vezes nas páginas 20 e 22.

# APÊNDICE A – SCRIPTS SQL UTILIZADOS NESSE TRABALHO

**Código 1** – Script para calcular data de inicio

```
DROP FUNCTION IF EXISTS CALCULATE_START_DATE;

DELIMITER //

CREATE FUNCTION CALCULATE_START_DATE(semester VARCHAR(255), course VARCHAR(255)) RETURNS
↪  VARCHAR(255) DETERMINISTIC
BEGIN
  DECLARE startdate BIGINT(10);

    IF course = 'adm' THEN
    CASE semester
      WHEN '2013.2' THEN SET startdate = UNIX_TIMESTAMP('2013-10-29');
      WHEN '2014.1' THEN SET startdate = UNIX_TIMESTAMP('2014-04-28');
      WHEN '2014.2' THEN SET startdate = UNIX_TIMESTAMP('2014-09-15');
      WHEN '2015.1' THEN SET startdate = UNIX_TIMESTAMP('2015-03-09');
      WHEN '2015.2' THEN SET startdate = UNIX_TIMESTAMP('2015-08-10');
      WHEN '2016.1' THEN SET startdate = UNIX_TIMESTAMP('2016-04-04');
      WHEN '2016.2' THEN SET startdate = UNIX_TIMESTAMP('2016-09-19');
      WHEN '2017.1' THEN SET startdate = UNIX_TIMESTAMP('2017-08-19');
      ELSE SET startdate = UNIX_TIMESTAMP('2017-08-19');
    END CASE;
  ELSE
    CASE semester
      WHEN '2014.2' THEN SET startdate = UNIX_TIMESTAMP('2014-09-10');
      WHEN '2015.1' THEN SET startdate = UNIX_TIMESTAMP('2015-03-09');
      WHEN '2015.2' THEN SET startdate = UNIX_TIMESTAMP('2015-08-10');
      WHEN '2016.1' THEN SET startdate = UNIX_TIMESTAMP('2016-04-18');
      WHEN '2016.2' THEN SET startdate = UNIX_TIMESTAMP('2016-09-19');
      WHEN '2017.1' THEN SET startdate = UNIX_TIMESTAMP('2017-02-20');
      WHEN '2017.2' THEN SET startdate = UNIX_TIMESTAMP('2017-09-25');
      WHEN '2018.1' THEN SET startdate = UNIX_TIMESTAMP('2018-03-01');
      ELSE SET startdate = UNIX_TIMESTAMP('2018-03-01');
    END CASE;
  END IF;

  RETURN startdate;
END//

DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 2** – Script para calcular data de fim

```
DROP FUNCTION IF EXISTS CALCULATE_END_DATE;

DELIMITER //

CREATE FUNCTION CALCULATE_END_DATE(semester VARCHAR(255), course VARCHAR(255)) RETURNS
↪  VARCHAR(255) DETERMINISTIC
BEGIN
  DECLARE enddate BIGINT(10);

    IF course = 'adm' THEN
    CASE semester
      WHEN '2013.2' THEN SET enddate = UNIX_TIMESTAMP('2014-04-13');
      WHEN '2014.1' THEN SET enddate = UNIX_TIMESTAMP('2014-08-31');
      WHEN '2014.2' THEN SET enddate = UNIX_TIMESTAMP('2015-02-09');
      WHEN '2015.1' THEN SET enddate = UNIX_TIMESTAMP('2015-07-18');
      WHEN '2015.2' THEN SET enddate = UNIX_TIMESTAMP('2015-12-31');
      WHEN '2016.1' THEN SET enddate = UNIX_TIMESTAMP('2016-08-07');
      WHEN '2016.2' THEN SET enddate = UNIX_TIMESTAMP('2016-12-25');
      WHEN '2017.1' THEN SET enddate = UNIX_TIMESTAMP('2017-10-25');
      ELSE SET enddate = UNIX_TIMESTAMP('2017-10-25');
    END CASE;
  ELSE
    CASE semester
      WHEN '2014.2' THEN SET enddate = UNIX_TIMESTAMP('2015-02-02');
      WHEN '2015.1' THEN SET enddate = UNIX_TIMESTAMP('2015-07-22');
      WHEN '2015.2' THEN SET enddate = UNIX_TIMESTAMP('2015-12-31');
      WHEN '2016.1' THEN SET enddate = UNIX_TIMESTAMP('2016-08-25');
      WHEN '2016.2' THEN SET enddate = UNIX_TIMESTAMP('2017-01-31');
      WHEN '2017.1' THEN SET enddate = UNIX_TIMESTAMP('2017-08-25');
      WHEN '2017.2' THEN SET enddate = UNIX_TIMESTAMP('2017-12-31');
      WHEN '2018.1' THEN SET enddate = UNIX_TIMESTAMP('2018-07-15');
      ELSE SET enddate = UNIX_TIMESTAMP('2018-07-15');
    END CASE;
  END IF;

  RETURN enddate;
END//

DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 3** – Script para remover espaços em branco internos

```
DELIMITER //
DROP FUNCTION IF EXISTS DELETE_INNER_SPACES//
CREATE FUNCTION DELETE_INNER_SPACES(str VARCHAR(255)) RETURNS VARCHAR(255) DETERMINISTIC
BEGIN
 while instr(str, ' ') > 0 do
 set str = replace(str, ' ', '');
 end while;
 return str;
END//
```

```
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 4** – Script para tratar os nomes dos semestres

```
DELIMITER //
DROP FUNCTION IF EXISTS HANDLE_SEMESTER//
CREATE FUNCTION HANDLE_SEMESTER(semester VARCHAR(255)) RETURNS VARCHAR(255) DETERMINISTIC
BEGIN
  return RIGHT(DELETE_INNER_SPACES(LCASE(semester)), 6);
END//
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 5** – Script para calcular o período

```
DELIMITER //
DROP FUNCTION IF EXISTS CALCULATE_PERIOD//
CREATE FUNCTION CALCULATE_PERIOD(first_semester VARCHAR(255), curr_semester VARCHAR(255))
↪  RETURNS INTEGER DETERMINISTIC
BEGIN
  SET @first_year = CAST(LEFT(first_semester,4)  AS UNSIGNED);
  SET @first_period = CAST(RIGHT(first_semester,1) AS UNSIGNED);
  SET @curr_year =  CAST(LEFT(curr_semester,4)  AS UNSIGNED);
  SET @curr_period =  CAST(RIGHT(curr_semester,1) AS UNSIGNED);
  return (@curr_year - @first_year) * 2 + @curr_period - @first_period + 1;
END//
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 6** – Script para remover espaços duplos

```
DELIMITER //
DROP FUNCTION IF EXISTS DELETE_DOUBLE_SPACES//
CREATE FUNCTION DELETE_DOUBLE_SPACES(str VARCHAR(255)) RETURNS VARCHAR(255) DETERMINISTIC
BEGIN
 while instr(str, '  ') > 0 do
 set str = replace(str, '  ', ' ');
 end while;
 return trim(str);
END//
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 7** – Script para remover pontos

```
DELIMITER //
DROP FUNCTION IF EXISTS REMOVE_DOTS//
CREATE FUNCTION REMOVE_DOTS(str VARCHAR(255)) RETURNS VARCHAR(255) DETERMINISTIC
BEGIN
  while instr(str, '.') > 0 do
    set str = replace(str, '.', '');
  end while;
  return str;
END//
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 8** – Script para criar as tabelas auxiliates

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_base;
CREATE PROCEDURE create_base
(
  IN base_table VARCHAR(30),
  IN first_semester VARCHAR(10),
  IN course_id BIGINT(10)
)
BEGIN
  SET @base = base_table;
  SET @first_semester = first_semester;
  SET @course_id = course_id;

  SET @query_drop_previous = CONCAT(
    'DROP TABLE IF EXISTS ', @base, ';'
  );

  PREPARE stmt FROM @query_drop_previous;
  EXECUTE stmt;
  DEALLOCATE PREPARE stmt;

  SET @query = CONCAT(
    ' CREATE TABLE  ', @base,
    '  ( periodo int(11) , ',
    ' disciplina_id bigint(10), ',
    'aluno_id bigint(10),',
    ' PRIMARY KEY (aluno_id, disciplina_id, periodo) ',
    ' ) ENGINE=InnoDB AS ',
    ' SELECT ',
      ' course.name AS \'curso\', ',
      ' HANDLE_SEMESTER(semester.name) AS \'semestre\', ',
      ' CALCULATE_PERIOD( ', @first_semester, ',HANDLE_SEMESTER(semester.name)) AS
    ↪  \'periodo\',',
      ' discipline.fullname AS \'disciplina_nome\', ',
```

```
    ' discipline.id  AS \'disciplina_id\', ',
    ' CALCULATE_START_DATE(HANDLE_SEMESTER(semester.name), SUBSTRING_INDEX(\'', @base, '\',
    ↪  \'_\', 1)) AS \'data_inicio\', ',
    ' CALCULATE_END_DATE(HANDLE_SEMESTER(semester.name), SUBSTRING_INDEX(\'', @base, '\',
    ↪  \'_\', 1)) AS \'data_fim\', ',
    ' participant.id AS \'aluno_id\', ',
    ' REMOVE_DOTS(DELETE_DOUBLE_SPACES(UCASE(CONCAT(participant.firstname, \'
    ↪  \',participant.lastname)))) AS \'aluno_nome\', ',
    ' participant.username as \'cpf\' ',
  ' FROM ',
    ' mdl_course discipline ',
      ' INNER JOIN ',
        ' mdl_course_categories semester ',
      ' ON ',
        ' (discipline.category = semester.id) ',
      ' INNER JOIN ',
        ' mdl_course_categories course ',
      ' ON ',
        ' (course.id = semester.parent) ',
      ' INNER JOIN ',
        ' mdl_enrol enrol ',
      ' ON ',
        ' (enrol.courseid = discipline.id) ',
      ' INNER JOIN ',
        ' mdl_user_enrolments user_enrolments ',
      ' ON ',
        ' (user_enrolments.enrolid = enrol.id) ',
      ' INNER JOIN ',
        ' mdl_user participant ',
      ' ON ',
        ' (participant.id = user_enrolments.userid) ',
      ' INNER JOIN ',
        ' mdl_role_assignments rs ',
      ' ON ',
        ' (rs.userid = participant.id) ',
      ' INNER JOIN ',
        ' mdl_context e ',
      ' ON ',
        ' (e.id = rs.contextid AND discipline.id = e.instanceid) ',
  ' WHERE ',
    ' course.id =  ', @course_id,' AND',
    ' e.contextlevel = 50 AND ',
    ' rs.roleid = 5 AND ',
    ' semester.name NOT REGEXP \'.*REOFERTA.*|.*REPERCURSO.*\' ',
  ' ORDER BY ',
    ' curso, ',
    ' semestre, ',
    ' periodo, ',
    ' disciplina_id, ',
    ' aluno_nome; '
);

PREPARE stmt FROM @query;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
```

```
END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 9** – Script para criar a tabela de alunos

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_view_alunos;
CREATE PROCEDURE create_view_alunos()
BEGIN

  DROP TABLE IF EXISTS alunos;
  CREATE TABLE alunos (
    disciplina_id BIGINT(10) NOT NULL,
    aluno_id BIGINT(10) NOT NULL,

    PRIMARY KEY(disciplina_id, aluno_id)
  ) AS
  SELECT c.id AS 'disciplina_id', u.id AS 'aluno_id'
  FROM mdl_role_assignments rs
  INNER JOIN mdl_context e ON rs.contextid=e.id
  INNER JOIN mdl_course c ON c.id = e.instanceid
  INNER JOIN mdl_user u ON u.id=rs.userid
  WHERE e.contextlevel=50 AND rs.roleid=5
  ORDER BY c.id, u.id;

END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 10** – Script para criar a tabela de professores

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_view_professores;
CREATE PROCEDURE create_view_professores()
BEGIN

  DROP TABLE IF EXISTS professores;

  CREATE TABLE professores (
    disciplina_id BIGINT(10) NOT NULL,
    professor_id BIGINT(10) NOT NULL,
    PRIMARY KEY (disciplina_id, professor_id)
  ) AS
  SELECT DISTINCT c.id AS 'disciplina_id', u.id AS 'professor_id'
  FROM mdl_role_assignments rs
  INNER JOIN mdl_context e ON rs.contextid=e.id
  INNER JOIN mdl_course c ON c.id = e.instanceid
```

```
  INNER JOIN mdl_user u ON u.id=rs.userid
  WHERE e.contextlevel=50 AND rs.roleid IN (3,4)
  ORDER BY c.id, u.id;

END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 11** – Script para criar a tabela de postagens em forúm

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_view_posts;
CREATE PROCEDURE create_view_posts()
BEGIN

  DROP TABLE IF EXISTS posts;

  CREATE TABLE posts AS
  SELECT
    fd.course AS 'disciplina_id',
    p2.created AS 'data',
    f.name AS 'nome_forum',
    p2.id AS 'post',
    p2.parent,
    p2.userid AS 'emissor',
    p1.userid AS 'receptor'
  FROM mdl_forum_posts p1
  INNER JOIN mdl_forum_posts p2 ON  p1.id=p2.parent
  INNER JOIN mdl_forum_discussions fd ON p1.discussion=fd.id
  INNER JOIN mdl_forum f ON fd.forum=f.id
  ORDER BY fd.course, p2.userid;

  ALTER TABLE posts ADD INDEX(disciplina_id);

END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 12** – Script para criar a tabelas compartilhadas

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_shared_views;
CREATE PROCEDURE create_shared_views()
BEGIN
  CALL create_view_alunos();
  CALL create_view_professores();
  CALL create_view_posts();
END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 13** – Script para criar tabela das disciplinas de um curso

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_view_disciplinas;

CREATE PROCEDURE create_view_disciplinas
(
  IN base_table VARCHAR(30)
)
BEGIN

  SET @base = base_table;
  SET @view_base = CONCAT(@base, '_disciplinas');

  SET @query = CONCAT(
    ' CREATE OR REPLACE VIEW ',
    @view_base,
    ' AS  SELECT DISTINCT (disciplina_id), data_inicio, data_fim FROM ',
    @base,
    '  ORDER BY disciplina_id; '
  );

  PREPARE stmt FROM @query;
  EXECUTE stmt;

  DEALLOCATE PREPARE stmt;

END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 14** – Script para criar tabela com os ids dos alunos de um curso

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_view_id_alunos;
CREATE PROCEDURE create_view_id_alunos
(
  IN base_table VARCHAR(30)
)
BEGIN

  SET @base = base_table;
  SET @view_base = CONCAT(@base, '_id_alunos');

  SET @query = CONCAT(
    ' CREATE OR REPLACE VIEW ',
```

```
    @view_base,
    ' AS  SELECT distinct(aluno_id) FROM ',
    @base,
    ' ORDER BY aluno_id;'
  );

  PREPARE stmt FROM @query;
  EXECUTE stmt;

  DEALLOCATE PREPARE stmt;

END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 15** – Script para criar tabela com os ids das disciplinas de um curso

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_view_id_disciplinas;
CREATE PROCEDURE create_view_id_disciplinas
(
  IN base_table VARCHAR(30)
)
BEGIN

  SET @base = base_table;
  SET @view_base = CONCAT(@base, '_id_disciplinas');

  SET @query = CONCAT(
    'CREATE OR REPLACE VIEW ',
    @view_base,
    ' AS SELECT distinct(disciplina_id) FROM ',
    @base,
    ' ORDER BY disciplina_id; '
  );

  PREPARE stmt FROM @query;
  EXECUTE stmt;

  DEALLOCATE PREPARE stmt;

END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 16** – Script para criar tabela com os logs de um curso

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_views_log_reduzido;
CREATE PROCEDURE create_views_log_reduzido
(
  IN base_table VARCHAR(30)
)
BEGIN

  SET @base = base_table;

  SET @view_base_log_reduzido = CONCAT(@base, '_base_log_reduzido');

  SET @query_drop_base_log_reduzido = CONCAT(
    ' DROP TABLE IF EXISTS ',
      @view_base_log_reduzido,
    ' ;'
  );

  PREPARE stmt FROM @query_drop_base_log_reduzido;
  EXECUTE stmt;
  DEALLOCATE PREPARE stmt;

  SET @query_base_log_reduzido = CONCAT(
    'CREATE TABLE ',
    @view_base_log_reduzido,
    ' AS SELECT @curRank := @curRank + 1 AS id,time,userid,course,module,action,ip,cmid FROM
    ↪  ',
    ' mdl_log , (SELECT @curRank := 0) r ',
    ' WHERE ',
      ' action IN (\'login\' , \'view\', \'view forum\') ',
      ' AND module IN ( ',
        ' \'assign\', ',
        ' \'forum\', ',
        ' \'assignment\', ',
        ' \'choice\', ',
        ' \'feedback\', ',
        ' \'survey\', ',
        ' \'chat\', ',
        ' \'quiz\', ',
        ' \'resource\', ',
        ' \'folder\', ',
        ' \'url\', ',
        ' \'page\', ',
        ' \'book\', ',
        ' \'user\'); '
  );

  PREPARE stmt FROM @query_base_log_reduzido;
  EXECUTE stmt;
  DEALLOCATE PREPARE stmt;

END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 17** – Script para criar as tabelas específicas de cada cruso

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_specific_views;
CREATE PROCEDURE create_specific_views(
  IN base VARCHAR(255)
)
BEGIN
  CALL create_view_disciplinas(base);
  CALL create_view_id_alunos(base);
  CALL create_view_id_disciplinas(base);
  CALL create_views_log_reduzido(base);
END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 18** – Script que prepara as tabelas base

```
DELIMITER //
DROP PROCEDURE IF EXISTS prepare_base_tables;
CREATE PROCEDURE prepare_base_tables()
BEGIN
  SET @adm = 'adm';
  SET @lic_pedagogia = 'lic_pedagogia';

  SELECT CONCAT('Criando tabela ', @adm);
  CALL create_base(@adm, '2013.2', 43);

  SELECT CONCAT('Criando tabela ', @lic_pedagogia);
  CALL create_base(@lic_pedagogia, '2014.2', 64);

  SELECT 'Criando Views compatilhadas...';
  CALL create_shared_views();

  SELECT CONCAT('Criando views ', @adm);
  CALL create_specific_views(@adm);

  SELECT CONCAT('Criando views ', @lic_pedagogia);
  CALL create_specific_views(@lic_pedagogia);
END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 19** – Script que cria a tabela com as variáveis

```
DELIMITER //
DROP PROCEDURE IF EXISTS transational_distance;
CREATE PROCEDURE transational_distance
(
  IN base VARCHAR(30)
)
BEGIN
  SET @base = base;
  SET @view_alunos = 'alunos';
  SET @view_disciplinas = CONCAT(@base, '_disciplinas');
  SET @view_professores = 'professores';
  SET @view_base_log_reduzido = CONCAT(@base, '_base_log_reduzido');
  SET @view_dist_tran = CONCAT(@base, '_dist_tran');

  SET @drop_query = CONCAT(
    'DROP TABLE IF EXISTS ', @view_dist_tran, ';'
  );

  PREPARE stmt FROM @drop_query;
  EXECUTE stmt;
  DEALLOCATE PREPARE stmt;

  SET @query = CONCAT(
    ' CREATE TABLE ', @view_dist_tran, ' AS ',
      ' SELECT ' ,
      ' ', @base,'.curso AS \'Curso\', ' ,
      ' ', @base,'.semestre AS \'Semestre\', ',
      ' ', @base,'.periodo AS \'Período\', ',
      ' ', @base,'.disciplina_nome AS \'Nome da Disciplina\', ',
      ' ', @base,'.disciplina_id AS \'ID da Disciplina\', ',
      ' ', @base,'.data_inicio AS \'Data de Início\', ',
      ' ', @base,'.data_fim AS \'Data de Final\', ' ,
      ' ', @base,'.aluno_nome AS \'Nome do Aluno\', ',
      ' ', @base,'.aluno_id AS \'ID do Aluno\', ',
      ' ', @base,'.cpf AS \'CPF\', ',
      ' IFNULL(VAR01.VAR01,0) AS \'VAR01\', ',
      ' IFNULL(VAR04.VAR04,0) AS \'VAR04\', ',
      ' IFNULL(VAR05.VAR05,0) AS \'VAR05\', ',
      ' IFNULL(VAR08.VAR08,0) AS \'VAR08\', ',
      ' IFNULL(VAR09.VAR09,0) AS \'VAR09\', ',
      ' IFNULL(VAR13a.VAR13a,0) AS \'VAR13a\', ',
      ' IFNULL(VAR13b.VAR13b,0) AS \'VAR13b\', ',
      ' IFNULL(VAR13c.VAR13c,0) AS \'VAR13c\', ',
      ' IFNULL(VAR16.VAR16,0) AS \'VAR16\', ',
      ' IFNULL(VAR18.VAR18,0) AS \'VAR18\', ',
      ' IFNULL(VAR19.VAR19,0) AS \'VAR19\', ',
      ' IFNULL(VAR20.VAR20,0) AS \'VAR20\', ',
      ' IFNULL(VAR21.VAR21,0) AS \'VAR21\', ',
      ' IFNULL(VAR22.VAR22,0) AS \'VAR22\', ',
      ' IFNULL(VAR23.VAR23,0) AS \'VAR23\', ',
      ' IFNULL(VAR27.VAR27,0) AS \'VAR27\', ',
      ' IFNULL(VAR28.VAR28,0) AS \'VAR28\', ',
      ' IFNULL(VAR31.VAR31,0) AS \'VAR31\' ',
      ' FROM ' ,
```

```
' (SELECT * FROM ', @base,') AS ', @base,' ' ,
' LEFT OUTER JOIN ' ,
  ' (SELECT b.disciplina_id, b.aluno_id, count(*) AS \'VAR01\' ',
  ' FROM mdl_forum_posts p ' ,
  ' INNER JOIN mdl_forum_discussions d ON d.id = p.discussion ',
  ' INNER JOIN ', @base,' b ON d.course=b.disciplina_id AND p.userid=b.aluno_id AND
  ↪  p.created BETWEEN b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id, b.aluno_id) AS VAR01 ',
  ' ON VAR01.disciplina_id = ', @base,'.disciplina_id AND VAR01.aluno_id = ',
  ↪  @base,'.aluno_id ' ,
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id, b.aluno_id, count(*) AS \'VAR04\' ' ,
  ' FROM mdl_message_read r ',
  ' INNER JOIN ', @base,' b ON b.aluno_id=r.useridfrom AND r.timecreated BETWEEN
  ↪  b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id, b.aluno_id) AS VAR04 ',
  ' ON VAR04.disciplina_id = ', @base,'.disciplina_id AND VAR04.aluno_id = ',
  ↪  @base,'.aluno_id ' ,
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id, b.aluno_id, count(*) AS \'VAR05\' ' ,
  ' FROM mdl_message_read r ',
  ' INNER JOIN ', @base,' b ON b.aluno_id=r.useridto AND r.timecreated BETWEEN
  ↪  b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id, b.aluno_id) AS VAR05 ',
  ' ON VAR05.disciplina_id = ', @base,'.disciplina_id AND VAR05.aluno_id = ',
  ↪  @base,'.aluno_id '  ,
' LEFT OUTER JOIN ',
  ' (SELECT temp.disciplina_id, count(*) AS \'VAR08\' ' ,
  ' FROM (SELECT b.disciplina_id,module,cmid, count(*) ' ,
  ' FROM (SELECT distinct(disciplina_id), data_inicio, data_fim FROM ', @base,') b ' ,
  ' INNER JOIN (SELECT * FROM ', @view_base_log_reduzido, ' WHERE cmid IS NOT NULL AND
  ↪  ',
  ' (module=\'resource\' OR ',
  ' module=\'folder\' OR ',
  ' module=\'url\' OR ',
  ' module=\'page\' OR ',
  ' module=\'book\')) l ',
  ' ON b.disciplina_id=l.course AND l.time BETWEEN b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id,l.module,cmid) AS temp ',
  ' GROUP BY temp.disciplina_id) AS VAR08 ',
  ' ON VAR08.disciplina_id = ', @base,'.disciplina_id ' ,
' LEFT OUTER JOIN ',
  ' (SELECT temp.disciplina_id, count(*) AS \'VAR09\' ' ,
  ' FROM (SELECT b.disciplina_id,module,cmid, count(*) ' ,
  ' FROM (SELECT distinct(disciplina_id), data_inicio, data_fim FROM ', @base,') b ' ,
  ' INNER JOIN (SELECT * FROM ', @view_base_log_reduzido, ' WHERE cmid IS NOT NULL AND
  ↪  /*Atividades*/(module=\'assign\' OR module=\'forum\' OR module=\'quiz\')) l ',
  ' ON b.disciplina_id=l.course AND l.time BETWEEN b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id,l.module,cmid) AS temp ',
  ' GROUP BY temp.disciplina_id) AS VAR09 ',
  ' ON VAR09.disciplina_id = ', @base,'.disciplina_id ' ,
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id,b.aluno_id, count(*) AS \'VAR13a\' ' ,
  ' FROM ', @base,' b ' ,
  ' INNER JOIN (SELECT * FROM ', @view_base_log_reduzido, ' WHERE action=\'login\' AND
  ↪  HOUR(FROM_UNIXTIME(time)) >= 6 AND HOUR(FROM_UNIXTIME(time)) < 12) l ',
```

```
  ' ON b.aluno_id=l.userid AND l.time BETWEEN b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id,b.aluno_id) AS VAR13a ',
  ' ON  VAR13a.aluno_id = ', @base,'.aluno_id AND VAR13a.disciplina_id = ',
  ↪  @base,'.disciplina_id ' ,
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id,b.aluno_id, count(*) AS \'VAR13b\' ',
  ' FROM ', @base,' b ' ,
  ' INNER JOIN (SELECT * FROM ', @view_base_log_reduzido, ' WHERE action=\'login\' AND
  ↪  HOUR(FROM_UNIXTIME(time)) >= 12 AND HOUR(FROM_UNIXTIME(time)) < 18) l ',
  ' ON b.aluno_id=l.userid AND l.time BETWEEN b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id,b.aluno_id) AS VAR13b ',
  ' ON  VAR13b.aluno_id = ', @base,'.aluno_id AND VAR13b.disciplina_id = ',
  ↪  @base,'.disciplina_id ',
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id,b.aluno_id, count(*) AS \'VAR13c\' ',
  ' FROM ', @base,' b ' ,
  ' INNER JOIN (SELECT * FROM ', @view_base_log_reduzido, ' WHERE action=\'login\' AND
  ↪  HOUR(FROM_UNIXTIME(time)) >= 18 AND HOUR(FROM_UNIXTIME(time)) < 24) l ',
  ' ON b.aluno_id=l.userid AND l.time BETWEEN b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id,b.aluno_id) AS VAR13c ',
  ' ON  VAR13c.aluno_id = ', @base,'.aluno_id AND VAR13c.disciplina_id = ',
  ↪  @base,'.disciplina_id ',
' LEFT OUTER JOIN ',
  ' (SELECT temp.disciplina_id, temp.aluno_id, count(*) AS \'VAR16\' ',
  ' FROM (SELECT b.disciplina_id, b.aluno_id, r.useridto, count(*) AS \'VAR16_temp\' ',
  ' FROM mdl_message_read r ',
  ' INNER JOIN ', @base, ' b ON b.aluno_id=r.useridfrom AND r.timecreated BETWEEN
  ↪  b.data_inicio and b.data_fim ',
  ' INNER JOIN ', @view_alunos, ' a ON a.aluno_id=r.useridto AND
  ↪  a.disciplina_id=b.disciplina_id ',
  ' GROUP BY b.disciplina_id, b.aluno_id,r.useridto) AS temp ',
  ' GROUP BY temp.disciplina_id, temp.aluno_id) AS VAR16 ',
  ' ON VAR16.disciplina_id = ', @base, '.disciplina_id AND VAR16.aluno_id = ',
  ↪  @base,'.aluno_id ',
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id,b.aluno_id, count(*) AS \'VAR18\' ',
  ' FROM ', @base,' b ' ,
  ' INNER JOIN (SELECT * FROM ', @view_base_log_reduzido, ' WHERE action=\'login\') l
  ↪  ',
  ' ON b.aluno_id=l.userid AND l.time BETWEEN b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id,b.aluno_id) AS VAR18 ',
  ' ON  VAR18.aluno_id = ', @base,'.aluno_id AND VAR18.disciplina_id = ',
  ↪  @base,'.disciplina_id ' ,
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id, b.aluno_id, count(*) AS \'VAR19\' ' ,
  ' FROM mdl_message_read r ',
  ' INNER JOIN ', @base,' b ON b.aluno_id=r.useridfrom AND r.timecreated BETWEEN
  ↪  b.data_inicio and b.data_fim ',
  ' INNER JOIN ', @view_professores, ' p ON p.professor_id=r.useridto AND
  ↪  p.disciplina_id=b.disciplina_id ',
  ' GROUP BY b.disciplina_id, b.aluno_id) AS VAR19 ',
  ' ON VAR19.disciplina_id = ', @base,'.disciplina_id AND VAR19.aluno_id = ',
  ↪  @base,'.aluno_id ',
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id, b.aluno_id, count(*) AS \'VAR20\' ',
```

```
  ' FROM mdl_message_read r ',
  ' INNER JOIN ', @base, ' b ON b.aluno_id=r.useridto AND r.timecreated BETWEEN
  ↪  b.data_inicio and b.data_fim ',
  ' INNER JOIN ', @view_professores, ' p ON p.professor_id=r.useridfrom AND
  ↪  p.disciplina_id=b.disciplina_id ',
  ' GROUP BY b.disciplina_id, b.aluno_id) AS VAR20 ',
  ' ON VAR20.disciplina_id = ', @base,'.disciplina_id AND VAR20.aluno_id = ',
  ↪  @base,'.aluno_id ',
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id, b.aluno_id, count(*) AS \'VAR21\' ' ,
  ' FROM mdl_message_read r ',
  ' INNER JOIN ', @base,' b ON b.aluno_id=r.useridto AND r.timecreated BETWEEN
  ↪  b.data_inicio and b.data_fim ',
  ' INNER JOIN ', @view_alunos, ' a ON a.aluno_id=r.useridfrom AND
  ↪  a.disciplina_id=b.disciplina_id ',
  ' GROUP BY b.disciplina_id, b.aluno_id) AS VAR21 ',
  ' ON VAR21.disciplina_id = ', @base,'.disciplina_id AND VAR21.aluno_id = ',
  ↪  @base,'.aluno_id ' ,
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id, b.aluno_id, count(*) AS \'VAR22\' ' ,
  ' FROM mdl_message_read r ',
  ' INNER JOIN ', @base,' b ON b.aluno_id=r.useridfrom AND r.timecreated BETWEEN
  ↪  b.data_inicio and b.data_fim ',
  ' INNER JOIN ', @view_alunos, ' a ON a.aluno_id=r.useridto AND
  ↪  a.disciplina_id=b.disciplina_id ',
  ' GROUP BY b.disciplina_id, b.aluno_id) AS VAR22 ',
  ' ON VAR22.disciplina_id = ', @base,'.disciplina_id AND VAR22.aluno_id = ',
  ↪  @base,'.aluno_id ' ,
' LEFT OUTER JOIN ',
  ' (SELECT b.disciplina_id, count(*) AS \'VAR23\' ' ,
  ' FROM mdl_assign a ',
  ' INNER JOIN (SELECT distinct(disciplina_id),data_inicio,data_fim  FROM ', @base,') b
  ↪  ' ,
  ' ON b.disciplina_id=a.course AND a.duedate BETWEEN b.data_inicio and b.data_fim ',
  ' GROUP BY b.disciplina_id) AS VAR23 ',
  ' ON VAR23.disciplina_id = ', @base,'.disciplina_id ' ,
' LEFT OUTER JOIN ',
  ' (SELECT temp.disciplina_id,temp.aluno_id, AVG(temp.Acesso_Objeto) AS \'VAR27\' ' ,
  ' FROM (SELECT b.disciplina_id,b.aluno_id, module,cmid, count(*) AS \'Acesso_Objeto\'
  ↪  ',
  ' FROM ', @base,' b ' ,
  ' INNER JOIN (SELECT * FROM ', @view_base_log_reduzido, ' WHERE cmid IS NOT NULL AND '
  ↪  ,
    ' (module=\'assign\' AND action=\'view\') OR ' ,
    ' action=\'view forum\' OR ',
    ' (module=\'assignment\' AND action=\'view\') OR ',
    ' (module=\'choice\' AND action=\'view\') OR ',
    ' (module=\'feedback\' AND action=\'view\') OR ',
    ' (module=\'survey\' AND action=\'view\') OR '
    ' (module=\'chat\' AND action=\'view\') OR '
    ' (module=\'quiz\' AND action=\'view\')) l ',
  ' ON b.disciplina_id=l.course AND b.aluno_id=l.userid AND l.time BETWEEN b.data_inicio
  ↪  and b.data_fim ',
  ' GROUP BY b.disciplina_id,b.aluno_id,l.module,cmid) AS temp ',
  ' GROUP BY temp.disciplina_id, temp.aluno_id) AS VAR27 ',
```

```
        ' ON VAR27.disciplina_id = ', @base,'.disciplina_id AND VAR27.aluno_id = ',
        ↪  @base,'.aluno_id '    ,
      ' LEFT OUTER JOIN ',
        ' (SELECT temp.disciplina_id, count(*) AS \'VAR28\' FROM ',
        ' (SELECT distinct b.disciplina_id,f.id, f.course ' ,
        ' FROM mdl_forum f ' ,
        ' INNER JOIN mdl_forum_discussions d ON f.id=d.forum ',
        ' INNER JOIN mdl_forum_posts p ON d.id=p.discussion ',
        ' INNER JOIN (SELECT distinct(disciplina_id),data_inicio,data_fim FROM ', @base,') b
        ↪  ',
        ' ON b.disciplina_id=f.course AND p.created BETWEEN b.data_inicio and b.data_fim ' ,
        ' ) temp ',
        ' GROUP BY temp.disciplina_id) AS VAR28 ',
        ' ON VAR28.disciplina_id = ', @base,'.disciplina_id ' ,
      ' LEFT OUTER JOIN ',
        ' (SELECT b.disciplina_id,b.aluno_id, count(*) AS \'VAR31\' ',
        ' FROM ', @base,' b ' ,
        ' INNER JOIN (SELECT * FROM ', @view_base_log_reduzido, ' WHERE action=\'view forum\')
        ↪  l ',
        ' ON b.aluno_id=l.userid AND b.disciplina_id=l.course AND l.time BETWEEN b.data_inicio
        ↪  and b.data_fim ',
        ' GROUP BY b.disciplina_id,b.aluno_id) AS VAR31 ',
        ' ON  VAR31.aluno_id = ', @base,'.aluno_id AND VAR31.disciplina_id = ',
        ↪  @base,'.disciplina_id ' ,
      ' ORDER BY ' ,
        ' ', @base,'.curso, ', @base,'.semestre, ', @base,'.periodo, ',
        ↪  @base,'.disciplina_nome, ', @base,'.aluno_nome; '
  );

  PREPARE stmt FROM @query;
  EXECUTE stmt;
  DEALLOCATE PREPARE stmt;

END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

**Código 20** – Script que executa o script da criação das tabelas

```
DELIMITER //
DROP PROCEDURE IF EXISTS mine_ead_moodle_data;
CREATE PROCEDURE mine_ead_moodle_data()
BEGIN
  SET @adm_turma1 = 'adm_turma1';
  SET @lic_pedagogia = 'lic_pedagogia_turma1';
  SET @adm_turma2 = 'adm_turma2_old';
  SET @lic_pedagogia_turma2 = 'lic_pedagogia_turma2_old';

  SELECT CONCAT('Criando dataset ', @adm_turma1);
  CALL transational_distance(@adm_turma1);

  SELECT CONCAT('Criando dataset ', @lic_pedagogia);
```

```
  CALL transational_distance(@lic_pedagogia);

  SELECT CONCAT('Criando dataset ', @adm_turma2);
  CALL transational_distance(@adm_turma2);

  SELECT CONCAT('Criando dataset ', @lic_pedagogia_turma2);
  CALL transational_distance(@lic_pedagogia_turma2);
END //
DELIMITER ;
```

**Fonte:** Elaborado pelo autor.

# APÊNDICE B – SCRIPTS PYTHON UTILIZADOS NESSE TRABALHO

**Código 21** – Script que limpa e prepara os dados do curso de Licenciatura em Pedagogia

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
get_ipython().run_line_magic('matplotlib', 'inline')


# In[2]:


df = pd.read_csv('data/pedagogiaDataframe.csv')


# In[3]:


df.head()


# Vamos manter apenas os dados utilizados pelos algoritmos de aprendizagem

# In[4]:


newIdx = ['VAR01',
          'VAR04',
          'VAR05',
          'VAR08',
          'VAR13a',
          'VAR13b',
          'VAR13c',
          'VAR16',
          'VAR18',
          'VAR19',
          'VAR20',
          'VAR21',
          'VAR22',
          'VAR23',
          'VAR27',
          'VAR28',
          'VAR31',
          'EVADIU'
         ]
```

```
df = df[newIdx]


# Adapta os nomes das variáveis

# In[5]:


df.columns = ['VAR01',
              'VAR02',
              'VAR03',
              'VAR04',
              'VAR05a',
              'VAR05b',
              'VAR05c',
              'VAR06',
              'VAR07',
              'VAR08',
              'VAR09',
              'VAR10',
              'VAR11',
              'VAR12',
              'VAR12',
              'VAR14',
              'VAR15',
              'EVADIU'
             ]


# In[6]:


df.describe().transpose()[['min', 'mean', '50%', 'max']]


# In[7]:


df.info()


# Mostrar a distribuição dos dados em relação à variável alvo

# In[8]:


sns.set_style("whitegrid")
sns.set_context("paper", font_scale=2.0)
ax = sns.countplot(x='EVADIU',
    data=pd.DataFrame(
    df['EVADIU'].replace({0: 'Não evadiu', 1: 'Evadiu'})),
    palette="cubehelix")
plt.ylabel('Quantidade')
plt.xlabel('')
```

```
total = len(df['EVADIU'])
plt.ylim(0, (total / 2 + 300))

for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_height()/total)
    x = (p.get_x() + p.get_width() / 2) - 0.15
    y = p.get_height() + 20.0
    ax.annotate(percentage, (x, y))

plt.savefig(fname='images/barplot_pedagogia.svg', format='svg')


# # Classificação

# In[9]:


from sklearn.model_selection import train_test_split


# In[10]:


X = df.drop(['EVADIU'],axis=1)
y = df['EVADIU']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=4)


# #### Relatório de balanceamento de classes

# Função auxiliar para plotagem

# In[11]:


def plot_class_balance(data, xlabel, ylabel, name):
    sns.set_style("whitegrid")
    sns.set_context("paper", font_scale=2.0)
    ax = sns.countplot(data.replace({0: 'Não evadiu', 1: 'Evadiu'}), palette="cubehelix")
    plt.ylabel(ylabel)
    plt.xlabel(xlabel)

    total = len(data)
    plt.ylim(0, (total / 2 + 250))
    for p in ax.patches:
        percentage = '{:.1f}%'.format(100 * p.get_height()/total)
        x = (p.get_x() + p.get_width() / 2) - 0.15
        y = p.get_height() + 20.0
        ax.annotate(percentage, (x, y))
    plt.savefig(fname=name, format='svg')


# Dados de treinamento

# In[12]:
```

```
plot_class_balance(y_train, '', 'Quantidade', 'images/barplot_pedagogia_treinamento.svg')


# Dados de teste

# In[13]:


plot_class_balance(y_test, '', 'Quantidade', 'images/barplot_pedagogia_testes.svg')


# ### Funções auxiliares para gerar métricas e relatórios

# In[14]:


from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels


# In[15]:


def plot_confusion_matrix(y_true, y_pred, classes, name,
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    sns.set_style('white')
    sns.set_context("paper", font_scale=1.5)

    # Compute confusion matrix
    cm = confusion_matrix(y_true, y_pred)
    # Only use the labels that appear in the data
    classes = classes[unique_labels(y_true, y_pred)]

    fig, ax = plt.subplots()
    im = ax.imshow(cm, interpolation='nearest', cmap=cmap)
    ax.figure.colorbar(im, ax=ax)
    # We want to show all ticks...
    ax.set(xticks=np.arange(cm.shape[1]),
           yticks=np.arange(cm.shape[0]),
           # ... and label them with the respective list entries
           xticklabels=classes, yticklabels=classes,
           title='',
           ylabel='Valores reais',
           xlabel='Valores preditos')

    # Rotate the tick labels and set their alignment.
    plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
             rotation_mode="anchor")

    # Loop over data dimensions and create text annotations.
```

```
    fmt = 'd'
    thresh = cm.max() / 2.
    for i in range(cm.shape[0]):
        for j in range(cm.shape[1]):
            ax.text(j, i, format(cm[i, j], fmt),
                    ha="center", va="center",
                    color="white" if cm[i, j] > thresh else "black")
    fig.tight_layout()

    plt.savefig(fname=name, format='svg')
    return ax


np.set_printoptions(precision=2)


# In[16]:


def custom_classification_report(y_true, y_pred, title='Relatórios das métricas de
↪  classificação'):
    tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()

    print(title + '\n')

    print('Acurácia: \t\t%5.4f' % ((tp + tn) / (tp + tn + fp + fn)))

    print('Precisão: \t\t%5.4f' % (tp / (tp + fp)))

    print('Sensibilidade: \t\t%5.4f' % (tp / (tp + fn)))

    print('Especificidade: \t%5.4f' % (tn / (tn + fp)))

    print('TFP: \t\t\t%5.4f' % (fp / (fp + tn)))

    print('TFN: \t\t\t%5.4f' % (fn / (fn + tp)))


# In[17]:


def make_accs_knn(x_train, x_test, y_train, y_test, test_values):
    accs = []
    best_k = 1
    maxi = 0.
    for k in test_values:
        knn = KNeighborsClassifier(n_neighbors=k, n_jobs=-1)
        knn.fit(x_train, y_train)
        acc = knn.score(x_test, y_test)
        if(acc > maxi):
            maxi = acc
            best_k = k
        accs.append(acc)
    return accs, maxi, best_k
```

```
# In[18]:


def plot_acc(x_train, x_test, y_train, y_test, test_values, name):
    sns.set_style("whitegrid")
    sns.set_context("paper", font_scale=1.5)
    accs, maxi, best_k = make_accs_knn(x_train, x_test, y_train, y_test, test_values)
    plt.plot(test_values, accs, 'k')
    plt.xlabel('K')
    plt.ylabel('Acurácia')
    plt.savefig(fname=name, format='svg')
    plt.show()
    print('Maior acurácia: '+ str(maxi))
    print('Melhor k: '+ str(best_k))


# ## KNN

# In[19]:


from sklearn.neighbors import KNeighborsClassifier, NeighborhoodComponentsAnalysis


# Testando os valores de K entre 0 e 203

# In[20]:


plot_acc(X_train, X_test, y_train, y_test, list(range(1, 203, 2)), 'images/knn_neigh_ped.svg')


# Na tentativa de melhorar a previsão com KNN, iremos normalizar os dados.

# In[21]:


from sklearn import preprocessing


# In[22]:


# Get column names first
names = X_train.columns

# Create the Scaler object
scaler = preprocessing.StandardScaler()

# Fit your data on the scaler object
X_train_scaled = scaler.fit_transform(X_train)
X_train_scaled = pd.DataFrame(X_train_scaled, columns=names)

X_test_scaled = scaler.fit_transform(X_test)
X_test_scaled = pd.DataFrame(X_test_scaled, columns=names)
```

```
# Realizando mesmo teste com k entre 0 e 203

# In[23]:


plot_acc(X_train_scaled, X_test_scaled, y_train, y_test, list(range(1, 203, 2)),
↪  'images/knn_neigh_norm_ped.svg')


# O melhor resultado foi para K = 3 sem normalização, vamos verificar todas as variáveis.

# In[24]:


knn = KNeighborsClassifier(n_neighbors=3, n_jobs=-1)


# In[25]:


knn.fit(X_train, y_train)


# In[26]:


predictions = knn.predict(X_test)


# In[27]:


custom_classification_report(y_test, predictions, 'Relatório de métricas para o KNN com dados
↪  não normalizados')


# In[28]:


plot_confusion_matrix(y_test, predictions, name='images/cm_knn_ped.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))
plt.savefig(fname='images/test.svg', format='svg')


# ### Decision Tree

# In[29]:


from sklearn.tree import DecisionTreeClassifier, export_graphviz
import graphviz


# In[30]:
```

```
treemodel = DecisionTreeClassifier(criterion='entropy')
treemodel.fit(X_train,y_train)


# In[31]:


predictions = treemodel.predict(X_test)


# In[32]:


custom_classification_report(y_test, predictions, 'Relatório de Métricas Para Árvore de
↪  Decisão')


# In[33]:


plot_confusion_matrix(y_test, predictions, name='images/cm_tree_ped.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))


# In[34]:


export_graphviz(treemodel, out_file='images/ped_tree.dot',
                           max_depth=3,
                           feature_names=X.columns,
                           class_names=['Não evadiu', 'Evadiu'],
                           filled=True, rounded=True,
                           special_characters=True)

dot_data = export_graphviz(treemodel, out_file=None,
                           max_depth=3,
                           feature_names=X.columns,
                           class_names=['Não evadiu', 'Evadiu'],
                           filled=True, rounded=True,
                           special_characters=True)
graph = graphviz.Source(dot_data)

graph


# #### Importância de variáveis utilizando SelectFromModel

# In[35]:


treemodel.feature_importances_


# In[36]:
```

```
from sklearn.feature_selection import SelectFromModel


# In[37]:


treemodel = DecisionTreeClassifier(criterion='entropy')
treemodel.fit(X, y)
model = SelectFromModel(treemodel, prefit=True)


# In[38]:


X_new = model.transform(X)
X_new.shape


# VAR07 é a mais importante para a árvore de decisão

# ### Regressão Logistica

# In[39]:


from sklearn.linear_model import LogisticRegression


# In[40]:


logmodel = LogisticRegression(solver='liblinear')
logmodel.fit(X_train,y_train)


# In[41]:


predictions = logmodel.predict(X_test)


# In[42]:


custom_classification_report(y_test, predictions, 'Relatório de Métricas Para Regressão
↪  Logistica')


# In[43]:


plot_confusion_matrix(y_test, predictions, name='images/cm_rl_ped.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))
```

```
# # Testes sem a variável `VAR07`, ela será removida por ser uma variável enviesada.

# In[44]:


df = df.drop('VAR07',axis=1)
df.head()


# In[45]:


X = df.drop(['EVADIU'],axis=1)
y = df['EVADIU']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=0)


# ### Testes KNN

# Escolha do parametro K novamente

# In[46]:


plot_acc(X_train, X_test, y_train, y_test, list(range(1, 203, 2)),
↪  'images/knn_neigh_ped_no_var07.svg')


# Normaliza os dados sem variável `VAR07`

# In[47]:


# Get column names first
names = X_train.columns

# Create the Scaler object
scaler = preprocessing.StandardScaler()

# Fit your data on the scaler object
X_train_scaled = scaler.fit_transform(X_train)
X_train_scaled = pd.DataFrame(X_train_scaled, columns=names)

X_test_scaled = scaler.fit_transform(X_test)
X_test_scaled = pd.DataFrame(X_test_scaled, columns=names)


# In[48]:


plot_acc(X_train_scaled, X_test_scaled, y_train, y_test, list(
    range(1, 203, 2)), 'images/knn_neigh_norm_ped_no_var07.svg')


# Resultado: K = 1 e dados não normalizados
```

```
# In[49]:


knn = KNeighborsClassifier(n_neighbors=1, n_jobs=-1)
knn.fit(X_train, y_train)


# Executando e gerando os relatórios para o KNN

# In[50]:


predictions = knn.predict(X_test)


# In[51]:


custom_classification_report(y_test, predictions, 'Relatório de métricas para o KNN com dados
↪  não normalizados')


# In[52]:


plot_confusion_matrix(y_test, predictions, name='images/cm_knn_ped_no_var07.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))


# ### Testes com Árvore de Decisão

# In[53]:


treemodel = DecisionTreeClassifier(criterion='entropy')
treemodel.fit(X_train,y_train)


# In[54]:


predictions = treemodel.predict(X_test)


# In[55]:


custom_classification_report(y_test, predictions, 'Relatório de Métricas Para Árvore de
↪  Decisão')


# In[56]:


plot_confusion_matrix(y_test, predictions, name='images/cm_tree_ped_no_var07.svg',
```

```
                    classes=np.array(['Evadiu', 'Não evadiu']))


# In[57]:


export_graphviz(treemodel, out_file='images/ped_tree_no_var07.dot',
                           max_depth=3,
                           feature_names=X.columns,
                           class_names=['Não evadiu', 'Evadiu'],
                           filled=True, rounded=True,
                           special_characters=True)

dot_data = export_graphviz(treemodel, out_file=None,
                           max_depth=3,
                           feature_names=X.columns,
                           class_names=['Não evadiu', 'Evadiu'],
                           filled=True, rounded=True,
                           special_characters=True)
graph = graphviz.Source(dot_data)
graph


# In[58]:


treemodel.feature_importances_


# In[59]:


treemodel = DecisionTreeClassifier(criterion='entropy')
treemodel.fit(X, y)
model = SelectFromModel(treemodel, prefit=True)


# In[60]:


X_new = model.transform(X)
X_new.shape


# ### Testes com RL

# In[61]:


logmodel = LogisticRegression(solver='liblinear')
logmodel.fit(X_train,y_train)


# In[62]:
```

```
predictions = logmodel.predict(X_test)


# In[63]:


custom_classification_report(y_test, predictions, 'Relatório de Métricas Para Regressão
↪  Logistica')


# In[64]:


plot_confusion_matrix(y_test, predictions, name='images/cm_rl_ped_no_var07.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))
```

**Fonte:** Elaborado pelo autor.

**Código 22** – Script que limpa e prepara os dados do curso de Bacharelado em Administração Pública

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
get_ipython().run_line_magic('matplotlib', 'inline')


# In[2]:


df = pd.read_csv('data/admDataframe.csv')


# In[3]:


df.head()


# Vamos manter apenas os dados utilizados pelos algoritmos de aprendizagem

# In[4]:


newIdx = ['VAR01',
          'VAR04',
```

```
           'VAR05',
           'VAR08',
           'VAR13a',
           'VAR13b',
           'VAR13c',
           'VAR16',
           'VAR18',
           'VAR19',
           'VAR20',
           'VAR21',
           'VAR22',
           'VAR23',
           'VAR27',
           'VAR28',
           'VAR31',
           'EVADIU'
          ]

df = df[newIdx]


# Adapta os nomes das variáveis

# In[5]:


df.columns = ['VAR01',
              'VAR02',
              'VAR03',
              'VAR04',
              'VAR05a',
              'VAR05b',
              'VAR05c',
              'VAR06',
              'VAR07',
              'VAR08',
              'VAR09',
              'VAR10',
              'VAR11',
              'VAR12',
              'VAR12',
              'VAR14',
              'VAR15',
              'EVADIU'
             ]


# In[6]:


df.describe().transpose()[['min', 'mean', '50%', 'max']]


# In[7]:
```

```
df.info()


# Mostrar a distribuição dos dados em relação à variável alvo

# In[8]:


sns.set_style("whitegrid")
sns.set_context("paper", font_scale=2.0)
ax = sns.countplot(x='EVADIU',
    data=pd.DataFrame(
    df['EVADIU'].replace({0: 'Não evadiu', 1: 'Evadiu'})),
    palette="cubehelix")
plt.ylabel('Quantidade')
plt.xlabel('')

total = len(df['EVADIU'])
evadedCount = df['EVADIU'].value_counts()[1]
plt.ylim(0, (evadedCount + 1000))

for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_height()/total)
    x = (p.get_x() + p.get_width() / 2) - 0.15
    y = p.get_height() + 20.0
    ax.annotate(percentage, (x, y))

plt.savefig(fname='images/barplot_adm.svg', format='svg')


# # Classificação

# In[9]:


from sklearn.model_selection import train_test_split


# In[10]:


X = df.drop(['EVADIU'],axis=1)
y = df['EVADIU']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=0)


# #### Relatório de balanceamento de classes

# Função auxiliar para plotagem

# In[11]:


def plot_class_balance(data, xlabel, ylabel, name):
    sns.set_style("whitegrid")
```

```
    sns.set_context("paper", font_scale=2.0)
    ax = sns.countplot(data.replace({0: 'Não evadiu', 1: 'Evadiu'}), palette="cubehelix")
    plt.ylabel(ylabel)
    plt.xlabel(xlabel)

    total = len(data)
    evadedCount = data.value_counts()[1]
    plt.ylim(0, (evadedCount + 600))
    for p in ax.patches:
        percentage = '{:.1f}%'.format(100 * p.get_height()/total)
        x = (p.get_x() + p.get_width() / 2) - 0.15
        y = p.get_height() + 20.0
        ax.annotate(percentage, (x, y))
    plt.savefig(fname=name, format='svg')


# Dados de treinamento

# In[12]:


plot_class_balance(y_train, '', 'Quantidade', 'images/barplot_adm_treinamento.svg')


# Dados de teste

# In[13]:


plot_class_balance(y_test, '', 'Quantidade', 'images/barplot_adm_testes.svg')


# ### Funções auxiliares para gerar métricas e relatórios

# In[14]:


from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels


# In[15]:


def plot_confusion_matrix(y_true, y_pred, classes, name,
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    sns.set_style('white')
    sns.set_context("paper", font_scale=1.5)

    # Compute confusion matrix
    cm = confusion_matrix(y_true, y_pred)
    # Only use the labels that appear in the data
```

```
    classes = classes[unique_labels(y_true, y_pred)]

    fig, ax = plt.subplots()
    im = ax.imshow(cm, interpolation='nearest', cmap=cmap)
    ax.figure.colorbar(im, ax=ax)
    # We want to show all ticks...
    ax.set(xticks=np.arange(cm.shape[1]),
           yticks=np.arange(cm.shape[0]),
           # ... and label them with the respective list entries
           xticklabels=classes, yticklabels=classes,
           title='',
           ylabel='Valores reais',
           xlabel='Valores preditos')

    # Rotate the tick labels and set their alignment.
    plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
             rotation_mode="anchor")

    # Loop over data dimensions and create text annotations.
    fmt = 'd'
    thresh = cm.max() / 2.
    for i in range(cm.shape[0]):
        for j in range(cm.shape[1]):
            ax.text(j, i, format(cm[i, j], fmt),
                    ha="center", va="center",
                    color="white" if cm[i, j] > thresh else "black")
    fig.tight_layout()

    plt.savefig(fname=name, format='svg')
    return ax


np.set_printoptions(precision=2)


# In[16]:


def custom_classification_report(y_true, y_pred, title='Relatórios das métricas de
↪  classificação'):
    tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()

    print(title + '\n')

    print('Acurácia: \t\t%5.4f' % ((tp + tn) / (tp + tn + fp + fn)))

    print('Precisão: \t\t%5.4f' % (tp / (tp + fp)))

    print('Sensibilidade: \t\t%5.4f' % (tp / (tp + fn)))

    print('Especificidade: \t%5.4f' % (tn / (tn + fp)))

    print('TFP: \t\t\t%5.4f' % (fp / (fp + tn)))

    print('TFN: \t\t\t%5.4f' % (fn / (fn + tp)))
```

```
# In[17]:


def make_accs_knn(x_train, x_test, y_train, y_test, test_values):
    accs = []
    best_k = 1
    maxi = 0.
    for k in test_values:
        knn = KNeighborsClassifier(n_neighbors=k, n_jobs=-1)
        knn.fit(x_train, y_train)
        acc = knn.score(x_test, y_test)
        if(acc > maxi):
            maxi = acc
            best_k = k
        accs.append(acc)
    return accs, maxi, best_k


# In[18]:


def plot_acc(x_train, x_test, y_train, y_test, test_values, name):
    sns.set_style("whitegrid")
    sns.set_context("paper", font_scale=1.5)
    accs, maxi, best_k = make_accs_knn(x_train, x_test, y_train, y_test, test_values)
    plt.plot(test_values, accs, 'k')
    plt.xlabel('K')
    plt.ylabel('Acurácia')
    plt.savefig(fname=name, format='svg')
    plt.show()
    print('Maior acurácia: '+ str(maxi))
    print('Melhor k: '+ str(best_k))


# ## KNN

# In[19]:


from sklearn.neighbors import KNeighborsClassifier, NeighborhoodComponentsAnalysis


# Testando os valores de K entre 0 e 203

# In[20]:


plot_acc(X_train, X_test, y_train, y_test, list(range(1, 203, 2)), 'images/knn_neigh_adm.svg')


# Na tentativa de melhorar a previsão com KNN, iremos normalizar os dados.

# In[21]:
```

```
from sklearn import preprocessing


# In[22]:


# Get column names first
names = X_train.columns

# Create the Scaler object
scaler = preprocessing.StandardScaler()

# Fit your data on the scaler object
X_train_scaled = scaler.fit_transform(X_train)
X_train_scaled = pd.DataFrame(X_train_scaled, columns=names)

X_test_scaled = scaler.fit_transform(X_test)
X_test_scaled = pd.DataFrame(X_test_scaled, columns=names)


# Realizando mesmo teste com k entre 0 e 203

# In[23]:


plot_acc(X_train_scaled, X_test_scaled, y_train, y_test, list(range(1, 203, 2)),
↪  'images/knn_neigh_norm_adm.svg')


# O melhor resultado foi para K = 1 sem normalização, vamos verificar todas as variáveis.

# In[24]:


knn = KNeighborsClassifier(n_neighbors=1, n_jobs=-1)


# In[25]:


knn.fit(X_train, y_train)


# In[26]:


predictions = knn.predict(X_test)


# In[27]:


custom_classification_report(y_test, predictions, 'Relatório de métricas para o KNN com dados
↪  não normalizados')
```

```
# In[28]:


plot_confusion_matrix(y_test, predictions, name='images/cm_knn_adm.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))


# ### Decision Tree

# In[29]:


from sklearn.tree import DecisionTreeClassifier, export_graphviz
import graphviz


# In[30]:


treemodel = DecisionTreeClassifier(criterion='entropy')
treemodel.fit(X_train,y_train)


# In[31]:


predictions = treemodel.predict(X_test)


# In[32]:


custom_classification_report(y_test, predictions, 'Relatório de Métricas Para Árvore de
↪  Decisão')


# In[33]:


plot_confusion_matrix(y_test, predictions, name='images/cm_tree_adm.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))


# In[34]:


export_graphviz(treemodel, out_file='images/adm_tree.dot',
                           max_depth=3,
                           feature_names=X.columns,
                           class_names=['Não evadiu', 'Evadiu'],
                           filled=True, rounded=True,
                           special_characters=True)

dot_data = export_graphviz(treemodel, out_file=None,
                           max_depth=3,
                           feature_names=X.columns,
```

```
                           class_names=['Não evadiu', 'Evadiu'],
                           filled=True, rounded=True,
                           special_characters=True)
graph = graphviz.Source(dot_data)
graph


# #### Importância de variáveis utilizando SelectFromModel

# In[35]:


treemodel.feature_importances_


# In[36]:


from sklearn.feature_selection import SelectFromModel


# In[37]:


treemodel = DecisionTreeClassifier(criterion='entropy')
treemodel.fit(X, y)
model = SelectFromModel(treemodel, prefit=True)


# In[38]:


X_new = model.transform(X)
X_new.shape


# VAR07 é a mais importante para a árvore de decisão

# ### Regressão Logistica

# In[39]:


from sklearn.linear_model import LogisticRegression


# In[40]:


logmodel = LogisticRegression(solver='liblinear')
logmodel.fit(X_train,y_train)


# In[41]:
```

```
predictions = logmodel.predict(X_test)


# In[42]:


custom_classification_report(y_test, predictions, 'Relatório de Métricas Para Regressão
↪  Logistica')


# In[43]:


plot_confusion_matrix(y_test, predictions, name='images/cm_rl_adm.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))


# # Testes sem a variável `VAR07`, ela será removida por ser uma variável enviesada.

# In[44]:


df = df.drop('VAR07',axis=1)
df.head()


# Dividir os dados agora sem a variável `VAR07`

# In[45]:


X = df.drop(['EVADIU'],axis=1)
y = df['EVADIU']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=0)


# ### Testes KNN

# Escolha do parametro K novamente

# In[46]:


plot_acc(X_train, X_test, y_train, y_test, list(range(1, 203, 2)),
↪  'images/knn_neigh_adm_no_var07.svg')


# Normaliza os dados sem variável `VAR07`

# In[47]:


# Get column names first
names = X_train.columns
```

```
# Create the Scaler object
scaler = preprocessing.StandardScaler()

# Fit your data on the scaler object
X_train_scaled = scaler.fit_transform(X_train)
X_train_scaled = pd.DataFrame(X_train_scaled, columns=names)

X_test_scaled = scaler.fit_transform(X_test)
X_test_scaled = pd.DataFrame(X_test_scaled, columns=names)


# Testa o parâmetro K novamente para os dados normalizados

# In[48]:


plot_acc(X_train_scaled, X_test_scaled, y_train, y_test, list(
    range(1, 203, 2)), 'images/knn_neigh_norm_adm_no_var07.svg')


# Resultado: K = 1 e dados não normalizados

# In[49]:


knn = KNeighborsClassifier(n_neighbors=1, n_jobs=-1)
knn.fit(X_train, y_train)


# Executando e gerando os relatórios para o KNN

# In[50]:


predictions = knn.predict(X_test)


# In[51]:


custom_classification_report(y_test, predictions, 'Relatório de métricas para o KNN com dados
↪  não normalizados')


# In[52]:


plot_confusion_matrix(y_test, predictions, name='images/cm_knn_adm_no_var07.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))


# ### Testes com Árvore de Decisão

# In[53]:
```

```
treemodel = DecisionTreeClassifier(criterion='entropy')
treemodel.fit(X_train,y_train)


# In[54]:


predictions = treemodel.predict(X_test)


# In[55]:


custom_classification_report(y_test, predictions, 'Relatório de Métricas Para Árvore de
↪  Decisão')


# In[57]:


plot_confusion_matrix(y_test, predictions, name='images/cm_tree_adm_no_var07.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))


# In[58]:


export_graphviz(treemodel, out_file='images/adm_tree_no_var07.dot',
                           max_depth=3,
                           feature_names=X.columns,
                           class_names=['Não evadiu', 'Evadiu'],
                           filled=True, rounded=True,
                           special_characters=True)

dot_data = export_graphviz(treemodel, out_file=None,
                           max_depth=3,
                           feature_names=X.columns,
                           class_names=['Não evadiu', 'Evadiu'],
                           filled=True, rounded=True,
                           special_characters=True)
graph = graphviz.Source(dot_data)
graph


# #### Importância de variáveis utilizando SelectFromModel

# In[59]:


treemodel.feature_importances_


# In[60]:


treemodel = DecisionTreeClassifier(criterion='entropy')
```

```
treemodel.fit(X, y)
model = SelectFromModel(treemodel, prefit=True)


# In[61]:


X_new = model.transform(X)
X_new.shape


# ### Testes com RL

# In[62]:


logmodel = LogisticRegression(solver='liblinear')
logmodel.fit(X_train,y_train)


# In[63]:


predictions = logmodel.predict(X_test)


# In[64]:


custom_classification_report(y_test, predictions, 'Relatório de Métricas Para Regressão
↪  Logistica')


# In[65]:


plot_confusion_matrix(y_test, predictions, name='images/cm_rl_adm_no_var07.svg',
                      classes=np.array(['Evadiu', 'Não evadiu']))
```

**Fonte:** Elaborado pelo autor.