



Revista Brasileira de Computação Aplicada, Abril, 2021

DOI: 10.5335/rbca.v13i1.11378 Vol. 13, No 1, pp. 98-111

Homepage: seer.upf.br/index.php/rbca/index

ARTIGO ORIGINAL

Descoberta de conhecimento em um jogo sério para o ensino de plantas industriais: um estudo de caso Utilizando árvores de decisão

Knowledge discovery in a serious game for teaching industrial plants: a case study using decision trees

Nilo Cesar da Silva Dutra Júnior¹, Cleo Zanella Billa¹, Diana Francisca Adamatti ¹⁰,1

¹Centro de Ciências Computacionais - Universidade Federal do Rio Grande (C3/FURG)

*{nilocesarjunior, cleo.billa, dianaada}@gmail.com

Recebido: 23/07/2020. Revisado: 25/01/2021. Aceito: 05/04/2021.

Resumo

Os jogos sérios são mais uma das ferramentas que podem auxiliar no processo de ensino-aprendizagem, visto seu aspecto lúdico e motivador. Paralelamente, cada vez mais são geradas informações em bases de dados em todas as áreas do conhecimento e é preciso obter dados conclusivos, que possam gerar conhecimento. Este processo é conhecido como Knowledge Discovery in Databases - KDD, e utiliza algoritmos de aprendizado de máquina para obter informações valiosas e posteriormente interpretá-las. Este artigo apresenta um estudo relativo ao tema de descoberta de conhecimento em bases de dados utilizando um jogo sério do tipo simulador, para estudantes de ensino superior. O processo de descoberta de conhecimento utilizou a técnica de árvores de decisão e os resultados obtidos apresentaram algumas das ações/estratégias utilizadas pelos estudantes e das dificuldades por eles encontradas durante a realização do jogo.

Palavras-Chave: Plantas Industriais; Jogos Sérios; Mineração de Dados

Abstract

Serious games are one of the tool that can help in the teaching-learning process, given them playful and motivating aspect. In parallel, each day, more information is generated in databases in all areas of knowledge and it is necessary to obtain conclusive data that can generate knowledge. This process is known as Knowledge Discovery in Databases – KDD, and uses machine learning algorithms to obtain valuable information and then interpret it. This paper presents a study about knowledge discovery in databases using a serious simulator game for higher education students. The knowledge discovery process used the decision tree technique and the results obtained presented some of the actions/strategies used by the students and the difficulties encountered by them during the game.

Keywords: Industrial Plants; Serious Game; Data Mining

1 Introdução

Jogos sérios são softwares desenvolvidos com o objetivo de ensinar ou treinar pessoas simulando situações do mundo real. Ao longo do tempo, as metodologias na área de educação vêm mudando de forma que o processo de aprendizagem não fique obsoleto. Nesse cenário que os jogos sérios atuam, de uma forma prática, lúdica e divertida, eles conseguem facilitar esse processo de aprendizagem. Segundo Mattar (2010), eles tem sido utilizados com sucesso nas áreas de saúde, defesa, gerenciamento de emergência, negócios, turismo, cultura, entre outras, para fins de educação e treinamento.

Paralelamente a isso, ao decorrer dos anos, cada vez mais se tem uma quantidade grandiosa de informação em bases de dados, em todas as áreas do conhecimento. Embora seja possível analisar dados com métodos estatísticos, há relações, classificações e previsões que somente são possíveis de se descobrir utilizando mineração de dados. E é esse o principal objetivo da mineração de dados: obter conhecimento a partir de um grande volume de dados. Conforme Santos Silva (2004), as ferramentas e técnicas empregadas para análise automática e inteligente destes imensos repositórios são os objetos tratados pelo campo emergente da descoberta de conhecimento em bancos de dados (Knowledge Discovery in Databases - KDD). Mineração de dados é a etapa em KDD responsável pela seleção dos métodos a serem utilizados para localizar padrões nos dados, seguida da efetiva busca por padrões de interesse numa forma particular de representação, juntamente com a busca pelo melhor ajuste dos parâmetros do algoritmo para a tarefa em questão.

Para efetivar esse processo de descoberta de conhecimento, a mineração de dados conta com diversos algoritmos e técnicas para processar grandes volumes de dados. Dentre eles, se destacam as árvores de decisão, pois conseguem obter um rápido aprendizado de conceitos e possuem fácil implementação e interpretação. Embora a técnica de árvores de decisão seja um método de classificação, também é muito utilizada para extrair regras em conjunto com outras técnicas de mineração de dados como as técnicas de associação (Tan et al., 2009). Um dos algoritmos mais conhecidos para árvores de decisão é o C4.5 (Quilan, 1993).

O objetivo desse trabalho é apresentar algumas ações/estratégias de estudantes, utilizando como principal foco a técnica de Árvores de Decisão, mais especificamente o algoritmo J48 (uma implementação do C4.5 no sofware Weka (Hall et al., 2009)). Para isso, é analisada a base de dados obtida em testes realizados com os alunos da graduação do curso de Engenharia de Computação da Universidade Federal do Rio Grande com um jogo sobre o uso de plantas industriais, que é disciplina importante para cursos de engenharia. Uma primeira versão deste jogo foi apresentada em Barbat et al. (2015), 27 onde os primeiros testes foram realizados. Contudo, uma série de aprimoramentos foram feitos no jogo e este artigo apresenta um jogo mais completo e robusto. Contudo, o foco deste artigo está em apresentar a mineração de dados realizada, onde espera-se descobrir quais aspectos/estratégias em comum foram empregadas pelos jogadores para que obtivessem sucesso nas fases do jogo. Mas também, espera-se descobrir a relação existente entre os alunos que não obtiveram sucesso nas fases do jogo, quais os passos que eles executaram e, consequentemente, levaram ao insucesso. Segundo Manjarres et al. (2018), estudos sobre a aplicação de mineração de dados no contexto educacional se concentram em analisar os dados educacionais para prever desempenho acadêmico, evasão de alunos, extração de informação em ambientes virtuais de aprendizagem, onde as técnicas mais utilizadas são análise de correlação,

análise de agrupamento de dados, árvores de decisão, árvores de regressão, cadeias de Markov, mineração de padrão sequencial, padrões sequenciais, redes Bayesianas, rede neurais, regras de associação e regressão linear.

A escolha da técnica de árvore de decisão para o processo de descoberta de conhecimento se deve ao fato de ser uma técnica com baixo custo computacional, permitir a modelagem de dados nominais ou numéricos e principalmente, por ser um modelo do tipo "caixa branca", que permite uma explicação direta dos resultados obtidos, realizando a leitura dos ramos da árvore (Tan et al., 2009).

Na literatura não foram encontrados muitos trabalhos que relacionam as áreas que envolvem o escopo desta pesquisa: jogos sérios, plantas industriais e mineração de dados. Motzev (2010) propõem o uso de diversas técnicas de inteligência artificial, como algoritmos genéticos, redes neurais e árvores de decisão, para realização de mineração de dados de forma híbrida para obter-se uma melhor modelagem para jogos empresariais. Já em Motzev and Lemke (2015), uma extensão desta proposta é realizada, onde são incluídas mais técnicas inteligentes e também é feita a simulação do jogo. O trabalho de Bruzzone et al. (2018) apresenta um cenário de desastres em plantas industriais e como os funcionários devem proceder para minimizar os danos a saúde e meio ambiente. Alguns autores apresentam trabalhos mais abrangentes sobre o uso de jogos nos cursos de engenharia, como (Horejsi et al., 2019), (Hauge et al., 2013) e (Ross et al., 2014).

O artigo está estruturado da seguinte forma: na Seção 2 está o referencial teórico utilizado neste artigo. A Seção 3 apresenta o jogo desenvolvido e utilizado para coleta de dados. Na seção 4, todas as etapas adotadas na metodologia para descoberta do conhecimento são explicadas. A Seção 5 apresenta os resultados encontrados na base de dados, bem como sua análise. O artigo é finalizado pela Seção 6, com as conclusões e propostas de futuros trabalhos.

2 Revisão de bibliográfica

2.1 Jogos Sérios

Ao longo dos anos, é cada vez mais perceptível o importante papel dos jogos na área educacional. Com seu caráter lúdico e prático, eles são uma excelente opção para que professores atraiam, de forma efetiva, a atenção dos jovens. Mas para serem utilizados com fins educacionais, os jogos precisam ter objetivos de aprendizagem bem definidos e ensinar conteúdos das disciplinas aos usuários, ou então, promover o desenvolvimento de estratégias ou habilidades importantes para ampliar a capacidade cognitiva e intelectual dos alunos (Glazier, 2003).

Além de objetivos, resultados e desafios, um ponto muito importante que os jogos, quando usados para fins educacionais devem ter é um sistema de regras. Essas incluem recompensas e punições. Segundo Ausubel et al. (1968), as recompensas influenciam a aprendizagem da seguinte forma: servindo como incentivos, ajudam na composição de um problema efetivo por estabelecer uma relação entre uma sequência específica ou organização de atividades e um determinado objetivo resultante. Concomitantemente, por fornecer informações significativas

sobre o sucesso ou o fracasso das respostas. Já a punição (no sentido de ausência de recompensa ou fracasso em obter recompensa) age como o contrário da recompensa, o indivíduo aprende que resposta Para que se faça a simulação de uma planta industrial de um modo próximo a realidade é preciso seguir algumas regras e normas quanto à simbologia utilizada, como mostra a Fig. 1 (ver mais exemplos em (Mulley, 1993)). Assim, é necessário o uso correto dessas simbologias para uma adequada interpretação da simulação realizada e, consequentemente, do processo de aprendiz

24 leva a situação de não recompensa e deve portanto ser evitada.

2.2 Plantas Industriais

A elaboração de plantas industriais envolve procedimentos bem estruturados que compõem uma série de mecanismos pertencentes à manufatura de um ou vários itens. Os processos representados e as diferentes estruturas que compõem uma planta industrial, são amparados por normas internacionais que proporcionam alto grau de utilização em todos os níveis de produção, pois evidenciam os componentes do sistema de forma que possam ser estipuladas maneiras adequadas de desenvolvimento, proporcionando melhor aproveitamento estratégico de recursos e possibilitando análises de viabilidade técnica, viabilidade econômica, estética, ergonomia do produto e impacto ambiental (Eykhoff, 1974).

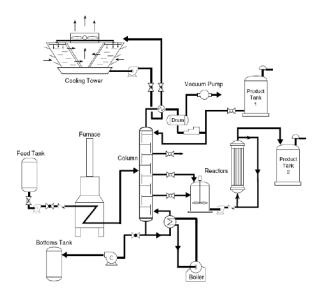


Figura 1: Exemplo de fluxograma de engenharia, com simbologia adotada na área (Eykhoff, 2008).

Para que se faça a simulação de uma planta industrial de um modo próximo a realidade é preciso seguir algumas regras e normas quanto à simbologia utilizada, como mostra a Fig. 1 (ver mais exemplos em (Mulley, 1993)). Assim, é necessário o uso correto dessas simbologias para uma adequada interpretação da simulação realizada e, conse-

quentemente, do processo de aprendizagem.

2.3 Mineração de Dados

Mineração de dados, além de ser parte integrante do processo de descoberta de conhecimento em bases de dados, é um processo responsável pela seleção dos métodos a serem utilizados para localizar padrões nos dados, seguida da efetiva busca por padrões de interesse numa forma particular de representação, juntamente com a busca pelo melhor ajuste dos parâmetros do algoritmo para a tarefa em questão (Santos Silva, 2004). Conforme Larose (2005), é foi modelada para, de uma forma inteligente, fácil e rápida, interpretar uma grande quantidade de dados e, assim, classificá-los, prevê-los ou associá-los de forma mais eficiente que os métodos estatísticos.

Segundo Santos Silva (2004), as etapas de mineração de dados utilizam técnicas e algoritmos de diferentes áreas do conhecimento, principalmente inteligência artificial (especialmente aprendizagem de máquina), banco de dados (recursos para manipular grandes bases de dados) e estatística (comumente na avaliação e validação de resultados).

2.3.1 Aprendizado de Máquina Supervisionado

O aprendizado de máquina supervisionado tem como objetivo principal, a partir de dados de treinamento, encontrar um modelo capaz de prever ou classificar um novo exemplo. Os algoritmos desse grupo possuem essa nomenclatura porque o especialista deve fornecer uma classe a qual cada dado do conjunto de treinamento pertence. Além disso, esses algoritmos podem desempenhar tarefas preditivas, ou seja, fornecem previsões ou classificações obtendo uma melhor combinação de relacionamentos entre atributos e, diferente do aprendizado de máquina não supervisionado, o atributo-alvo.

2.3.2 Classificação

Conforme Tan et al. (2009), classificação é a tarefa de aprender uma função alvo f que mapeie cada conjunto de atributos X para um dos rótulos de classes Y prédeterminadas.

Os dados de entrada, também chamados de instâncias, registros ou exemplos, são caracterizados por duas partes interdependentes: uma, trata do conjunto de atributos e a outra é o atributo-alvo. Nos algoritmos de classificação, o atributo-alvo é importante, visto que é através dele que o aprendizado do modelo é induzido e, além disso, deve ser um atributo discreto, ou seja, tem um conjunto de valores finito.

Dentre as técnicas de classificação, a escolhida para este trabalho foi a de Árvores de Decisão, que embora simples, são amplamente utilizadas. Segundo Gama (2004), as árvores de decisão utilizam a estratégia de dividir para conquistar, ou seja, um problema complexo é decomposto em sub-problemas mais simples e recursivamente esta técnica é aplicada a cada sub-problema e, além disso, aplicam a estratégia gulosa para escolha de melhores atributos.

Conforme a Fig. 2, uma árvore de decisão possui estrutura hierárquica e é formada pelos seguintes elementos:

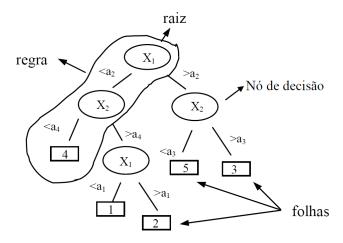


Figura 2: Representação de uma árvore de decisão Gama (2004).

- · Nodo Raiz: é o ponto de início da árvore.
- Nodo interno não-terminal: se divide em nodos filhos. Essa divisão é condicionada pelo valor do atributo analisado de modo a separar registros que possuem características distintas e, assim, cada ramo descendente possui um possível valor desse atributo.
- Nodo interno terminal: não se divide. A este nodo é atribuído uma classe. Todos os registros da base de dados utilizados caem em algum dos nós terminais e, assim, são atribuídos a uma classe.

Além disso, cada percurso da árvore de decisão, do nodo raiz até um nodo terminal, também chamado de folha, corresponde a uma regra de classificação. Essa é uma grande vantagem dessa técnica já que tais regras são do tipo seentão, facilmente compreendidas pelos usuários.

Para construir uma árvore de decisão há muitos algoritmos de indução, como ID3, C4.5 e CART, sendo o algoritmo de Hunt é a base de todos eles (Witten and Frank, 2005). De acordo com Tan et al. (2009), para entender como esse algoritmo funciona, suponha-se que Dt seja o conjunto de registros de treino que estão associados ao nodo t e y = y1,y2,...,yc sejam os rótulos das classes, o algoritmo é executado conforme os seguintes passos:

i. Se todos os registros em Dt pertencerem à mesma classe y1, então t é um nodo folha rotulado como yt.
ii. Se Dt contiver registros que percençam a mais de uma classe, uma condição de teste de atributo é selecionada para particionar os registros em subconjuntos menores. Um nodo filho é criado para cada resultado da condição de teste e os registros de Dt são distribuídos para os filhos baseados nos resultados. O algoritmo é então aplicado recursivamente a cada nodo filho.

Na construção de árvores de decisão, cada algoritmo utiliza seus próprios procedimentos para encontrar o atributo mais informativo. Assim, esse atributo é associado ao nodo interno de decisão e os dados são divididos. O algoritmo utilizado nesse trabalho é o C4.5 Quilan (1993), que na sua versão em Java no software Weka denomina-se J48,

e ele executa essa divisão dos dados por meio da entropia.

2.3.3 Ganho da Informação e a Entropia

Segundo Tan et al. (2009), as métricas desenvolvidas para selecionar a melhor divisão são muitas vezes baseadas no grau de impureza dos nodos filhos. Quanto menor o grau de impureza, mais distorcida é a distribuição das classes. Por exemplo, um nodo com uma distribuição de classe (0,1) possui impureza zero, enquanto que um nodo com distribuição de classes uniforme (0.5,0.5) possui a maior impureza. Assim, o objetivo da maior parte dos algoritmos de indução de árvores de decisão buscam reduzir ao máximo a impureza dos nodos-filhos.

O ganho da informação é uma das técnicas mais utilizadas ao escolher o melhor atributo para associá-lo ao nodo. Para minimizar a impureza dos dados e maximizar o ganho da informação do atributo, o algoritmo C4.5 calcula a entropia dos dados. A entropia caracteriza a impureza dos dados num conjunto de dados. Ela caracteriza a falta de homogeneidade dos dados de entrada em relação a sua classificação. Então, a entropia é máxima (igual a 1) quando o conjunto de dados é heterogêneo (Mitchell, 1997).

O cálculo da entropia e do ganho da informação é dado na forma como mostram as Eq. (1) e Eq. (2).

$$Entropia(S) = \sum_{n=1}^{c} -p_i log_2(p_i)$$
 (1)

Onde:

S representa um conjunto de exemplos;

C é o número de classes; e

Pi é a proporção de dados em S que percentem a classe i.

$$Ganho(S,A) = Entropia(S) - \sum \frac{S_{\nu}}{S} * Entropia(S_{\nu})$$
 (2)

Onde

S representa o total de atributos da classe e; Sv é o valor do atributo na classe.

Dessa forma, o algoritmo C4.5, para determinar a melhor divisão dos dados, seleciona o atributo que mais reduz a aleatoriedade dos dados, ou seja, com o maior ganho de informação. Pode-se observar na ?? que esse ganho da informação de cada atributo é calculado através da diferença entre a entropia total dos dados e a soma da entropia das partições.

3 Jogo Desenvolvido

O jogo proposto é do tipo simulador e apresenta componentes de uma planta industrial que devem ser utilizados para "resolver" desafios propostos. Este jogo foi desenvolvido em Java e possui três fases (mais uma fase bônus aberta), nas quais o aluno poderá manipular os seguintes componentes: tanques cilíndricos, válvulas de controle, bombas hidráulicas e dutos hidráulicos. Como parte do processo de aprendizado, no decorrer das fases, os componentes serão configurados e relacionados pelo usuário, ou

seja, atributos como vazão, diâmetro e potência serão ajustados pelo aluno de modo a simular o processo dinâmico solicitado. É importante salientar que a cada nova fase, telas explicativas sobre o problema a ser solucionado serão apresentadas. Contudo, a qualquer momento, o aluno poderá requerer uma explicação do problema proposto ou até mesmo reiniciar a fase. A principal ideia do jogo é mostrar como os conceitos teóricos de cada componente se inter-relacionam e como funciona sua dinâmica, em um ambiente simulado e seguro.

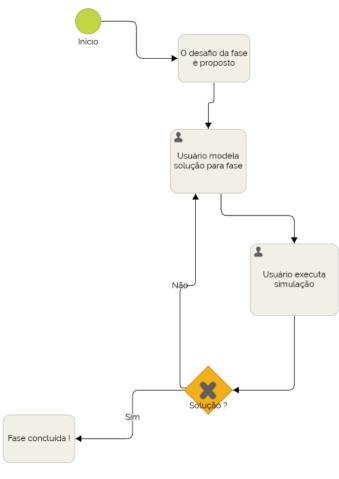


Figura 3: Fluxograma do funcionamento do jogo.

Conforme a Fig. 3, pode-se entender que o funcionamento das fases do jogo consiste em três etapas em que o jogador estará imerso:

- i. Inicialmente, será apresentado ao jogador o desafio da fase e informações quanto aos passos necessários para conclusão do mesmo. Além disso, telas explicativas são mostradas ao usuário de modo que ele saiba como executar a simulação proposta e como utilizar os componentes necessários.
- ii. Depois, o usuário deverá elaborar a planta industrial requisitada, para isso, deve utilizar, conectar e configu-

rar os componentes disponibilizados em cada fase. iii. Ao final da elaboração da planta industrial, o usuário deverá executar a simulação para concluir a solução do problema e, caso conclua a execução no tempo limite, passará à fase seguinte.

As fases do jogo, que são explicitadas pelas Fig. 4, Fig. 5, Fig. 6 e Fig. 7 e possuem um nível de dificuldade crescente, sendo a primeira fase a mais simples e a última, a mais difícil.

Primeira Fase (Fig. 4): nesse primeiro desafio encontrado pelo aluno, o objetivo é simular o processo industrial que ocorre ao passar a água, através de uma bomba hidráulica, de um tanque hidráulico a outro. O usuário deve configurar a potência da bomba hidráulica de modo a concluir esse processo dentro de um tempo limite.

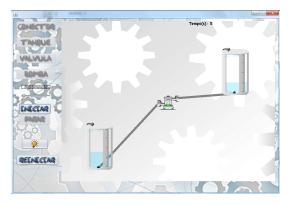


Figura 4: Primeira fase do jogo.

Segunda Fase (Fig. 5): o jogador também deve configurar a bomba hidráulica, passando a água do tanque maior para os dois menores, de modo que eles fiquem em equilíbrio.

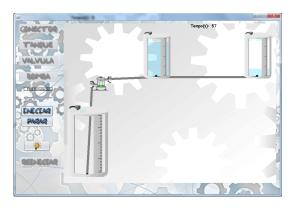


Figura 5: Segunda fase do jogo.

 Terceira Fase (Fig. 6): nesse último problema proposto ao jogador o foco são os registros hidráulicos. O jogador deve configurar a vazão de cada registro, e assim, encher o tanque de cada casa em um ordem específica. Essa ordem deve ser rigorosamente respeitada.

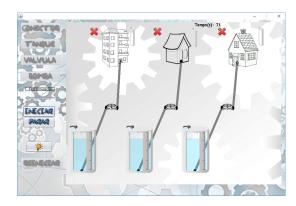


Figura 6: Terceira Fase.

 Quarta Fase (Fig. 7): é um espaço que o jogador pode montar qualquer planta industrial que ele queira. Logo, o aluno pode criar tanques, registros, bombas e conectores e simular o processo desejado.

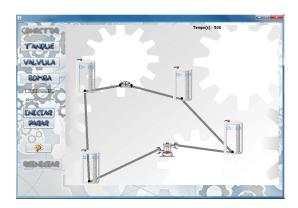


Figura 7: Quarta fase do jogo – exemplo de planta industrial montada.

Em todas as fases, antes de iniciar a simulação do processo industrial, o jogador deve sempre conectar os componentes através de dutos hidráulicos e configurá-los para concluir a fase. Ao passo que o aluno não consegue simular o processo corretamente e, assim, o tempo limite é atingido, ele deve recomeçá-la. Caso o jogador atinja o tempo limite e, assim, não conclua a fase, há três chances ao longo do jogo para recomeçar a fase. No final das três tentativas, haverá um "fim de jogo" e o usuário deve começar todo o jogo novamente, independente da fase que se encontra. Nota-se que essas tentativas por falhas são diferentes do botão reiniciar, o qual o usuário pode usar quantas vezes quiser.

O banco de dados é implementado através do PostgreSQL. A base de dados desse trabalho funciona como um "log", ou seja, um histórico das atividades do jogador em cada fase. Ao passo que o aluno configura os componentes do processo de simulação da fase, todos os parâmetros configuráveis, técnicas de jogo, botões utilizados e tempo de conclusão da fase são salvos automaticamente na base de dados.

4 Metodologia Adotada para Descoberta de Conhecimento

A Fig. 8 apresenta os processos realizados na base de dados obtida nos jogos para descoberta de conhecimento.

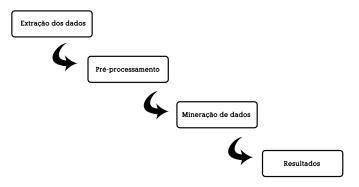


Figura 8: Processos aplicados para descoberta do conhecimento.

 Extração dos dados: Para popular a base de dados foram realizados dois testes com 44 alunos do primeiro ano de graduação da Universidade Federal do Rio Grande, no período de maio e julho de 2017. Ambos os testes tiveram a duração média de 30 minutos. Ao final dos testes, as quantidades de registros obtidas estão expostas na Tabela 1.

Tabela 1: Quantidade total de instâncias antes da etapa de pré-processamento.

Tabela	Número total de instâncias
Bomba Hidráulica	956
BotaoUsado	877
Conector	2.498
Fase	368
Registro	421
Tanque	3.053
Usuário	208

 Pré-processamento: Nessa primeira etapa de préprocessamento, foi definido a utilização de um subconjunto dos atributos salvos pelo jogo no banco. Esta escolha ocorreu pois alguns atributos possuem nenhuma ou pouca informação e poderiam gerar conflitos nos algoritmos. Esse tipo de pré-processamento chamase Seleção de Dados. A Tabela 2 indica quais atributos foram utilizados no desenvolvimento dos resultados.

Tabela	Atributo	Descrição
Conector	Diâmetro	Valor do tipo real inse- rido pelo jogador para configurar o compo- nente
Bomba Hidráulica	Potência	Valor do tipo real inse- rido pelo jogador para configurar o compo- nente
Registro	Parâmetro	Valor do tipo real inse- rido pelo jogador para configurar o compo- nente
BotaoUsado	Numero	Valor do tipo inteiro re- presentando quantas vezes o jogador clicou durante a fase
Fase	Tempo	Valor do tipo inteiro indicando em quanto tempo o jogador concluiu a fase, valores iguais a 0 indicam que o usuário não obteve sucesso na fase.

Tabela 2: Atributos escolhidos para geração do modelo

Além de selecionar os atributos certos para se utilizar no processo de mineração de dados, foi preciso, nos atributos referentes aos conectores, às bombas hidráulicas e aos cliques, combinar dois ou mais objetos em um único registro no arquivo. Esse processo, chamado de Agregação, ocorreu de duas formas: somando a quantidade de instâncias e obtendo a média de objetos. Essa combinação de atributos se justifica pois a análise de dados foi baseada em fases, ou seja, cada instância no arquivo, após o processo de agregação, representa os dados relativos a somente um registro na tabela de Fase da base de dados.

Pode-se observar na Fig. 9 que houve somente uma instância no arquivo para cada vez que a fase fosse jogada. Assim, o número de registros de conectores que pertenciam a mesma fase, ou seja, com a mesma chave estrangeira (ID_Fase) foram somados. Além disso, foi extraída a média aritmética simples dos diâmetros de cada conector da fase configurado pelo usuário. O mesmo processo foi feito para os atributos referentes à bomba hidráulica e aos cliques, pelas Fig. 10 e Fig. 11.

A Tabela 3 apresenta os atributos selecionados e agregados.

Depois de selecionar os atributos e agregá-los foi necessário a formatação e conversão para o formato que o Weka utiliza, ".arff". Por fim, quando se utiliza algoritmos de classificação, o principal requisito é que o atributo-alvo esteja em intervalos, ou seja, categorizado. Nos algoritmos de associação o requisito é mais forte: todos os atributos utilizados devem ser categóricos, caso não o sejam, deverão ser discretizados. Todos os atributos selecionados e agregados apresentavam valores contínuos, ou seja, poderiam assumir qualquer valor dentro de um intervalo. Neste trabalho, buscou-se o resultado principal utilizando árvores de decisão e foi escolhido discretizar todos os atributos, mesmo as árvores de decisão permitindo valores contínuos.



Figura 9: Agregação quanto aos atributos relacionados ao Conector.



Figura 10: Agregação quanto aos atributos relacionados à Bomba Hidráulica.



Figura 11: Agregação quanto aos atributos relacionados aos Cliques.

Tabela 3: Atributos após seleção e agregação.

Atributo	Descrição		
Qtd_Conector	Quantidade de conectores que o jogador		
	utilizou na fase, valor contínuo		
Media_Conector	Média aritmética simples de todos os valo-		
	res inseridos pelo jogador dos conectores		
	na fase, valor contínuo		
Qtd_Bomba	Quantidade de bombas hidráulicas que o		
	jogador utilizou na fase, valor contínuo		
Media_Bomba	Média aritmética simples de todos os va-		
	lores inseridos pelo jogador das bombas		
	hidráulicas na fase, valor contínuo		
Cliques	Quantidade de cliques que o usuário execu-		
	tou na fase, valor contínuo. Esses cliques		
	incluem reinicialização da fase, inicializa-		
	ção de simulação, dica e criação/alteração		
	de componentes.		
Resultado	Tempo final da fase, valor contínuo.		

O Weka, software utilizado para o desenvolvimento deste trabalho, disponibiliza duas abordagens para discretização de atributos contínuos: discretização supervisionada que utiliza informações sobre classes, e a discretização não supervisionada, aquela em que não são utilizadas informações sobre classes. A técnica empregada foi a discretização não supervisionada.

Na discretização não supervisionada, o algoritmo pode utilizar duas metodologias, que produzem resultados, muitas vezes, completamente diferentes: largura igual, que divide a faixa dos atributos em intervalos de mesma largura, e frequência igual, que divide os dados em intervalos contendo aproximadamente o mesmo número de objetos em cada intervalo. Embora as duas técnicas sejam amplamente utilizadas, a discretização não supervisionada de largura igual pode resultar em a maioria dos dados se concentrando em um intervalo prejudicando o modelo de classificação. Com todos os atributos utilizados neste trabalho esse fato ocorreu, ou seja, a maioria dos registros se concentraram em apenas um intervalo.

Sendo assim, a metologia utilizada foi discretização não supervisionada de frequência igual. Os atributos foram discretizados nos seguintes intervalos:

Atributo Qtd_Conector:

- Intervalo qtd_conector_inalterado: representa os registros em que não houve nenhuma alteração de conectores na fase, ou seja, o usuário não utilizou nenhum conector.
- Intervalo qtd_conector_baixo: representa os registros em que houve uma quantidade baixa de conectores na fase.
- Intervalo qtd_conector_medio: representa os registros em que houve uma quantidade média de conectores na fase.
- Intervalo qtd_conector_alto: representa os registros em que houve uma quantidade alta de conectores na fase

· Atributo Media_Conector:

 Intervalo media_conector_inalterado: representa os registros em que não houve nenhuma alteração no

- diâmetro do conector permanecendo com o valor inicial de 10 centímetros.
- Intervalo media_conector_baixo: representa os registros em que houve valores baixos no diâmetro do conector na fase.
- Intervalo media_conector_medio: representa os registros em que houve valores médios no diâmetro do conector na fase.
- Intervalo media_conector_alto: representa os registros em que houve valores altos no diâmetro do conector na fase.

· Atributo Qtd Bomba:

- Intervalo qtd_bomba_baixo: representa os registros em que houve uma quantidade baixa de bombas hidráulicas na fase.
- Intervalo qtd_bomba_medio: representa os registros em que houve uma quantidade média de bombas hidráulicas na fase.
- Intervalo qtd_bomba_alto: representa os registros em que houve uma quantidade alta de bombas hidráulicas na fase.

· Atributo Media Bomba:

- Intervalo media_bomba_inalterado: representa os registros em que não houve nenhuma alteração na potência da bomba hidráulica permanecendo com o valor 0.
- Intervalo media_bomba_baixo: representa os registros em que houve valores baixos na potência da bomba hidráulica na fase.
- Intervalo media_bomba_medio: representa os registros em que houve valores médios na potência da bomba hidráulica na fase.
- Intervalo media_bomba_alto: representa os registros em que houve valores altos na potência da bomba hidráulica na fase.

· Atributo Cliques:

- Intervalo clique_baixo: representa os registros em que houve uma quantidade baixa de cliques na fase.
- Intervalo clique_medio: representa os registros em que houve uma quantidade média de cliques na fase.
- Intervalo clique_alto: representa os registros em que houve uma quantidade alta de cliques hidráulicas na fase.

· Atributo Resultado:

- Intervalo perdeu: representa os registros em que o jogador não obteve sucesso na fase.
- Intervalo ganhou: representa os registros em que o jogador obteve sucesso na fase.

Os intervalos de cada resultado de cada atributo podem ser visualizados nas Tabela 4 e Tabela 5:

Na terceira fase, além do conector, outro componente utilizado na simulação é o registro hidráulico, conforme Tabela 6.

Ao final da etapa de pré-processamento foi gerado o seguinte resultado:

Tabela 4: Intervalos de valores dos atributos na primeira fase.

	Valores dos intervalos			
Atributo	Inalt.	Baixo	Medio	Alto
Qtd_Conector	0	[1,2]	(2,5]	(5,+inf)
Media_Conector	10	(10,24]	(24,110]	(110,+inf)
Qtd_Bomba	Não	1	2	(2,+inf)
Media_Bomba	0	(0,4]	(4,27]	(27,+inf)
Cliques	Não	[1,4]	(4,7]	(7,+inf)

Tabela 5: Intervalos de valores dos atributos na segunda fase.

	Valores dos intervalos			
Atributo	Inalt.	Baixo	Medio	Alto
Qtd_Conector	0	[1,3]	(3,8]	(8,+inf)
Media_Conector	10	(10,15]	(15,87]	(87,+inf)
Qtd_Bomba	Não	1	(1,3]	(3,+inf)
Media_Bomba	0	[1,5]	(5,60]	(60,+inf)
Cliques	Não	[1,3]	(3,10]	(10,+inf)

Tabela 6: Intervalos de valores dos atributos na terceira fase.

	Valores dos intervalos			
Atributo	Inalt.	Baixo	Medio	Alto
Qtd_Conector	0	[1,11]	(11,22]	(22,+inf)
Media_Conector	10	(-inf,8]	9∪[11,20]	(20,+inf)
Qtd_Registro	Não	(-inf,7]	(7,10]	(10,+inf)
Media_Registro	0	(0,0.23]	(0.23,0.39]	(0.39,+inf)
Cliques	Não	[0,10]	(10,19]	(19,+inf)

Primeira fase: 186 instâncias;
Segunda fase: 96 instâncias;
Terceira fase: 41 instâncias.

5 Análise dos Resultados

As árvores de decisão apresentadas são referentes a primeira e a segunda fase do jogo. Não foi possível obter o modelo de classificação utilizando árvores de decisão da terceira fase porque, além de haver poucas instâncias, menos estudantes ainda concluíram a fase.

Os resultados são apresentados a partir do atributo-alvo discretizado e numérico em perdeu e ganhou.

5.1 Atributo-alvo Discretizado

5.1.1 Primeira Fase

A Fig. 12 representa a árvore de decisão extraída.

Pode-se observar que, de acordo com o modelo gerado, em relação as regras referentes aos usuários ganhadores, indicados com retângulos azuis, tem-se que:

Se quantidade de conectores utilizados na simulação
 baixa e a média na potência da bomba hidráulica =
 baixa ou intermediária ou alta, então

Resultado = ganhou (regras 4, 5 e 6);

 Se quantidade de conectores utilizados na simulação = intermediária e a média na potência da bomba hidráulica = baixa, então Resultado = ganhou (regra 8);

 Se quantidade de conectores utilizados na simulação = intermediária e a média na potência da bomba hidráulica = alta e

a quantidade de cliques = baixa ou intermediária, então **Resultado** = ganhou (regras 10 e 11).

Conforme a árvore de decisão, foram encontradas 6 regras para o atributo alvo. De um modo geral, segundo o modelo, os jogadores conseguiram completar a fase se utilizaram quantidades baixas ou intermediárias de conectores e ainda que atribuindo para a bomba hidráulica qualquer valor de potência maior que o. Além disso, podese observar a regra 11, que classificou corretamente todos os registros, sendo no caso do atributo alvo ganhou o ramo da árvore com maior acerto.

Pode-se observar que, de acordo com o modelo gerado, em relação as regras referentes aos usuários perdedores, indicados com retângulos vermelhos, tem-se que:

 Se quantidade de conectores utilizados na simulação = inalterada ou alta, então

Resultado = perdeu (regras 1 e 2);

 Se quantidade de conectores utilizados na simulação = baixa e

a média na potência da bomba hidráulica = inalterada, então

Resultado = perdeu (regra 3);

 Se quantidade de conectores utilizados na simulação = intermediária e

a média na potência da bomba hidráulica = inalterada ou intermediária, então

Resultado = perdeu (regras 7 e 9);

 Se quantidade de conectores utilizados na simulação = intermediária e

a média na potência da bomba hidráulica = alta e a quantidade de cliques = alta, então **Resultado** = perdeu (regra 12).

Em relação às regras referentes aos registros dos usuários que perderam nas fases, o modelo encontrou 6 regras, das quais pode-se extrair informações interessantes. As que menos podem ser contestadas são as de valores inalterados, já que para executar a simulação das plantas industriais, em cada fase, o usuário deve utilizar os conectores para ligar cada componente e depois, em cada fase, também deve inserir um valor na potência da bomba acima de o. Se alguma dessas duas ações não forem executadas, ou seja, alguma permanecer inalteradas, o resultado da fase será diretamente perdeu. Nesses ramos da árvores todos os registros foram classificados corretamente. Referente as altas quantidades de conectores ou cliques, o modelo classificou 14 instâncias de 73 incorretamente. Pode-se inferir que para esses usuários, ou o tempo não foi suficiente para realizar a simulação, ou não foi assimilado o objetivo da fase, sendo que o segundo parece fazer mais sentido.

Deve-se atentar para a regra 9, referente a quantidade de conectores intermediária e valores de potência também intermediários nas bombas hidráulicas, que de 13 registros, 5 foram classificados incorretamente, o que indica que esse ramo da árvore chegou ao erro de 38%, sendo a mais incerta do modelo.

Na Fig. 13, as características da árvore de decisão ge-

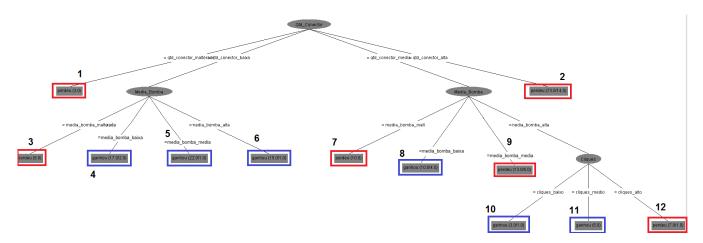


Figura 12: Árvore de decisão da primeira fase discretizada

rada são apresentadas. Pode-se concluir nesse modelo que embora tem uma boa acurácia (82,25% de acerto - de 153 instâncias, 33 foram classificadas de forma incorreta) para classificar instâncias, houve muitos erros da árvore gerada em classificar corretamente os registros referentes aos jogadores ganhadores. Fato que pode ser ratificado pela matriz de confusão, que, de 83 instâncias, 24 foram classificadas incorretamente. Já nos registros que deveriam ser rotulados como perdeu, de 103 instâncias, 9 foram classificadas incorretamente. Assim, essa árvore de decisão gerada encontrou um bom padrão de classificação e regras.

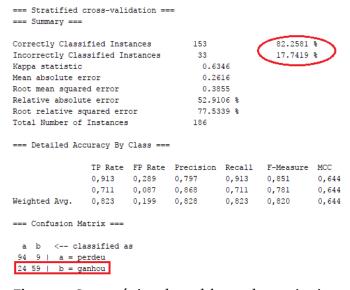


Figura 13: Características do modelo gerado na primeira fase discretizada.

5.1.2 Segunda Fase

A Fig. 14 representa a árvore de decisão extraída.

Pode-se observar que, de acordo com o modelo gerado, em relação as regras referentes aos usuários ganhadores, indicados com retângulos azuis, tem-se que:

- Se quantidade de conectores utilizados na simulação = baixa, então
 - Resultado = ganhou (regra 2)
- Se quantidade de conectores utilizados na simulação = intermediária e
- a média na potência da bomba hidráulica = intermediária, então

Resultado = ganhou (regra 6)

A segunda fase, além de haver menos instâncias que a primeira, também tem menos registros rotulados como ganhadores, o que justifica o menor número de regras se comparadas com o primeiro modelo gerado. Mas também possui o ramo da árvore, regra 2, com menor acurácia, de 31 registros, 8 foram classificados incorretamente, resultando em um percentual de 25,8% de erro.

Pode-se observar que, de acordo com o modelo gerado, em relação as regras referentes aos usuários perdedores, indicados com retângulos vermelhos, tem-se que:

- Se quantidade de conectores utilizados na simulação = inalterada ou alta, então
 - **Resultado** = perdeu (regras 1 e 3)
- Se quantidade de conectores utilizados na simulação = intermediária e
 - a média na potência da bomba hidráulica = inalterada, baixa ou alta, então

Resultado = perdeu (regras 4,5 e 7)

Ao contrário da primeira fase, a árvore de decisão da segunda fase teve muito mais folhas com classes rotuladas como perdeu do que rotuladas como ganhou: 2 folhas rotuladas como ganhou e 5 rotuladas como perdeu. Isso indica uma dificuldade maior nessa fase por parte dos jogadores. Em contrapartida, percebe-se uma melhoria significativa nas estratégias utilizadas pelos jogadores e, consequentemente, na compreensão do jogo, enquanto que no modelo gerado na primeira fase (Fig. 12) haviam 11 registros em que os usuários, ou não utilizaram conectores, ou não inse-

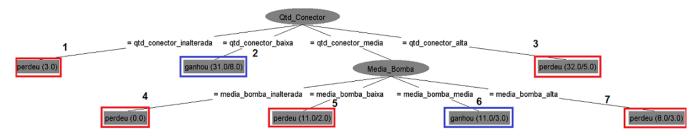


Figura 14: Árvore de decisão da segunda fase discretizada.

riram nenhum valor para a bomba (regras 1 e 3 da Fig. 12), na segunda fase houveram somente 3 registros que não utilizaram conectores e nenhum que não inseriu valores para a bomba.

Na Fig. 15 são apresentadas características do modelo gerado. Já na segunda fase, a acurácia do modelo não tão foi boa (63,5% de acerto - de 86 instâncias, 35 foram classificadas de forma incorreta), e, além disso, o modelo também teve mais dificuldades em classificar corretamente os jogadores que obtiveram sucesso na fase. Fato que pode ser ratificado pela matriz de confusão, que, de 41 instâncias, 19 foram classificadas incorretamente, quase a metade. Outrossim, ao contrário do primeiro modelo gerado (Fig. 12), o algoritmo não considerou o atributo Clique como importante e não o utilizou na árvore de decisão.

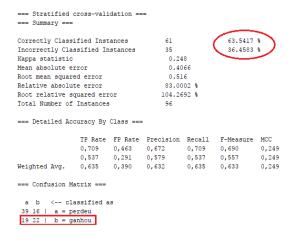


Figura 15: Características do modelo gerado na segunda fase discretizada

5.2 Atributo-alvo Numérico

Os próximos dois modelos gerados são diferentes dos vistos até agora no fato de que os atributos utilizados são numéricos, ou seja, as regras obtidas são comparando números exatos e não em intervalos.

5.2.1 Primeira Fase

A Fig. 16 representa a árvore de decisão extraída.

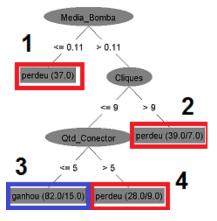


Figura 16: Árvore de decisão da primeira fase numérica.

Em comparação com o primeiro modelo gerado (primeira fase mas com atributos discretizados - Fig. 12), este, além de ser muito menor, considerou o atributo Media_Bomba como raiz e o atributo Qtd_Conector no último nível da árvore. Além disso, 3 regras rotuladas como perdeu e somente 1 regra como ganhou.

Pode-se observar que, de acordo com o modelo gerado, em relação as regras referentes aos usuários ganhadores, indicados com retângulos azuis, tem-se que:

 Se média na potência da bomba hidráulica > 0.11 e a quantidade de cliques <= 9, e a quantidade de conectores utilizados na simulação <= 5, então Resultado = ganhou (regra 3)

Foi proposto analisar tanto os modelos com atributos discretizados quanto numéricos para comparar o tipo de regra e conhecimento gerado. Na regra 3, pode-se observar que a média necessária na potência da bomba hidráulica para que o usuário completasse a fase foi maior que 0.11, mas o quanto maior que 0.11 não é possível saber através desse modelo.

Pode-se observar que, de acordo com o modelo gerado, em relação as regras referentes aos usuários perdedores, indicados com retângulos vermelhos, tem-se que:

i. Se média na potência da bomba hidráulica <= 0.11, então

Resultado = perdeu (regra 1)

ii. Se média na potência da bomba hidráulica > 0.11 e a quantidade de cliques > 9, então

Resultado = perdeu (regra 2)

iii. Se média na potência da bomba hidráulica > 0.11 e a quantidade de cliques <= 9 e

a quantidade de conectores utilizados na simulação > 5, então

Resultado = perdeu (regra 4)

Enquanto que no primeiro modelo (Fig. 12) os valores inalterados levaram diretamente ao insucesso na fase, neste modelo, que tem acurácia de 100%, o conhecimento descoberto foi diferente. Aqui, valores com médias na potência da bomba hidráulica menores ou iguais a 0.11 foram decisivos para este fato (regra 1), o que mostra uma dificuldade de parte dos jogadores em interpretar o papel da bomba hidráulica na simulação da planta industrial requisitada. Deve-se notar também que a regra 4 foi a mais incerta do modelo, com 32,1% de erro.

Na Fig. 17 apresenta as características do modelo gerado.

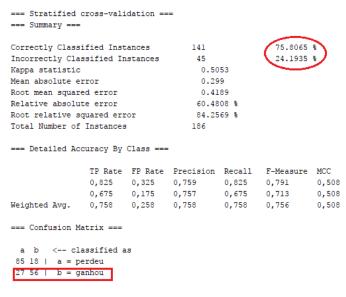


Figura 17: Características do modelo gerado na primeira fase numérica.

Na primeira fase, mas com os atributos numéricos, pode-se observar que a acurácia do modelo foi razoável (75 % - de 186 instâncias, 45 foram classificadas de forma incorreta). Além disso, o modelo, como os anteriores, teve mais dificuldades para classificar corretamente os jogadores que obtiveram sucesso na fase. Fato que pode ser ratificado pela matriz de confusão, que de 83 instâncias, 27 foram classificadas incorretamente, praticamente um terço das instâncias.

5.2.2 Segunda Fase

A Fig. 18 representa a árvore de decisão extraída. Nos modelos gerados, esse foi o mais distinto e com acurácia baixa. Diferentemente dos anteriores, o algoritmo elegeu como

nodo raiz o atributo Cliques.

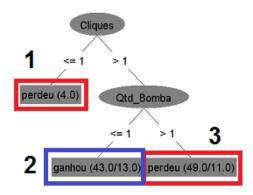


Figura 18: Árvore de decisão da segunda fase numérica.

Pode-se observar que, de acordo com o modelo gerado, em relação as regras referentes aos usuários ganhadores, indicados com o retângulo azul, tem-se que:

Se quantidade de cliques > 1 e

 a quantidade de bombas hidráulicas utilizadas na simulação <= 1, então
 Resultado = ganhou (regra 2)

Além de ser a única regra gerada para o resultado ganhou também foi o ramo que errou mais, 30,23%. Essa regra diz que os usuários, para ganhar, usaram uma bomba hidráulica apenas (a fase já começa com uma, então não é possível não usar nenhuma). Embora essa regra possua um alto erro em comparação as outras, é interessante constatar que apenas uma bomba foi suficiente para grande parte dos jogadores concluir a fase, o que indica uma melhora na percepção da função da bomba hidráulica na fase e seu funcionamento.

Pode-se observar que, de acordo com o modelo gerado, em relação as regras referentes aos usuários perdedores, indicados com retângulos vermelhos, tem-se que:

- Se quantidade de cliques <= 1, então Resultado = perdeu (regra 1)
- Se quantidade de cliques > 1 e

 a quantidade de bombas hidráulicas utilizadas na simulação > 1, então

Resultado = perdeu (regra 3)

As regras dessa árvore de decisão, diferente das outras, não levou em consideração o atributo Qtd_Conector, o que é inesperado, visto que antes de configurar os componentes o jogador deve conectá-los. Além disso, considerou o atributo Qtd_Bomba também, ao contrário dos demais. Reitera-se novamente o ganho de conhecimento e informações diferentes com relação as estratégias empregadas pelos jogadores, que é o foco deste trabalho. A primeira regra, sem erros, demonstra que os jogadores que não executaram cliques ou o fizeram apenas uma vez perderam. Tal informação já era esperada, já que para configurar qualquer componente e iniciar a simulação deve-se executar

no mínimo 2 cliques. Logo, esses jogadores não tentaram concluí-la. Além disso, os jogadores que utilizaram muitas bombas hidráulicas para chegar na simulação correta, provenientes de reinicialização de fase, também não chegaram ao resultado esperado. De um modo geral, os modelos gerados para a segunda fase tiveram menos acurácia e mais dificuldade por parte dos alunos em concluí-la.

Na Fig. 19 apresenta as características do modelo gerado. Na primeira fase com os atributos numéricos (Fig. 19), a acurácia do modelo não foi melhor que as demais (66,66% de acerto - de 96 instâncias, 32 foram classificadas de forma incorreta), e, além disso, o modelo, diferentemente dos anteriores, teve mais dificuldades para classificar corretamente os jogadores que não obtiveram sucesso na fase. Fato que pode ser ratificado pela matriz de confusão, que, de 55 instâncias, 20 foram classificadas incorretamente.

```
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances
                                                          66.6667 %
                                        64
                                                          33.3333 %
Incorrectly Classified Instances
                                        32
Kappa statistic
                                         0.3354
Mean absolute error
                                         0.4085
Root mean squared error
                                         0.4925
Relative absolute error
                                        83.3826 %
Root relative squared error
                                        99.5187 %
Total Number of Instances
=== Detailed Accuracy By Class ===
                 TP Rate FP Rate Precision Recall
                                                       F-Measure MCC
                 0.636
                         0,293 0,745
                                              0.636
                                                        0,686
                                                                   0.340
                 0,707
                          0,364
                                   0,592
                                              0,707
                                                        0,644
                                                                   0,340
Weighted Avg.
                 0,667
                          0.323
                                   0,679
                                              0.667
                                                        0.668
                                                                   0.340
=== Confusion Matrix ===
  a b <-- classified as
35 20 | a = perdeu
12 29 | b = ganhou
```

Figura 19: Características do modelo gerado na segunda fase numérica.

Esse último modelo gerado foi o mais distinto de todos os outros por ter mais dificuldades em rotular corretamente os registros que deveriam se classificados em perdeu. Além disso, comparando os modelos numéricos e discretizados pode-se constatar que o número de regras é bem diferente, os modelos numéricos obtiveram menos regras. Outro aspecto importante está relacionado ao atributo quantidade de cliques, que nos modelos discretizados estava presente apenas em três regras. Já nos modelos numéricos, este atributo estava presente em 6 das 7 regras.

De um modo geral, após a utilização da técnica de árvores de decisão sobre a base de dados, percebe-se o uso de todos os atributos pelos jogadores, tanto em estratégias para ganhar como para perder o jogo. Contudo, em todos os modelos gerados não foi considerado como decisivo o atributo Media_Conector. Assim, constata-se que a informação desse atributo não foi importante nesses registros para influenciar no resultado da fase.

Conclusões e Trabalhos Futuros

Em um tempo em que a internet e as mídias estão cada vez mais atraindo a atenção dos jovens, as salas de aula no mundo acadêmico estão se tornando cada vez mais monótonas para os alunos. A missão de cada professor em motivar seus alunos ao estudo e ao aprendizado nunca foi uma tarefa simples, mas está cada vez mais desafiadora. É nesse contexto que os jogos sérios se destacam, com recursos dinâmicos e imersivos, podendo auxiliar no processo de aprendizagem de uma forma lúdica e divertida.

Da mesma forma, tanto em áreas acadêmicas como corporativas se tem um grande volume de dados gerado a cada dia. A necessidade de se obter conhecimento de uma forma inteligente e efetiva evolui cada vez mais, já que as técnicas e ferramentas convencionais de busca e extração de conhecimento já não cumprem mais seu papel. Nesse sentido que a mineração de dados, um dos processos do KDD, atua de forma rápida e eficiente, disponibiliza vários algoritmos inteligentes objetivando a descoberta de conhecimento. Dentre esse algoritmos se destaca a técnica de árvores de decisão, que embora seja uma técnica simples, é um dos métodos mais usados e práticos por ser capaz de lidar com qualquer tipo de dados e ser de fácil compreensão e interpretabilidade.

O objetivo deste trabalho foi, através da técnica de árvores de decisão, extrair conhecimento de uma base de dados populada por alunos do primeiro ano do curso de engenharia de computação da Universidade Federal do Rio Grande. Esse conhecimento adquirido buscou entender quais ações/estratégias eram utilizadas pelos jogadores. Retomando-se as motivações do trabalho, foi observado o que os alunos que obtiveram sucesso (e que perderam) têm em comum e em relação aos seus desempenhos, quais jogadas foram predominantes para um bom ou mau desempenho na fase. Existe uma relação ao componente conector, pois constatou-se que os jogadores que não concluíram a fase não utilizaram conectores ou os utilizaram muito (talvez aqui um problema de interface). Além disso, um importante conhecimento adquirido foi que os jogadores tiveram uma dificuldade inicial em manipular os conectores necessários para simulação, mas da segunda fase em diante tal dúvida foi solucionada. Já em relação as diferenças nas estratégias utilizadas pelos jogadores diferenciando os desempenhos, os resultados dos modelos não foram conclusivos.

Não estava no escopo deste trabalho uma avaliação da usabilidade do jogo desenvolvido, nem a avaliação qualitativa do jogo pelos alunos. Contudo, é importante ressaltar que a versão do jogo que é apresentada neste artigo é a versão estável e que passou por diversos testes anteriores com diversas turmas, que deram feedback quanto a melhorias e erros de funcionamento. Os resultados aqui apresentados são apenas dos testes com o jogo consolidado e com dados completos.

Como trabalhos futuros, no processo de descoberta de conhecimento, pretende-se reavaliar os dados através de outras técnicas de mineração de dados, assim como analisá-las junto a um especialista. Para que o conhecimento se consolide de uma forma mais efetiva também será necessário realizar mais testes em turmas de alunos de cursos diversos de engenharia, para adquirir um volume maior de dados e modelos com maiores taxas de acerto. Além disso, em relação ao jogo será necessário adicionar mais componentes de plantas industriais para simular processos mais complexos.

Referências

- Ausubel, P. E., Novak, J. D. and Hanesian, H. (1968). *Psicologia Educacional*, Vol. 2, Holt, Rinehart and Winston, Inc.
- Barbat, M. M., Dutra, N. C., Adamatti, D. F. and Werhli, A. V. (2015). Teaching industrial plant using serious games, Bulletin of the IEEE Technical Committee on Learning Technology 17(4): 10-13. Disponível em http://tc.computer.org/tclt/wp-content/uploads/sites/5/2018/01/IEEE-TCLT-December2015.pdf.
- Bruzzone, A. G., Massei, M. and Di Matteo, R. (2018). Modeling, interoperable simulation and serious games (ms2g) for healthcare and first responders in disasters within industrial plants, *Proceedings of the 50th Computer Simulation Conference*, SummerSim '18, Society for Computer Simulation International, San Diego, CA, USA. Disponível em https://dl.acm.org/doi/pdf/10.5555/3275382.3275404.
- Eykhoff, P. (1974). System identification: parameter and state estimation.
- Eykhoff, P. (2008). Instrumentation Lessons In Industrial Instrumentation, Vol. 0.2.
- Gama, J. A. (2004). Functional trees, *Mach Learn* **55**: 219–250. https://doi.org/10.1023/B:MACH. 0000027782.67192.13.
- Glazier, R. (2003). The impact of digital games in education
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009). The weka data mining software: An update, *SIGKDD Explor. Newsl.* **11**(1): 10–18. https://doi.org/10.1145/1656274.1656278.
- Hauge, J. M. B., Pourabdollahian, B. and Riedel, J. C. K. H. (2013). The use of serious games in the education of engineers, in C. Emmanouilidis, M. Taisch and D. Kiritsis (eds), Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services, Vol. 397, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 622–629. https://doi.org/10.1007/978-3-642-40352-1_78.
- Horejsi, P., Vysata, J., Rohlikova, L., Polcar, J. and Gregor, M. (2019). Serious games in mechanical engineering education, *Research & Innovation Forum* 2019. https://doi.org/10.1007/978-3-030-30809-4_6.
- Larose, D. T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley and Sons, Inc.
- Manjarres, A. V., Sandoval, L. G. M. and Suárez, M. S. (2018). Data mining techniques applied in educational environments: Literature review, *Digital Education Review* (33): 235–266. https://doi.org/10.1344/der. 2018.33.235-266.

- Mattar, J. (2010). Games em educação: como os nativos digitais aprendem, Vol. 1, São Paulo: Pearson Prentice Hall.
- Mitchell, T. M. (1997). *Machine Learning*, Vol. 1, McGraw-Hill.
- Motzev, M. (2010). Intelligent techniques in business games and simulations a hybrid approach, *International Simulation and Gaming Association 41st Annual Conference*, Spokane WA, pp. 81–86.
- Motzev, M. and Lemke, F. (2015). Self-organizing data mining techniques in model based simulation games for business training and education, *Vanguard Scientific Instruments in Management* 11(1).
- Mulley, R. (1993). *Control System Documentation Applying Symbols and Identification*, Vol. 2, ISA The Instrumentation, Systems, and Automation Society.
- Quilan, J. R. (1993). *C4.5: Programs for Machine Learning*, Vol. 1, Morgan-Kaufmann Publishers, San Mateo, CA.
- Ross, A. M., Fitzgerald, M. E. and Rhodes, D. H. (2014). Game-based learning for systems engineering concepts, *Procedia Computer Science* **28**: 430 440. 2014 Conference on Systems Engineering Research.
- Santos Silva, M. P. (2004). Mineração de dados conceitos, aplicações e experimentos com weka. Disponível em http://www.lbd.dcc.ufmg.br/colecoes/erirjes/2004/004.pdf.
- Tan, P.-N., Steinbach, M. and Kumar, V. (2009). *Introdução ao Data Mining*, Vol. 1, Ciência Moderna.
- Witten, I. H. and Frank, E. (2005). DATA MINING Practical Machine Learning Tools and Techniques, Vol. 2.