Relatório Técnico
sobre Árvore de Decisão

Thiago M. de Sousa *Luana G. B. Martins*
Ruan C. Rodrigues

Technical Report - RT-INF_000-19 - Relatório Técnico
July - 2019 - Julho

Instituto de Informática
Universidade Federal de Goiás
www.inf.ufg.br

Relatório Técnico sobre Árvore de Decisão

Thiago M. de Sousa
thiagomontelesofc@gmail.com

Luana G. B. Martins
luanagbmartins@gmail.com

Ruan C. Rodrigues
ruanchaves93@gmail.com

Abstract. *This report describes what were the decisions, solving process and results of the proposed problem in the field of Artificial Intelligence. The challenge of inducing a decision tree that correlates with performance in the first two periods of the course with the final performance was introduced and allows to make predictions about the final performance of new students.*

Keywords: Technical Report, Decision Tree.

Resumo. *Este relatório descreve quais foram as decisões, processo de resolução e resultados do problema proposto na matéria de Inteligência Artificial. Foi introduzido o desafio de induzir uma árvore de decisão que tenha uma correlação com o desempenho nos dois primeiros períodos do curso com o desempenho final e permite fazer previsões sobre o desempenho final de novos alunos.*

Palavras-Chave: Relatório Técnico, Árvore de Decisão.

1 Introdução

Este Relatório Técnico consiste na documentação de uma estratégia tomada para se obter um processo de tomada de decisões, buscando resolver o problema de se criar um modelo para obter previsões de desempenho final de novos alunos no Bacharelado em Ciência da Computação da Universidade Federal de Goiás, tendo em mãos apenas o histórico realizado nas matérias pertinentes aos dois primeiros períodos do curso.

O processo de decisão se deu por meio da utilização de árvores de decisão, as quais por sua vez tiram proveito de uma estrutura de dados com informações que levam o processo à sua conclusão através da criação de regras, que chegam a um lista de condições para uma decisão final.

O problema consiste, em um primeiro momento, utilizar de dados de alunos que já concluíram o curso para definir parâmetros como período de ingresso, ano de conclusão e notas dos dois primeiros semestres para, em um segundo momento, criar um modelo que possa ser utilizado nas árvores de decisão, com o objetivo final de gerar previsões de desempenho de novos estudantes do curso com base em seu desempenho nos dois primeiros períodos.

No restante deste documento estão definidas a forma abordada na base de dados (Seção 2), descrição geral da solução proposta contendo nela a descrição geral do modelo utilizado e dos dados selecionados para o modelo (Seção 3), dos resultados obtidos (Seção 4), das propostas para como utilizar os resultados obtidos (Seção 5), conclusões finais (Seção 6) e referências.

2 Descrição da base de dados

Os dados consistem em um arquivo no formato csv (Comma-separated values) que é representado por uma matriz de 22361 linhas por 66 colunas, onde existe em cada coluna um determinado atributo referente a relação de um aluno com as disciplinas que cursou durante os anos e períodos.

Os atributos nas colunas contém dados referente ao aluno e sua passagem no curso. Dados como ano de nascimento, idade de ingresso à universidade e nota do Enem são exemplos de dados relacionados ao aluno anteriormente ao ingresso na faculdade. Já atributos como quantidade de reprovações, média global e ano de conclusão estão ligados ao aluno após a entrada na universidade.

Devido às restrições da definição de aluno de bom desempenho fornecidas pelo problema, alunos que ingressaram após o primeiro semestre de 2015 foram desconsiderados na etapa de treinamento.

3 Descrição da solução

Com o objetivo de permitir fazer previsões sobre o desempenho final de novos alunos, foram considerados alguns aspectos como a criação de uma classificação de bom aluno , treinamento do modelo proposto utilizando disciplinas dos dois primeiros períodos do curso e a utilização algoritmos de tomada de decisão C4.5.

3.1 Descrição do modelo utilizado

3.1.1 Estrutura do modelo de árvore de decisão

Árvores de decisão são modelos que utilizam um treinamento supervisionado para a classificação e previsão de dados. [1]

Primeiro, o conjunto de treinamento, onde as classificações dos exemplos são conhecidos, é utilizado para construir uma árvore de decisão. Após a construção, esse classificador pode ser aplicado para predizer os rótulos das classes dos exemplos do conjunto de teste (exemplos de classes desconhecidas). [4]

Nesse modelo, o conhecimento adquirido é representado por meio de regras. Sua estrutura é definida por: Nós folhas, indicando uma classe; Nós de decisão, que definem algum teste sobre o valor de um atributo específico, com um ramo e sub-árvore para cada um dos valores possíveis do teste. [2]

Para esse trabalho, as perguntas são constituídas de respostas do tipo Verdadeiro ou Falso, criando-se assim uma árvore binária de decisão. Um limiar é estabelecido para divisão dos exemplos de forma binária: isto é, aqueles exemplos que possuem o valor do atributo maior ou igual que o limiar e aqueles cujo valor do atributo é menor que o limiar estabelecido.

Para determinar o quão boa é a condição de teste realizada, ou seja, se o atributo escolhido irá resultar em um maior ganho de informação dentre todos os atributos testados, compara-se o grau de entropia do nó antes da divisão e dos nós gerados após a divisão. O atributo que gerar uma maior diferença é escolhido como condição para teste.

Durante o processo de geração da árvore, a condição de parada é atingida quando observa-se que não há ganho de informação significativo.

3.1.2 Algoritmo de classificação C4.5

C4.5 é um algoritmo do tipo de árvore de decisão que foi inicialmente desenvolvido por Ross Quinlan, e é utilizado como um algoritmo classificador. Trata-se de uma evolução do algoritmo ID3, também feito pelo mesmo autor.

Algumas características do algoritmo usado foi a denominação de um limiar de separação que representa um valor que será usado para verificar a condição. Objetivo é escolher o atributo que melhor divida o conjunto de dados, e para isso é realizado uma busca gulosa sobre todos os atributos. Para determinar a qualidade da condição realizada olha-se o ganho de informação obtido com aquele dado atributo. O ganho de informação é descrito abaixo:

Para um conjunto S de exemplos temos que:

$$Entropia(S) \equiv -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus}$$

$$Ganho(S, A) \equiv Entropia(S) - \sum_{v \in val(A)} p(A_v) * Entropia(A_v)$$

O cálculo do ganho consiste em comparar o grau de entropia do nó anterior a divisão e dos nós gerados posterior a divisão, o atributo que gerar o maior resultado, ou seja, o maior ganho de informação, é escolhido para divisão.

Seu processo de classificação transcorre da seguinte maneira:

1. Inicialmente, verifica-se os casos básicos.
2. Para cada atributo selecionado é encontrada uma razão de ganho de informação normalizada pelo partição do atributo.
3. É verificado o atributo com maior ganho de informação normalizado.
4. Cria-se um nó de decisão feito no conjunto de dados que apresentou o maior ganho de informação.
5. Faz-se a repetição dos passos com os subconjuntos obtidos através das divisões.

A condição de parada consistem em avaliar na observação se não há mais um ganho de informação significativo com as divisões dos conjuntos.

Uma boa característica do algoritmo é a possibilidade que seja realizada a classificação utilizando tanto dados conhecidos como também não conhecidos (sem ou com falta de dados), o procedimento necessário para que seja possível realizar a classificação com dados não conhecidos é feito criando uma associação de probabilidade a cada um dos possíveis valores do atributo faltante. A estimativa da probabilidade se dá por conta da verificação da média dos valores para o atributo nos exemplos do nó daquele atributo. A partir desses valores é calculado o ganho de informação desse atributo. Para cada ramo em que o valor do atributo desconhecido foi visto, é calculado um peso para essa classe. A saída consiste em um dicionário contendo todas as possíveis classes com seus determinados pesos.

3.2 Descrição dos dados selecionados

De acordo com a definição do problema, aluno de bom desempenho é aquele que se forma em até 4 anos e meio, ou se forma com média igual ou superior a 7, ou se forma com número de reprovações igual inferior a 5.

Temos, portanto, a seguinte representação lógica para esta classificação:

$$A \vee (B \wedge C) \vee (B \wedge D) \tag{-1}$$

A: O aluno se forma em 4 anos e meio.
B: O aluno se formou.
C: O aluno tem média igual ou superior a 7.
D: O aluno tem número de reprovações igual ou inferior a 5.

Seja "bom aluno"um termo para o aluno que possui bom desempenho. Considere um aluno qualquer a e uma matéria m. A este par (a, m) iremos atribuir as variáveis $X_{a,m}$ e $Y_{a,m}$.

Seja $T_{m,n}$ a quantidade de bons alunos que passaram na matéria m na n-ésima tentativa, e U_m a quantidade total de bons alunos que já cursaram a matéria m. Então:

$$P_{m,n} = \frac{T_{m,n}}{U_m}$$

Seja $f_{a,m}$ uma função que retorna a quantidade de tentativas que o aluno a realizou para passar na matéria m.

Seja $g_{a,m}$ uma função booleana que retorna 1 caso o aluno a tenha sido aprovado na matéria m, e 0 caso ele tenha reprovado na matéria m mesmo após todas as suas tentativas. Então:

$$X_{a,m} = P_{m,f_{a,m}} * g_{a,m}$$

Seja $M_{m,n}$ a média geral dos bons alunos que cursaram a matéria m por n vezes.
Seja $N_{a,n}$ a nota do aluno a na n-ésima tentativa de passar na matéria.
Então:

$$Y_{a,m} = \frac{\sum_{i=1}^{f(a,m)} P_{m,i} * D_{a,m,i}}{\sum_{i=1}^{f(a,m)} P_{m,i}}$$

Sendo $D_{a,m,n}$ o desvio normalizado:

$$D_{a,m,n} = \frac{(N_{a,n} - M_{m,n}) + 10}{20}$$

Após a inspeção visual dos dados através de recursos gráficos, foram selecionadas 4 matérias a serem introduzidas nos dados finais para o treinamento da árvore de decisão. As demais matérias foram desconsideradas.
Cada matéria contribui com um valor $X_{a,m}$ e um valor $Y_{a,m}$ para cada aluno a.

- Cálculo 1
- Lógica Matemática
- Física para Computação
- Cálculo 2

Adicionalmente, a matéria Matemática Discreta foi introduzida junto às quatro demais em um conjunto de dados alternativo, feito especificamente para resolver problemas de predição entre alunos pertencentes à grade antiga.

Devido ao remanejamento do plano de ensino da matéria Matemática Discreta na grade nova, constatamos que não existe para ela um esquema simples de equivalência tal como para as demais quatro matérias selecionadas. Sendo assim, para um treinamento generalizado, que deve predizer resultados tanto de alunos da grade antiga quanto para alunos da grade nova, esta matéria foi desconsiderada.

Os dados finais que serão considerados no treinamento consistem em um arquivo CSV contendo 643 linhas por 12 colunas.

	Unnamed: 0	0	1	2	3	5	10	11	12	13	15	20
0	0	0.114286	0.114286	0.861111	0.161765	0.000000	0.353357	0.353357	0.554274	0.363824	0.438167	1.0
1	1	1.000000	0.950820	1.000000	0.114286	0.138889	0.480515	0.477155	0.577063	0.385214	0.370347	1.0
2	2	0.161765	0.161765	0.845070	0.923077	0.969231	0.368750	0.368750	0.498167	0.483583	0.413968	1.0
3	3	0.114286	0.861111	0.161765	0.161765	0.923077	0.352357	0.494274	0.419706	0.419706	0.448583	1.0
4	4	1.000000	0.000000	0.861111	0.161765	0.845070	0.452063	0.403952	0.519274	0.313015	0.493167	1.0

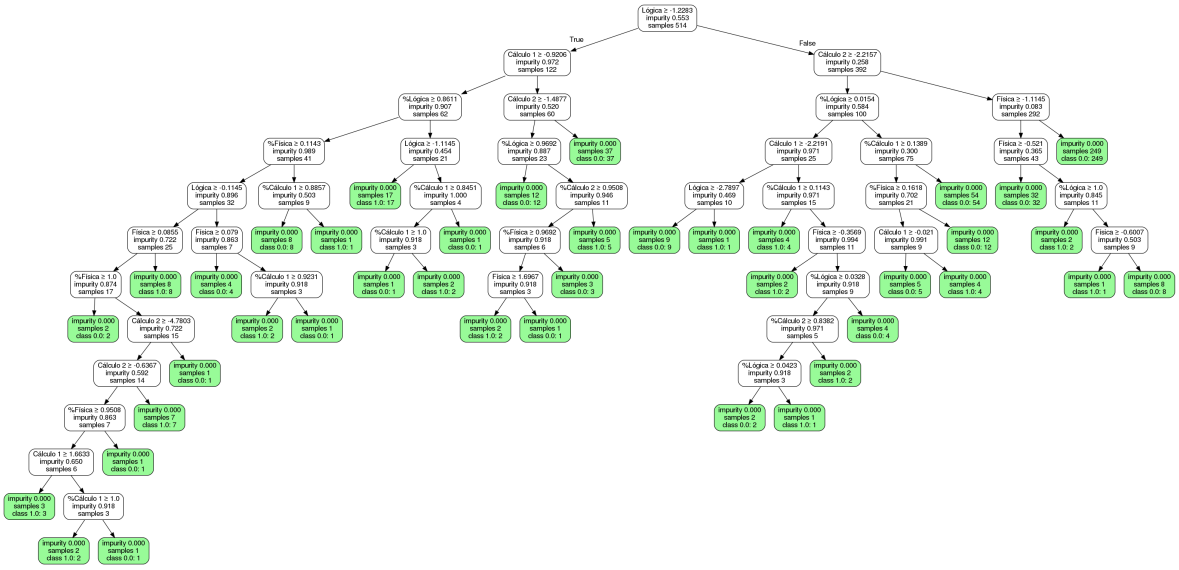
A primeira coluna representa uma identificação do aluno.
As colunas nomeadas de 0, 1, 2, 3, 5 representam o atributo $X_{a,m}$ de cinco matérias. Já as colunas 10, 11, 12, 13, 15 representam o atributo $Y_{a,m}$ destas mesmas cinco matérias. Atributos referentes à mesma matéria estão separados por dez unidades: sendo assim, as colunas 0 e 10, 1 e 11, 2 e 12, 3 e 13 e 5 e 15 são referentes à mesma matéria.

A coluna nomeada 20 é a classificação final, que denota se o aluno pode ser classificado como aluno de bom desempenho (valor 1) ou não (valor 0).

4 Resultados obtidos

4.1 Árvore gerada

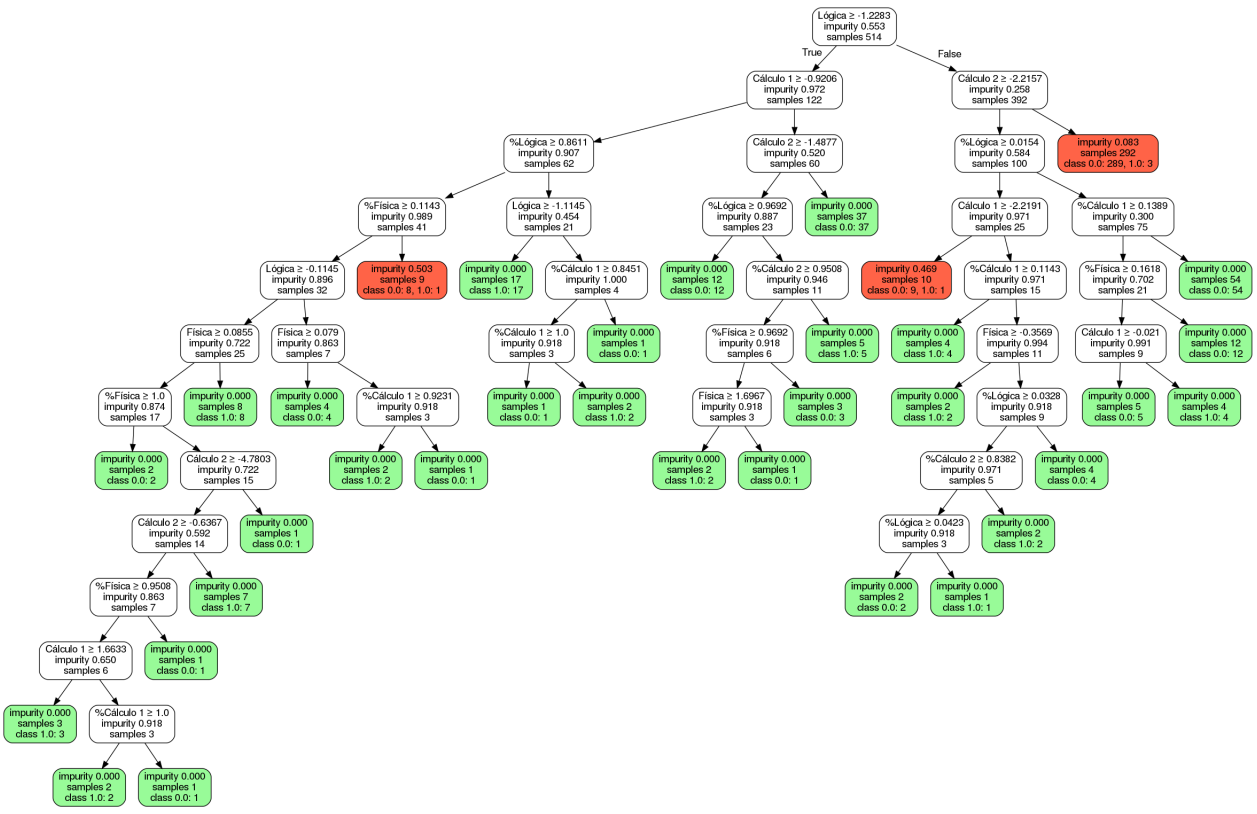
Para gerar uma visualização da árvore gerada foi utilizado o pydoplus, que trata-se de uma interface Python para a linguagem Dot do Graphviz.



Os nós folhas que apresentam a cor verde sãos os que tem o número de impureza em 0, ou seja está totalmente classificado em uma única classe, já os nós folhas na cor vermelha são os que apresenta um grão de impureza maior ou igual a 1.

O processo de construção consiste em pegar todas as informações do nó e transformar em um dicionario de strings e posteriormente é gerado um modelo conforme definido no formato DOT, o retorno do DOT será representado a árvore gerada.

Logo após foi realizada a poda da árvore, onde é realizado uma busca na árvore de baixo pra cima, transformando os ramos que não apresentam nenhum ganho significativo em nós folhas.



4.2 Acurácia

Utilizando a função *train_test_split*() do scikit-learn [3], onde é produzido subconjuntos de dados que serão utilizados para testes e treinamento de maneira aleatória, foram realizados um total de 186 testes com os datasets disponíveis.

- O *dataset_reduzido.csv* : sem a matéria de Matemática Discreta (80 testes)
- O *dataset_reduzido_grade_antiga.csv* : com a matéria de Matemática Discreta (106 testes)

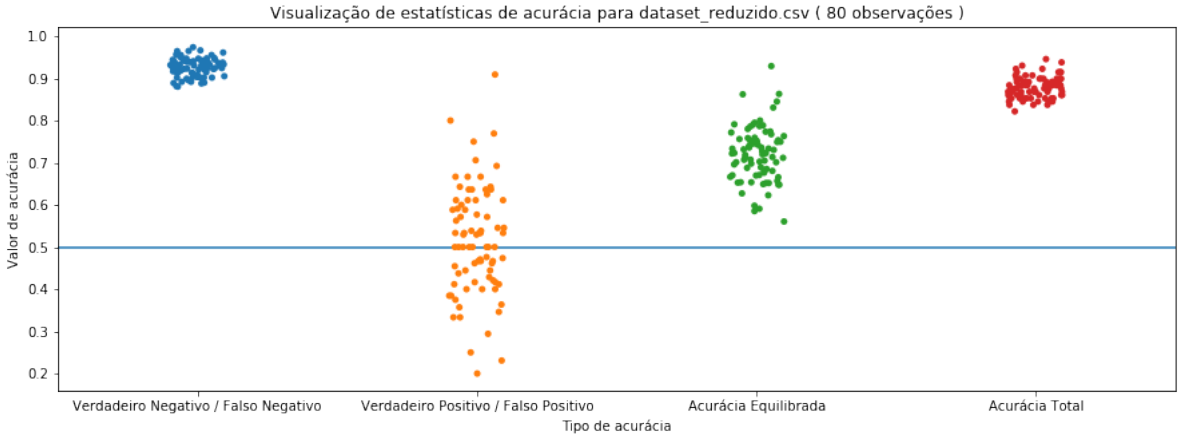
	Verdadeiro Negativo / Falso Negativo		Verdadeiro Positivo / Falso Positivo		Acurácia Equilibrada		Acurácia Total	
	mean	std	mean	std	mean	std	mean	std
arquivo								
dataset_reduzido.csv	0.93	0.02	0.51	0.13	0.72	0.07	0.88	0.03
dataset_reduzido_grade_antiga.csv	0.93	0.02	0.56	0.13	0.75	0.07	0.89	0.03

O modelo foi capaz de distinguir entre:

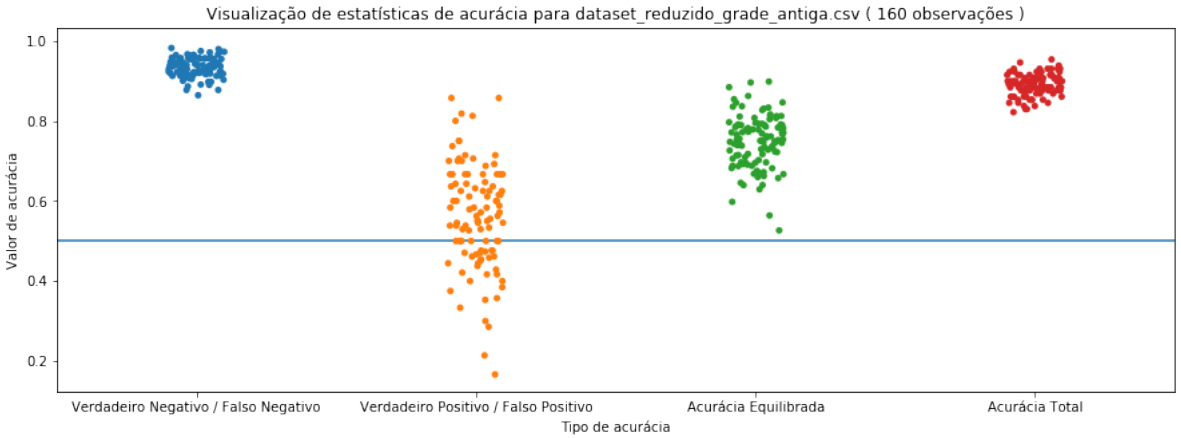
- Verdadeiro positivo (modelo previu corretamente que o desempenho do aluno foi bom) e falso positivo (modelo previu incorretamente que o desempenho do aluno foi bom) com 51% de ocurácia para o *dataset_reduzido.csv*, e de 56% para o *dataset_reduzido_grade_antiga.csv*.
- Verdadeiro negativo(modelo previu corretamente que o desempenho do aluno foi mau) e falso negativo(modelo previu incorretamente que o desempenho do aluno foi mau) com 93% de ocurácia para o *dataset_reduzido.csv*, e de 93% para o *dataset_reduzido_grade_antiga.csv*.

Foi resultado uma acurácia total de 88% para o *dataset_reduzido.csv*, e de 80% para o *dataset_reduzido_grade_antiga.csv*.

Algumas observações foram feitas com base na visualização de estatísticas de acurácia para os dois dataset.



Nessa visualização, se vê que o resultado de 51% na acurácia de verdadeiro positivo ou falso positivo se dá por conta da distribuição dos resultados obtidos.



Nessa segunda visualização, a distribuição de maior influência no resultados do final para os dois datasets se deu pela acurácia de verdadeiro positivo e falso positivo. Isso mostra que ao tentar verificar se o aluno obteve o bom desempenho, o modelo irá ter uma grande variação no seu resultado.

5 Como a solução proposta pôde resolver o problema

Diante do desafio de criar previsões para auxiliar novos alunos, alguns passos foram decididos em grupo com objetivo de ser justo com os parâmetros escolhidos e a sua influência no desempenho final do aluno em geral.

O modelo atingiu alta precisão na detecção de verdadeiros negativos, porém, na detecção de verdadeiros positivos, seu desempenho se encontra um pouco acima da aleatoriedade. Devido ao tamanho reduzido da base de dados, estratégias de *downsampling* e *oversampling* não foram eficientes para melhorar a abordagem. Como direção futura, portanto, propõe-se um ajuste nas heurísticas que governam as decisões do algoritmo, e principalmente procurar meios de aumentar a base de dados para que resultados possam ser gerados a partir de volumes mais significativos das classes minoritárias (como, por exemplo, através da incorporação de dados de outras universidades com perfis semelhantes).

6 Conclusões

Neste trabalho busca-se observar todos aspectos necessários para se elaborar uma boa solução para o problema de previsão de desempenho de um novo aluno, contudo, algumas dificuldades se mostraram presentes nos dados e na escolha dos atributos que apresentam uma boa correlação com o desempenho final do aluno. Dados com ruídos tiveram um grande impacto na tentativa de criar uma previsão positiva; entretanto, isso pôde ser minimizado com algumas boas práticas de tratamento de dados e estratégias em relação aos atributos escolhidos. Conclui-se então que o modelo proposto obteve um bom resultado, tendo em consideração o problema apresentado ao grupo.

Referências

[1] Attux, P. F. J. V. Z. . R. R. F. **Árvores de Decisão** . `ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia004_1s10/notas_de_aula/topico7_IA004_1s10.pdf`, último acesso em Julho de 2019.

[2] Faceli, K. **Inteligência Artificial**. In: *Uma Abordagem de Aprendizado de Máquina*, p. 83–106. Publishing Press, 2011.

[3] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. **Scikit-learn: Machine learning in Python**. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] Peter Norvig, S. R. **Inteligência Artificial**. Prentice Hall, 3 ed. edition, 1994.