A Importância da Educação dos Pais na Performance dos Estudantes

Nicholas Richers[1]

I. DESCRIÇÃO DO PROBLEMA

Educação é reconhecidamente um fator chave para se alcançar uma economia próspera de longo prazo. Nas últimas décadas do século XX, o nível educacional entre os portugueses melhorou, no entanto, as estatísticas da primeira década do século XXI mantiveram Portugal entre os países com maiores taxas de insucesso e abandono escolar. Por exemplo, em 2006, 40% dos jovens portugueses entre 18 e 24 anos abandonavam precocemente a escola, enquanto a média da União Europeia foi de apenas 15% [1].

A área de educação é um terreno fértil para aplicações de Data Mining (DM) devido as diversas fontes de dados e muitos grupos de interesse como alunos, professores, administradores, os próprios familiares ou ex alunos [2]. Existem várias questões interessantes que podem ser respondidas usando técnicas de Machine Learning nesse âmbito [3]: Quem são os estudantes que recebem mais créditos por horas de aula? Quem é provável que retorne para as classes? Quais cursos podem ser oferecidos para atrair mais alunos? Quais são as principais razões para transferências de estudantes? É possível prever o desempenho do aluno? Quais são os fatores que afetam o desempenho do aluno?

Neste trabalho, analisaremos um conjunto de dados do repositório *UCI - Student performance Dataset*, de duas escolas públicas da região do Alentejo, em Portugal, durante o ano letivo de 2005 – 2006. Trazendo diversas informações demográficas, socioeconômicas e referentes a vida estudantil do aluno.

O presente artigo tem como objetivo analisar os dados referentes do nível de escolaridade dos pais e as características pertencentes aos estudantes entrevistados. Dessa forma, espera-se realizar a classificação dentro deste conjunto de dados e desenvolver a capacidade de predição do nível de escolaridade da mãe de acordo com as características dos atributos envolvidos.

Para cada uma dessas abordagens, serão testadas duas configurações de entrada (com e sem a escolaridade do pai) e seis algoritmos de DM (por exemplo, Árvores de decisão e Random Forest). Além disso, uma análise explicativa será realizada sobre os melhores modelos, a fim de identificar as características mais relevantes.

II. PESQUISA BIBLIOGRÁFICA

Data mining é o campo da descoberta de novas informações potencialmente aproveitáveis a partir de grandes quantidades de dados [4]. Nesse contexto, técnicas de DM aplicadas ao ramo da educação ainda estão nos primeiros passos [5]. O artigo de Minaei-bigdoli [7], é citado por diversos autores [8][6], como um dos primeiros trabalhos a utilizarem algoritmos genéticos para prever performance acadêmica de estudantes.

Ao longo dos anos diversos artigos foram publicados com esse tema, contudo, nota-se divergências quanto ao modelo de melhor performance nesses estudos. Em [9] Naive Bayes é citado como a melhor performance, já em [10] o melhor resultado encontrado foi com o SVM e em [8] Random Forest.

Essa diferença pode ser causada por diferenças nos atributos considerados e no tamanho do conjunto de dados, onde em [11] três métodos foram usados para lidar com o problema de desequilíbrio de classes e todos eles mostram resultados satisfatórios. Primeiro, balancearam os conjuntos de dados e usaram o o SVM para os pequenos conjuntos de dados e Decision Tree para os conjuntos de dados maiores.

Nesse contexto se destaca uma revisão sistemática da literatura [12] que realiza um levantamento quantitativos das principais técnicas de DM e também procura identificar quais os atributos mais importantes nos dados dos estudantes. Dez dos trinta artigos avaliados consideram o histórico de notas do aluno o principal atributo para predição de desempenho [12], seguidos por dados demográficos como a escolaridade dos pais.

O levantamento das técnicas mais usadas [12] revela que Decision Tree (DT) é a técnica mais utilizada, presente em dez dos trinta estudos avaliados, seguido por Neural Network (NN) com oito, Naive Bayes (NB) com quatro e K-Nearest Neighbor (kNN) com apenas três.

Levando em conta o melhor desempenho desempenho de cada técnica [12] considerando todos os artigos temos: NN com 98% como o melhor resultado, isso ocorreu devido a influência de um híbrido de dois dos principais atributos que eram o sistema de avaliação interno e externo da escola. Em seguida DT com 91%, SVM e kNN com 83% e por fim NB com 76%.

III. DESCRIÇÃO DOS DADOS

O Conjunto de Dados escolhido possui 1044 registros, distribuídos em 649 registros relacionados à disciplina de português e 395 à matemática. Contudo devido a 382 registros de intercessão entre os arquivos, de forma que só seriam adicionados 13 registros, havendo perda de informações como as notas dos alunos. Então, para o trabalho de classificação foi decidido utilizar o arquivo relacionado à disciplina de português, por possuir um número maior de

[1]N. Richers - Programa de Engenharia de Produção COPPE/UFRJ nicholasrichers@gmail.com

TABLE I
DESCRIÇÃO DOS ATRIBUTOS DO CONJUNTO DE DADOS

Atributo	Descrição
gênero	gênero do estudante (binário: feminino ou masculino)
idade	idade do estudante (numérico: 15 a 22 anos)
escola	de qual escola é o estudante (binário: Gabriel Pereira ou Mousinho da Silveira)
endereço	o tipo de endereço do estudante (binário: urbano ou rural)
Pstatus	binário: se os pais moram juntos ou separados
Medu	educação da mãe (numérico: 0- nenhuma, 1- educação primária (4o Ano), 2- 5o ao 9o ano, 3- escola secundária ou 4 -Ensino Superior)
Mjob	trabalho da mãe (nominal)
Fedu	educação do pai (numérico: 0- nenhuma, 1- educação primária (4o Ano), 2- 5o ao 9o ano, 3- escola secundária ou 4 -Ensino Superior)
Fjob	trabalho do pai (nominal)
guardian	responsável pela guarda do estudante (nominal: mãe, pai, outro)
famsize	tamanho da família (binário: <= 3 ou >3
famrel	qualidade das relações familiares (numérico: 1 - muito ruim até 5 - excelente)
reason	razão pela qual escolheu esta escola (nominal: próximo de casa, reputação da escola, preferência do curso, ou outro)
traveltime	tempo do percurso de casa até a escola (numérico: 1 - <15 min , 2 - 15 a 30 min, 3- 30 min a 1 hora, ou 4 - >1 hora)
studytime	tempo de estudo por semana (numérico: 1 - <2 horas, 2 2 - 2 a 5 horas, 3 - 5 a 10 horas, 4 - >10 horas)
failures	numero de reprovações (numérico: n, se 1<= n <3, se não 4)
schoolsup	apoio escolar extra (binário: sim ou não)
famsup	suporte educacional da família (binário: sim ou não)
activities	atividades extracurriculares (binário:sim ou não)
paidclass	aulas particulares (binário: sim ou não)
internet	acesso a internet em casa (binário: sim ou não)
nursery	frequentou o maternal (binário: sim ou não)
higher	pretende cursar o ensino superior (binário: sim ou não)
romantic	está em algum relacionamento (binário: sim ou não)
freetime	tempo livre fora da escola (numérico: 1 - bastante tempo livre até 5 - pouco tempo livre)
goout	sai com amigos (numérico: 1 - muito pouco, até, 5 - bastante)
Walc	consumo de álcool semanal (numérico: 1 - muito pouco até 5 - bastante)
Dalc	consumo de álcool diário (numérico: 1 - muito pouco até 5 - bastante)
health	estado de saúde atual (numérico: 1 - muito ruim até 5 - muito bom)
absences	número de faltas na escola (de 0 até 93)
G1	nota do primeiro período (numérico: 0 a 20)
G2	nota do segundo período (numérico: 0 a 20)
G3	nota do terceiro período (numérico: 0 a 20)

registros. Dessa forma foi realizada uma análise estatística nos registros relacionados apenas a essa disciplina.

A Documentação original do dataset menciona um questionário contendo 37 perguntas para a formulação desse conjunto de dados, a descrição dos atributos pode ser visto na Tabela I. A documentação menciona que alguns desses atributos foi descartado devido ao excesso de valores ausentes, especialmente em variáveis referentes a renda familiar. Também é mencionado que 111 registros de alunos foram descartados devido a falta de detalhes de identificação, devido a esse pré-processamento já realizado nos dados disponibilizados, não há a presença de valores ausentes, como pode ser visto na Figura 1.

Para melhorar a avaliação e a análise dos dados, as variáveis nominais foram sofreram um agrupamento de opções e foram binarizadas de forma a não criar uma falsa proximidade entre as opções disponíveis. Os atributos 'Fjob'e 'Mjob' foram agrupados em trabalhos dentro e fora de casa. No atributo 'Guardian', as opções mãe e pai foram agrupadas como uma opção, deixando a outra opção como outros. Por fim, o atributo relacionado a escolha da escola foi excluído. A tabela com as descrições modificadas pode ser encontrada no Apêndice A.

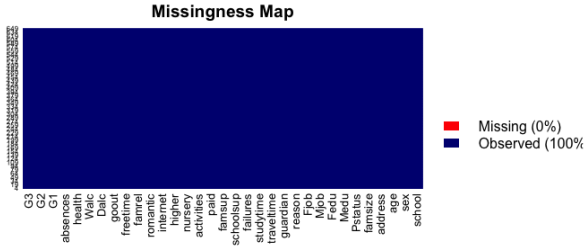

Fig. 1. Valores Ausentes

IV. APRESENTAÇÃO TECNOLÓGICA

Para a etapa de pré processamento de dados foi utilizado o software R, suportado pelos pacotes **fBasics**, **grid**, **gridExtra**, **corrplot** entre outros. Para a geração de modelos foi utilizado a linguagem python por entender que a biblioteca scikit-learn é mais adequada essa tarefa.

V. AVALIAÇÃO PRELIMINAR DOS DADOS

Nessa seção é feita uma análise preliminar dos dados, considerando as alterações previstas nas seções anteriores. Os códigos desenvolvidos na etapa de pré processamento encontram-se no Apêndice B desse artigo.

A. *Análise Exploratória*

A tabela no Apêndice C apresenta estatísticas básicas dos atributos numéricas do conjunto de dados de forma a auxiliar a análise dos dados. É interessante observar que a média da escolaridades das mães (MEdu) é um pouco superior a dos pais (FEdu), mesmo assim, há uma proporção maior de mulheres trabalhando apenas em casa.

B. *Histogramas*

O Apêndice D dispões os respectivos histogramas das variáveis numéricas do conjunto de dados. Através das distribuições dos atributos, seguindo as tabelas e os histogramas apresentados, verifica-se como o comportamento dos estudantes entrevistados se encontram no dataset. Alguns atributos seguem uma distribuição próxima à normal, como idade (age), tempo livre (freetime), sai com os amigos (goout) e notas dos alunos (G1, G2 e G3).

É interessante também notar a assimetria de certas distribuições, como a de quem quer seguir para o ensino superior (higher) e se possui acesso à Internet em casa (internet), enquanto outros atributos são muito mais próximos de uma distribuição simétrica, como se o aluno faz atividades extra-curriculares (activities)

C. *Verificação de Outliers*

Com o objetivo de facilitar a visualização foi feita uma padronização dos atributos usando o método Z-Score com o objetivo de identificar outliers como verificado Na Figura 2. Dessa forma, avaliando os gráficos, verificamos a existência de Outliers especialmente no atributo referente as faltas (absences) e alguns poucos nos atributos referentes às notas (G1 e G2), assim como na idade (age), reprovações (failures), tempo de viagem para chegar à escola (traveltime) e tempo de estudo (studytime). Também é possível notar que a maioria dos atributos são binários e por isso há atributos como acesso a rede (internet) onde praticamente não há variância.

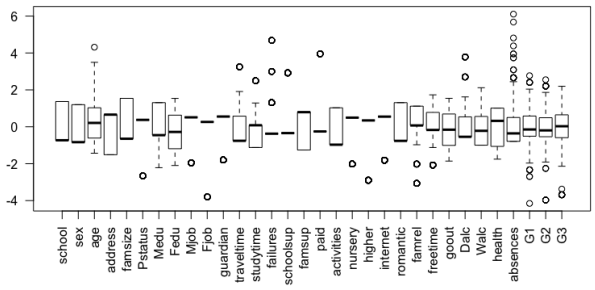

Fig. 2. Gráfico Boxplot

Para a remoção dos Outliers o método da médias das distâncias euclidianas para cada entrada foi empregado conforme a Figura 3, e após uma análise visual determinou-se δ = 10 como valor limite para o valor médio da distância euclidiana, resultando na eliminação de 6 registros do conjunto de dados, restando então 643 registros.

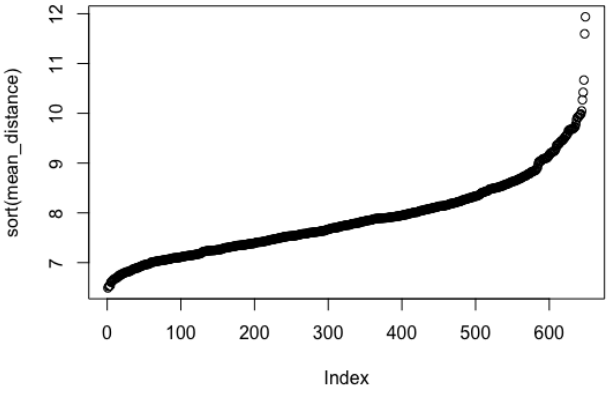

Fig. 3. Média Distancias Euclidiana

D. *Correlações*

Analisando a matriz de correlação das variáveis numéricas na Figura 4 verificamos que a maioria das correlações entre os atributos é fraca, mas podemos destaxar uma correlação próxima a 0.6 entre a educação do Pai (Fedu) com a da Mãe (Medu), o consumo de álcool em dias de semana (Dalc) com o consumo em finais de semana (Walc), e uma correlação próxima a 0.8 entre as primeiras notas (G1 e G2) e a variável de saída (G3), por fim há uma correlação negativa entre o número de reprovações (Failures) e suas notas.

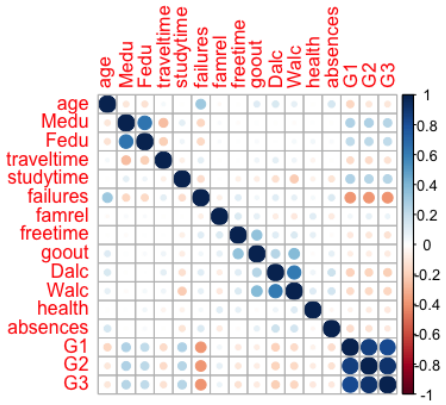

Fig. 4. Matriz de Correlação

VI. METODOLOGIA

Os atributos relacionados a educação dos pais (Medu e Fedu), estão distribuídos em 5 classes, de acordo com a TabelaI e podem ser vistos na Figura 5. A Partir dessa análise, foi definido que o atributo **Medu será o target**, por possuir uma distribuição menos assimétrica.

Através da Figura 5, podemos inferir que uma parcela muito pequena não possui nenhum tipo de educação formal, dessa forma as classes 1 e 2 serão compiladas de uma de forma que o problema se aproxime de uma distribuição uniforme.

Nesse artigo iremos avaliar o problema de 4 classes com e sem a presença do atributo (Fedu) no modelo, de forma a verificar a influência dessa variável nos resultados. Para classificar os registros foram utilizadas diferentes modelos, desde os mais simples até outros mais complexos, buscando aquele dentro do problema proposto que pudesse obter os melhores resultados. Ao final os resultados serão comparados para decidir os modelos que melhor se aplicam ao problema.

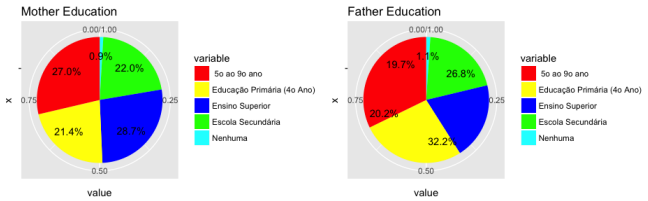

Fig. 5. Distribuição das variáveis Medu e Fedu

Para a avaliação dos resultados foi utilizado validação cruzada de 10 ciclos, evitando possível overfitting, e métricas como precisão (fração de instâncias recuperadas que foram previstas corretamente), recall (fração de instâncias de uma classe que foram previstas como sendo daquela classe), f1 (média ponderada entre precisão e recall para cada classe), f1-weighted (média ponderada dos resultados do f1-score levando em conta o tamanho de cada classe) e erro quadrático médio (média da diferença entre o valor do estimador e do valor real ao quadrado) de cada modelo.

VII. Resultados

Nessa seção os principais resultados relativos a cada modelo, somados a breves comentários sobre a formulação de cada um deles.

A. Regressão Logística

O modelo de regressão logística adapta técnicas de regressão linear para a determinação de uma superfície de separação entre duas classes (numericamente 0 ou 1, classificação binária) discriminada por uma curva sigmóide regularizada por uma função custo com fatores exponenciais que penalizam severamente o modelo em caso de erros na predição. Isto ocorre para que a curva possa ser melhor ajustada ao modelo, evitando o overfitting.

Apesar de ser um modelo mais simples comparado aos outros modelos, a regressão logística apresentou resultados semelhantes e com uma leve melhora quando a variável relacionada a educação do pai foi adicionada.

B. Naive Bayes

A classificação por um classificador bayesiano é obtida por uma regra de decisão sobre as probabilidades a posteriori, podendo esta regra ser a decisão pelo mínimo erro de classificação ou pelo mínimo risco condicional.

TABLE II
Resultados Regressão Logística sem Fedu

Class	precision	recall	f1-score	support
1	0.41	0.48	0.44	149
2	0.3	0.27	0.28	184
3	0.19	0.09	0.12	137
4	0.44	0.61	0.51	173
Avg./Total	0.34	0.37	0.35	643

TABLE III
Resultados Regressão Logística com Fedu

Class	precision	recall	f1-score	support
1	0.48	0.56	0.52	149
2	0.42	0.42	0.42	184
3	0.22	0.11	0.14	137
4	0.59	0.73	0.65	173
Avg./Total	0.43	0.47	0.45	643

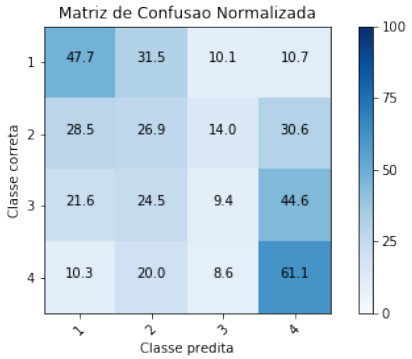

Fig. 6. Matriz de Confusão Reg. Logística Sem 'Fedu'

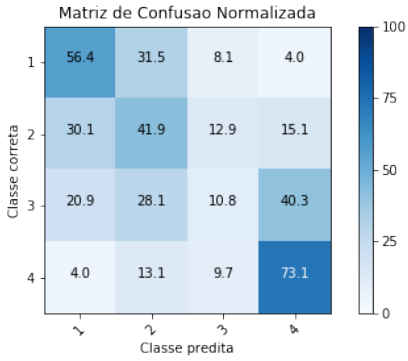

Fig. 7. Matriz de Confusão Reg. Logístic Com 'Fedu'

A decisão pelo mínimo erro de classificação pode ser explicada utilizando a região em comum do gráfico de distribuição de probabilidade condicional de modo a minimizar a área da qual pertence ambas as distribuições, em um caso com duas classes, de modo a minimizar a probabilidade de classificação incorreta.

Já a decisão pelo mínimo risco condicional utiliza a matriz de confusão, explicada em um tópico a seguir, é utilizada quando se tem uma classe positiva e outra negativa, na qual

TABLE IV

RESULTADOS NAIVE BAYES SEM FEDU

Class	precision	recall	f1-score	support
1	0.42	0.44	0.43	149
2	0.39	0.17	0.24	184
3	0.3	0.23	0.26	137
4	0.4	0.69	0.51	173
Avg./Total	0.34	0.37	0.35	643

TABLE V

RESULTADOS NAIVE BAYES COM FEDU

Class	precision	recall	f1-score	support
1	0.46	0.48	0.47	149
2	0.38	0.19	0.25	184
3	0.29	0.23	0.26	137
4	0.43	0.72	0.54	173
Avg./Total	0.39	0.41	0.38	643

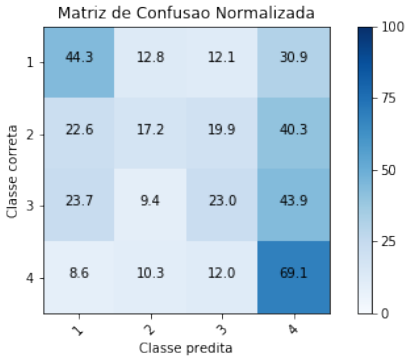

Fig. 8. Matriz de Confusão Naive Bayes Sem 'Fedu'

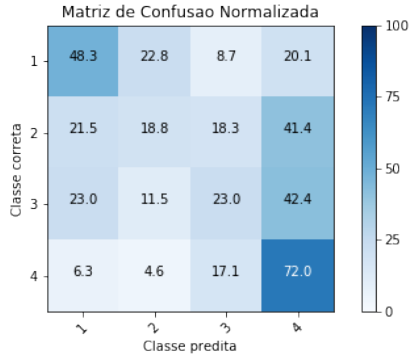

Fig. 9. Matriz de Confusão Naive Bayes Com 'Fedu'

o erro de classificação ocorre quando uma grandeza positiva é predita como negativa ou o oposto. Pela regra de Bayes é possível associar uma probabilidade a cada decisão com a utilização de um termo relacionado ao custo de escolha da classe errada.

Entre as possibilidades de modelo de classificação bayesiana, (Bernoulli, Multinomial e Gaussiana), o modelo gaussiano foi utilizado por considerar que todas as variáveis possuem uma distribuição normal, indo de acordo com a padronização de variáveis feita anteriormente.

C. Redes Neurais

Redes neurais são modelos computacionais de Machine Learning que são inspirados por e pretendem simular o funcionamento e complexidade do cérebro humano, como neurônios artificiais.

Redes neurais são compostas por um número variável de camadas contendo neurônios que são ligados aos neurônios das camadas anterior e seguinte. A primeira e a última camadas contém as variáveis de entrada e de saída respectivamente. As camadas intermediárias, ditas ocultas fazem o processamento das variáveis, com cada neurônio fazendo um único processamento baseado no peso das conexões, combinando-as e depois aplicando uma função de ativação sobre o resultado. A saída de uma camada é utilizada como entrada da camada seguinte.

O primeiro modelo testado, com duas camadas intermediárias de dois neurônios cada, apresentou o problema citado anteriormente relacionado a ausência da possibilidade de um ajuste fino. Pode-se notar como o modelo realizou a classificação atribuindo todos os registros do dataset como sendo da classe 1, a classe majoritária. Esse enviesamento da classificação diminuiu com o aumento da capacidade de ajuste das camadas, ou seja, o aumento no número de neurônios.

(i) RN 2 camadas e 2 Neurônios

TABLE VI

RESULTADOS RN2 SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0	0	0	149
2	0.29	1	0.45	184
3	0	0	0	137
4	0	0	0	173
Avg./Total	0.08	0.29	0.13	643

TABLE VII

RESULTADOS RN2 COM 'FEDU'

Class	precision	recall	f1-score	support
1	0	0	0	149
2	0.29	1	0.45	184
3	0	0	0	137
4	0	0	0	173
Avg./Total	0.08	0.29	0.13	643

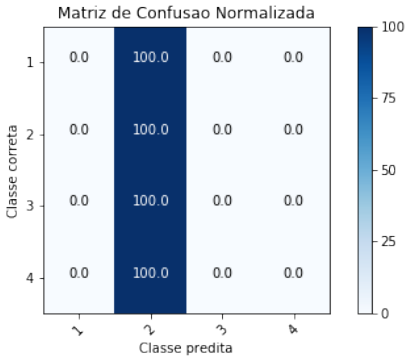

Fig. 10. Matriz de Confusão RN2 Sem 'Fedu'

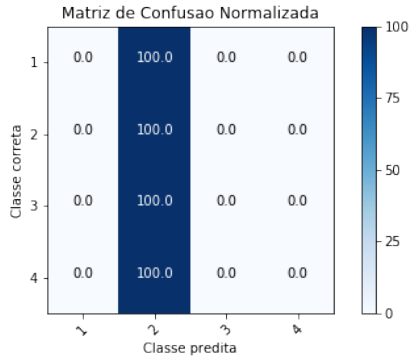

Fig. 11. Matriz de Confusão RN2 Com 'Fedu'

(ii) RN 2 camadas e 5 Neurônios

TABLE VIII
RESULTADOS RN5 SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.38	0.11	0.17	149
2	0.29	0.38	0.32	184
3	0	0	0	137
4	0.29	0.6	0.39	173
Avg./Total	0.25	0.29	0.24	643

TABLE IX
RESULTADOS RN5 COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.46	0.08	0.14	149
2	0.29	0.93	0.44	184
3	0	0	0	137
4	0.48	0.06	0.1	173
Avg./Total	0.32	0.3	0.19	643

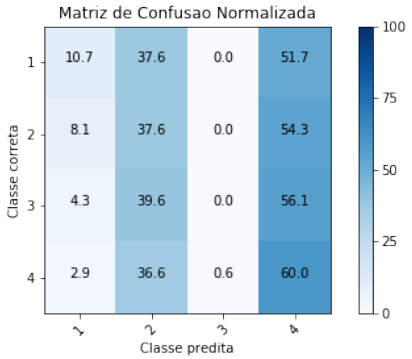

Fig. 12. Matriz de Confusão RN5 Sem 'Fedu'

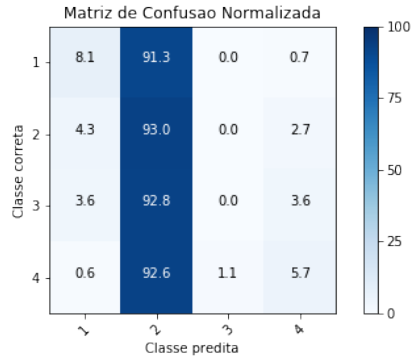

Fig. 13. Matriz de Confusão RN5 Com 'Fedu'

(iii) RN 2 camadas e 10 Neurônios

TABLE X
RESULTADOS RN10 SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.39	0.5	0.44	149
2	0.29	0.3	0.29	184
3	0.22	0.07	0.11	137
4	0.41	0.53	0.46	173
Avg./Total	0.33	0.36	0.33	643

TABLE XI
RESULTADOS RN10 COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.54	0.52	0.53	149
2	0.43	0.58	0.49	184
3	0.4	0.17	0.23	137
4	0.64	0.71	0.67	173
Avg./Total	0.5	0.51	0.49	643

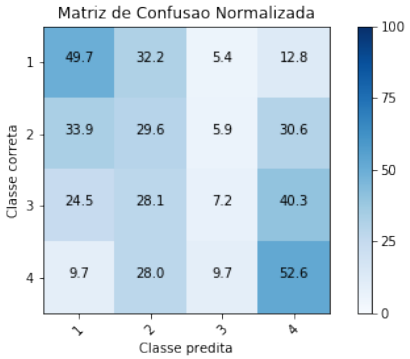

Fig. 14. Matriz de Confusão RN10 Sem 'Fedu'

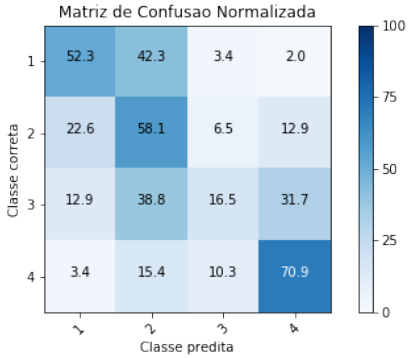

Fig. 15. Matriz de Confusão RN10 Com 'Fedu'

(iv) RN 2 camadas e 20 Neurônios

TABLE XII
RESULTADOS RN20 SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.54	0.52	0.53	149
2	0.43	0.58	0.49	184
3	0.4	0.17	0.23	137
4	0.64	0.71	0.67	173
Avg./Total	0.5	0.51	0.49	643

TABLE XIII
RESULTADOS RN20 COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.49	0.54	0.51	149
2	0.41	0.43	0.42	184
3	0.23	0.15	0.18	137
4	0.6	0.67	0.63	173
Avg./Total	0.44	0.46	0.45	643

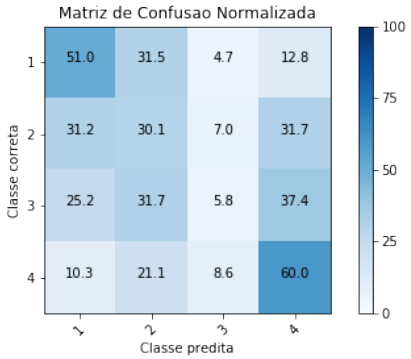

Fig. 16. Matriz de Confusão RN2020 Sem 'Fedu'

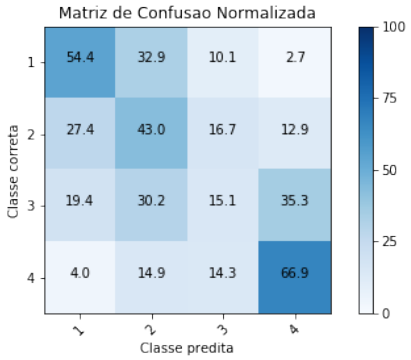

Fig. 17. Matriz de Confusão RN2020 Com 'Fedu'

D. SVM

SVM é um modelo de aprendizado supervisionado usado para classificação. A partir de um conjunto de dados de teste já classificados, o modelo é capaz de prever em que classe as novas amostras pertencem. O método realiza uma separação entre as classes definindo um hiperplano entre elas de forma a maximizar a distância entre os pontos mais próximos de cada classe. Apesar de ser um modelo computacionalmente custoso, foi possível executar diferentes tipos de kernel usados pelo algoritmo já que o problema trata de um dataset relativamente pequeno. Os diferentes kernels estão relacionados à função de similaridade que será usada pelo algoritmo.

(i) SVM - Linear

TABLE XIV
RESULTADOS SVM LINEAR SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.4	0.46	0.43	149
2	0.31	0.33	0.32	184
3	0.2	0.09	0.12	137
4	0.46	0.57	0.51	173
Avg./Total	0.35	0.37	0.35	643

TABLE XV

RESULTADOS SVM LINEAR COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.52	0.57	0.54	149
2	0.44	0.49	0.47	184
3	0.31	0.19	0.23	137
4	0.63	0.69	0.66	173
Avg./Total	0.48	0.5	0.49	643

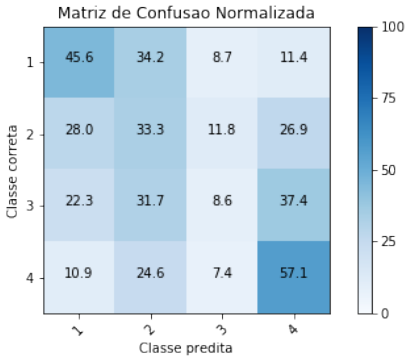

Fig. 18. Matriz de Confusão SVM Linear Com 'Fedu'

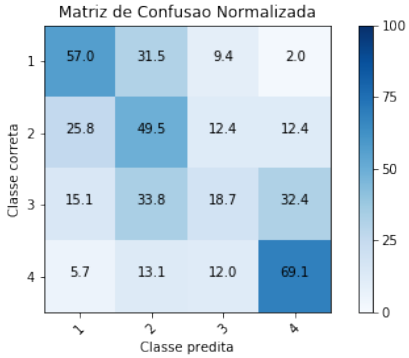

Fig. 19. Matriz de Confusão SVM Linear Com 'Fedu'

(ii) SVM Polinomial

TABLE XVI

RESULTADOS SVM POLINOMIAL SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.37	0.46	0.41	149
2	0.27	0.29	0.28	184
3	0.21	0.19	0.2	137
4	0.42	0.33	0.37	173
Avg./Total	0.32	0.32	0.32	643

TABLE XVII

RESULTADOS SVM POLINOMIAL COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.44	0.54	0.49	149
2	0.34	0.36	0.35	184
3	0.23	0.21	0.22	137
4	0.6	0.48	0.53	173
Avg./Total	0.41	0.4	0.4	643

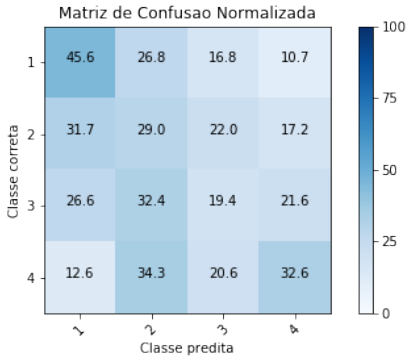

Fig. 20. Matriz de Confusão SVM Polinomial Sem 'Fedu'

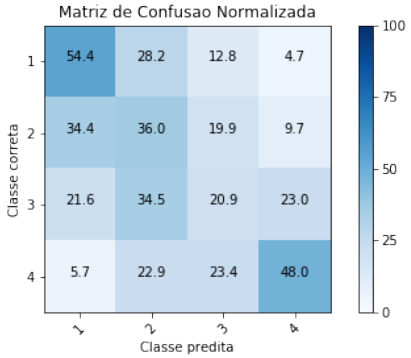

Fig. 21. Matriz de Confusão SVM Polinomial Com 'Fedu'

(iii) SVM RBF

TABLE XVIII

MY CAPTION

TABLE XIX

RESULTADOS SVM RBF SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.33	0.33	0.33	149
2	0.33	0.4	0.36	184
3	0.09	0.03	0.04	137
4	0.45	0.59	0.51	173
Avg./Total	0.31	0.36	0.33	643

TABLE XX

RESULTADOS SVM RBF COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.43	0.41	0.42	149
2	0.38	0.49	0.43	184
3	0.2	0.1	0.13	137
4	0.59	0.65	0.62	173
Avg./Total	0.41	0.43	0.41	643

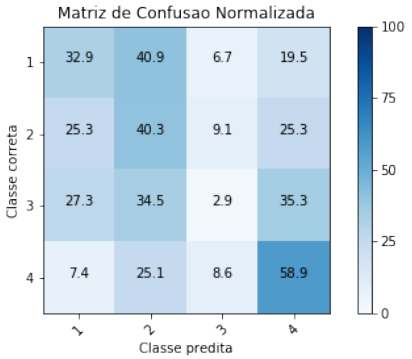

Fig. 22. Matriz de Confusão SVM RBF Sem 'Fedu'

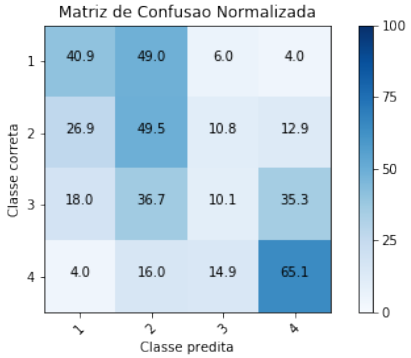

Fig. 23. Matriz de Confusão SVM RBF Com 'Fedu'

E. Árvore de Decisão

O algoritmo de árvore de decisão (Decision Tree — DT) é um algoritmo supervisionado que monta uma árvore onde cada nó é responsável pelo teste de um atributo do sistema. Os valores de probabilidade de cada classe são armazenados naquele nó e servem como parâmetros para a tomada de decisão. A cada nova amostra desconhecida que entra no modelo, o algoritmo percorre os nós avaliando os respectivos atributos para estimar a probabilidade daquela amostra pertencer a uma determinada classe. Dessa forma, usualmente não é necessário percorrer todos os atributos da amostra para realizar a classificação e poupa-se tempo de processamento.

A geração da árvore inicia-se pela caracterização de um nó raiz, que possui meramente a probabilidade de cada classe na amostragem. A partir de então, o nó é dividido sucessivamente, de forma que cada filho represente uma nova característica da amostragem, associado à um conjunto de probabilidades para cada classe relativos a essa característica. Esse processo é repetido para todos os nós até que estes atinjam probabilidade de 100% para alguma classe, configurando-se como um nó folha

(i) Árvore de Decisão Sem Limite

TABLE XXI
RESULTADOS DECISION TREE SEM LIMITE SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.35	0.4	0.37	149
2	0.25	0.23	0.24	184
3	0.15	0.15	0.15	137
4	0.4	0.4	0.4	173
Avg./Total	0.29	0.3	0.29	643

TABLE XXII
RESULTADOS DECISION TREE SEM LIMITE COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.47	0.46	0.46	149
2	0.4	0.44	0.42	184
3	0.18	0.17	0.18	137
4	0.53	0.5	0.51	173
Avg./Total	0.4	0.4	0.4	643

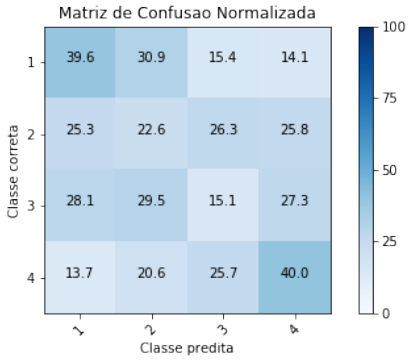

Fig. 24. Matriz de Confusão DT Sem 'Fedu'

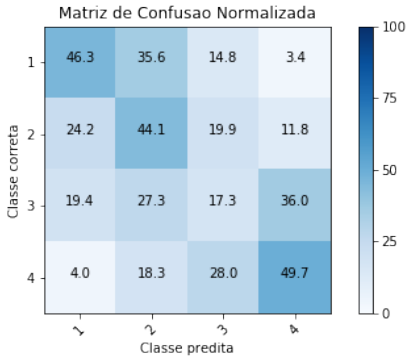

Fig. 25. Matriz de Confusão DT com 'Fedu'

(ii) Árvore de Decisões Profundidade 3

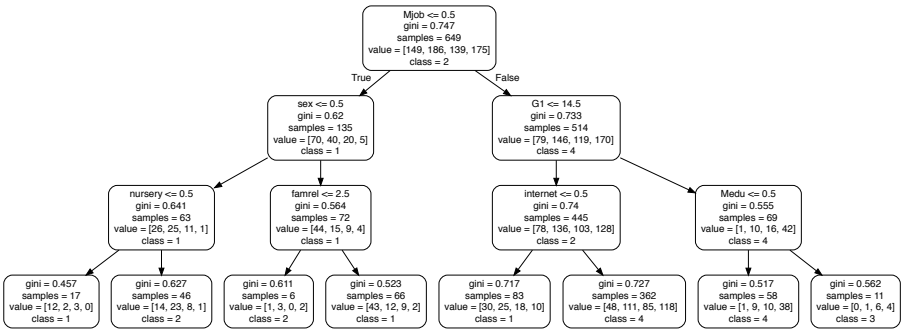

Fig. 26. Arvore de Decisão Sem Fedu com Profundidade 3

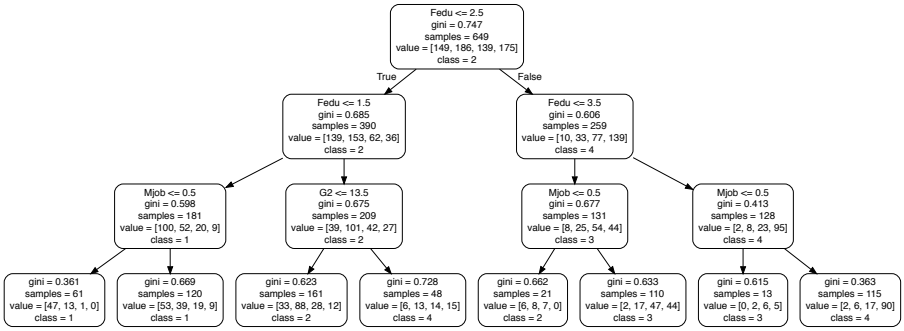

Fig. 27. Arvore de Decisão Com Fedu com Profundidade 3

TABLE XXIII
RESULTADOS DECISION TREE COM PROFUNDIDADE 3 SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.4	0.49	0.44	149
2	0.27	0.17	0.21	184
3	0.18	0.01	0.03	137
4	0.36	0.7	0.48	173
Avg./Total	0.31	0.35	0.3	643

TABLE XXIV
RESULTADOS DECISION TREE COM PROFUNDIDADE 3 COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.52	0.53	0.53	149
2	0.46	0.56	0.5	184
3	0.36	0.31	0.33	137
4	0.64	0.55	0.59	173
Avg./Total	0.5	0.5	0.5	643

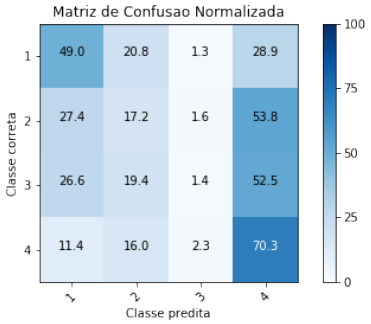

Fig. 28. Matriz de Confusão DT3 Sem 'Fedu'

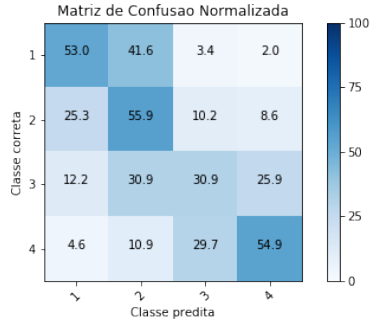

Fig. 29. Matriz de Confusão Dt3 Com 'Fedu'

(iii) Árvore de Decisões Profundidade 5

TABLE XXV
RESULTADOS DECISION TREE COM PROFUNDIDADE 5 SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.41	0.48	0.44	149
2	0.29	0.28	0.28	184
3	0.25	0.1	0.14	137
4	0.4	0.54	0.46	173
Avg./Total	0.34	0.36	0.34	643

TABLE XXVI
RESULTADOS DECISION TREE COM PROFUNDIDADE 5 COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.46	0.44	0.45	149
2	0.46	0.56	0.51	184
3	0.28	0.17	0.21	137
4	0.61	0.7	0.65	173
Avg./Total	0.46	0.49	0.47	643

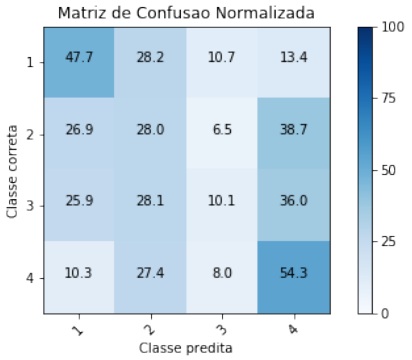

Fig. 30. Matriz de Confusão RT5 Sem 'Fedu'

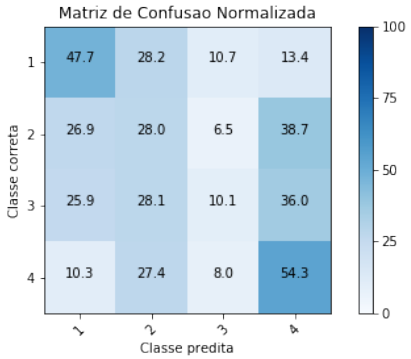

Fig. 31. Matriz de Confusão RT5 Com 'Fedu'

F. Random Forest

O classificador Random Forest (RF) consiste na agregação do resultado de várias árvores de decisão aplicadas a diferentes subsets de registros e atributos. Para um problema de classificação, a RF agrega os resultados das variadas Decision Trees (DT), retornando como predição final a moda (classe mais frequentemente predita). O objetivo é o de reduzir a variância e possível overfitting da DT. Se as regras de decisão forem muito estritas, fazendo a DT fitar muito bem o conjunto de treino, certamente ocorrerá tendência a overfit.

Importante notar que o fato de reunir resultados de vários classificadores em uma única predição diferencia a avaliação de um classificador ensemble dos demais classificadores simples, não sendo "justo" compará-los diretamente, dado o custo computacional consideravelmente maior para efetuar a agregação de resultados. Para este relatório foram gerados 2 modelos de Floresta Aleatória, com 100 árvores e com 500 árvores.

(i) RF - 100 árvores

TABLE XXVII
RESULTADOS RANDOM FOREST 100 ÁRVORES SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.43	0.55	0.48	149
2	0.3	0.31	0.3	184
3	0.16	0.06	0.09	137
4	0.47	0.57	0.51	173
Avg./Total	0.35	0.38	0.36	643

TABLE XXVIII
RESULTADOS RANDOM FOREST 100 ÁRVORES COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.48	0.58	0.53	149
2	0.45	0.51	0.48	184
3	0.37	0.16	0.22	137
4	0.61	0.7	0.65	173
Avg./Total	0.48	0.5	0.48	643

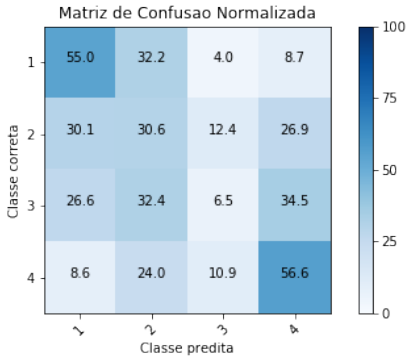

Fig. 32. Matriz de Confusão RF100 Sem 'Fedu'

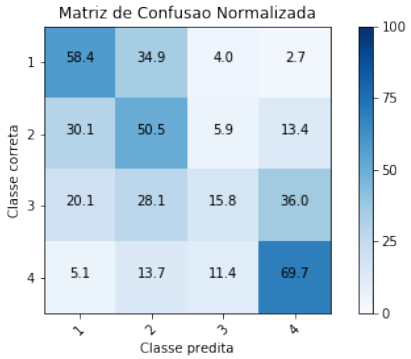

Fig. 33. Matriz de Confusão RF100 Com 'Fedu'

(ii) RF - 500 árvores

TABLE XXIX
RESULTADOS RANDOM FOREST 500 ÁRVORES SEM 'FEDU'

Class	precision	recall	f1-score	support
1	0.42	0.54	0.47	149
2	0.27	0.28	0.27	184
3	0.15	0.06	0.08	137
4	0.46	0.56	0.51	173
Avg./Total	0.33	0.37	0.34	643

TABLE XXX
RESULTADOS RANDOM FOREST 500 ÁRVORES COM 'FEDU'

Class	precision	recall	f1-score	support
1	0.52	0.58	0.55	149
2	0.47	0.53	0.5	184
3	0.29	0.12	0.17	137
4	0.61	0.74	0.67	173
Avg./Total	0.48	0.51	0.49	643

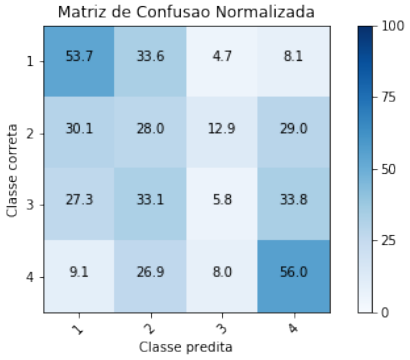

Fig. 34. Matriz de Confusão RF500 Sem 'Fedu'

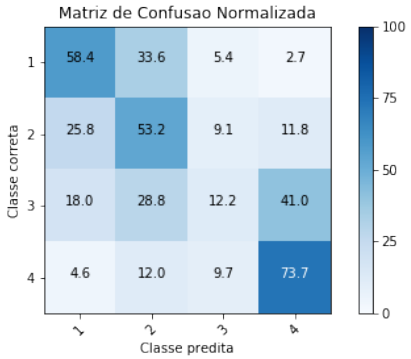

Fig. 35. Matriz de Confusão RF500 Com 'Fedu'

VIII. CONCLUSÕES

Dentro da execução dos diferentes modelos pode-se notar que não existe um modelo único que irá se comportar com a melhor classificação no problema apresentado. Há uma sequência de ajustes que podem ser realizados em cada modelo com o objetivo de tornar sua classificação mais precisa (precision), mais completa (em relação a uma classe de interesse específica) (recall), ou com menor estimativa escolhida de erro e consequentemente ultrapassar outros modelos que se saíram melhor neste relatório.

Na avaliação do problema e com os modelos escolhidos foi possível notar que o atributo Fedu (que representa o nível de escolaridade do Pai do aluno) usualmente agrega uma melhoria considerável na classificação do problema. Esse resultado porém já era esperado pois a variável possui forte correlação com o atributo Medu que é o target do estudo. A comparação entre os modelos pode ser observado na Tabela XXXI e XXXII, com a presença do atributo Fedu e sem o atributo, respectivamente.

É possível notar também que as variáveis que tiveram as maiores correlações com o target do problema, como verificado na figura 4, foram também as consideradas com maior grau de decisão pelo modelo da árvore de decisão. Destaca-se aqui a variável Fedu (quando presente no modelo), que apresentava a maior correlação com a variável Medu no relatório preliminar, além das variáveis Mjob e as notas do aluno (G1, G2 e G3).

Os modelos que apresentaram um maior f1 weighted, a média ponderada dos valores de f1 de cada uma das classes, levando em conta a quantidade de registros de cada classe, e um menor erro quadrático médio, soma do quadrado dos erros de previsão, foram diferentes com e sem a presença da variável Fedu. O modelo de Floresta Aleatória com 500 árvores apresentou maior f1 weighted para ambas as configurações. Já o modelo de Floresta Aleatória com 100 árvores o menor erro quadrático médio sem o atributo Fedu, e o modelo da árvore de decisão com profundidade 3 menor erro quadrático médio na presença do atributo Fedu.

Por fim, vale destacar que, na hipótese da presença da variável Fedu, o modelo de Floresta Aleatória com 500 árvores apresentou maior recall para a classe 1, onde estão

TABLE XXXI

RESULTADOS CONSOLIDADOS SEM FEDU

Modelo	f1	Mean Squared Error
Regressao Logistica	0.3328	1.6810
Bayesiano Multinomial	0.3508	2.0308
RN (2,2)	0.1277	1.5223
RN (5,5)	0.1720	2.5069
RN (20,20)	0.3389	1.7720
RN (20,20)	0.3389	1.7165
SVM Linear	0.3384	1.7042
SVM Polinomial	0.3114	1.8629
SVM RBF	0.3232	1.8043
Decision Tree	0.2952	1.9337
Decision Tree (D=3)	0.3545	2.1941
Decision Tree (D=5)	0.3571	1.9060
Random Forest 100	0.3729	1.5886
Random Forest 500	0.3836	1.6533

TABLE XXXII

RESULTADOS CONSOLIDADOS COM FEDU

Modelo	f1	Mean Squared Error
Regressao Logistica	0.4373	1.1156
Bayesiano Multinomial	0.3711	1.6980
RN (2,2)	0.1277	1.5223
RN (5,5)	0.1532	1.5193
RN (20,20)	0.4428	0.9399
RN (20,20)	0.4428	1.1002
SVM Linear	0.4812	1.0354
SVM Polinomial	0.4004	1.3020
SVM RBF	0.4098	1.1248
Decision Tree	0.4037	1.2203
Decision Tree (D=3)	0.4967	0.9029
Decision Tree (D=5)	0.4853	0.9985
Random Forest 100	0.5096	1.0431
Random Forest 500	0.5162	0.9877

as mães de alunos com nível de escolaridade mais baixo, classe essa que dentro do target do problema se mostra como a de maior interesse em classificação. Sem essa variável, o modelo de SVM Linear teve o melhor recall da classe 1. Esses resultados reforçam nossa conclusão de que diferentes modelos podem ser escolhidos como os mais adequados dependendo do seu objetivo, de forma que não existe um modelo chave que irá sempre ser a melhor opção.

TABLE XXXIII

APÊNDICE A - DESCRIÇÃO DOS ATRIBUTOS DO CONJUNTO DE DADOS MODIFICADOS

Atributo	Descrição
gênero	gênero do estudante (binário: feminino = 0 ou masculino = 1)
idade	idade do estudante (numérico: 15 a 22 anos)
escola	de qual escola é o estudante (numerico: Gabriel Pereira = 0 ou Mousinho da Silveira = 1)
endereço	o tipo de endereço do estudante (binário: urbano = 1 ou rural = 0)
Pstatus	binário: se os pais moram juntos = 1 ou separados = 0)
Medu	educação da mãe (numérico: de 1 a 4ª)
Mjob	trabalho da mãe (numérico: em casa=0, fora de casa=1)
Fedu	educação do pai (numérico: 1 a 4ª)
Fjob	trabalho do pai (numérico: em casa=0, fora de casa=1)
guardian	responsável pela guarda do estudante (binário: um dos pais=1, outro=0)
famsize	tamanho da família (binário: <= 3 ou >3
famrel	qualidade das relações familiares (numérico: 1 - muito ruim até 5 - excelente)
traveltime	tempo do percurso de casa até a escola (numérico: 1 - <15 min , 2 - 15 a 30 min, 3- 30 min a 1 hora, ou 4 - >1 hora)
studytime	tempo de estudo por semana (numérico: 1 - <2 horas, 2 2 - 2 a 5 horas, 3 - 5 a 10 horas, 4 - >10 horas)
failures	numero de reprovações (numérico: n, se 1<= n <3, se não 4)
schoolsup	apoio escolar extra (binário: sim ou não)
famsup	suporte educacional da família (binário: sim=1 ou não=0)
activities	atividades extracurriculares (binário:sim=1 ou não=0)
paidclass	aulas particulares (binário: sim=1 ou não=0)
internet	acesso a internet em casa (binário: sim=1 ou não=0)
nursery	frequentou o maternal (binário: sim=1 ou não=0)
higher	pretende cursar o ensino superior (binário: sim=1 ou não=0)
romantic	está em algum relacionamento (binário: sim=1 ou não=0)
freetime	tempo livre fora da escola (numérico: 1 - bastante tempo livre até 5 - pouco tempo livre)
goout	sai com amigos (numérico: 1 - muito pouco, até, 5 - bastante)
Walc	consumo de álcool semanal (numérico: 1 - muito pouco até 5 - bastante)
Dalc	consumo de álcool diário (numérico: 1 - muito pouco até 5 - bastante)
health	estado de saúde atual (numérico: 1 - muito ruim até 5 - muito bom)
absences	número de faltas na escola (de 0 até 93)
G1	nota do primeiro período (numérico: 0 a 20)
G2	nota do segundo período (numérico: 0 a 20)
G3	nota do terceiro período (numérico: 0 a 20)

IX. Apêndice B.1 - Código Pré-Processamento R

```r

Pre_load <- function(){

library("pdist")
library("ggplot2")
library("gridExtra")

#Read CSV
Raw_Dataset=read.table("student-por.csv",sep=";",
    header=TRUE)

#Features Transformation (factor to binary)
Raw_Dataset <- Raw_Dataset[,c(-11)] #delete reason
    column
for (j in 1:ncol(Raw_Dataset)){



  if(is.factor(Raw_Dataset[,j])){
    Raw_Dataset[,j] <- as.numeric(Raw_Dataset[,j])
    for (i in 1:nrow(Raw_Dataset)){

      if(j==12){ #Guardian Column
        if( Raw_Dataset[i,j]==3){Raw_Dataset[i,j
            ]=0} else {Raw_Dataset[i,j]=1}
      }

      else if( Raw_Dataset[i,j]==1){Raw_Dataset[i,j
          ]=0} else {Raw_Dataset[i,j]=1}
    } }
  Raw_Dataset[,j] <- as.numeric(Raw_Dataset[,j])
}

#Balancing Dataset
for (i in 1:nrow(Raw_Dataset)){
  if( (Raw_Dataset[i,7]==0)|(Raw_Dataset[i,7]==1)|(
      Raw_Dataset[i,7]==2)) {Raw_Dataset[i,7]=0} #
      Compile Medu
  if( (Raw_Dataset[i,7]==3)|(Raw_Dataset[i,7]==4))
      {Raw_Dataset[i,7]=1} #Compile Medu
  if( (Raw_Dataset[i,8]==0)|(Raw_Dataset[i,8]==1)|(
      Raw_Dataset[i,8]==2)) {Raw_Dataset[i,8]=0} #
      Compile Fedu
  if( (Raw_Dataset[i,8]==3)|(Raw_Dataset[i,8]==4))
      {Raw_Dataset[i,8]=1} #Compile Fedu
}


#Outliers Dettection
scaled.Raw_Dataset <- scale(Raw_Dataset)
boxplot(scaled.Raw_Dataset,las = 2)
mean_distance = vector(length = nrow(Raw_Dataset))
for (i in 1:nrow(Raw_Dataset)){
  euclidian_dist = pdist(scaled.Raw_Dataset,
      indices.A = i, indices.B = c(-i))
  mean_distance[i]=  mean(as.matrix(euclidian_dist)
      )
  outliers_index=which(mean_distance > 10) #10 was
      chosen after an visual analysis of the mean
      values
}

scaled.Raw_Dataset <- scaled.Raw_Dataset[c(-
    outliers_index),]
Raw_Dataset <- Raw_Dataset[c(-outliers_index),]



piecharts <- function(value, pie_label){
  df <- data.frame( variable = c("Ate Educacao
      Primaria (4o Ano)"," 5o ao 9o ano","Escola
      Secundaria","Ensino Superior"), value = c(
      value))

  graf <- ggplot(transform(transform(df, value=
      value/sum(value)), labPos=cumsum(value)-value
      /2),
                 aes(x="", y = value, fill =
                     variable)) +
    geom_bar(width = 1, stat = "identity") +
    scale_fill_manual(values = c("red", "yellow","
        blue", "green", "cyan")) +
    coord_polar(theta = "y") +
    labs(title = pie_label) +
    geom_text(aes(x=1.2, y=labPos, label=scales::
        percent(value)))
  return(graf)
}


#Pie charts
Mother_Edu_Ratio <- summary(as.factor(Raw_Dataset$
    Medu))
Father_Edu_Ratio <- summary(as.factor(Raw_Dataset$
    Fedu))
graf_mom <- piecharts(Mother_Edu_Ratio, "Mother
    Education")
graf_dad <- piecharts(Father_Edu_Ratio, "Father
    Education")
grid.arrange(graf_mom , graf_dad , ncol=2)

  return(scaled.Raw_Dataset)
}
```

Listing 1. Código fonte em R

APÊNDICE B.2 - CÓDIGO PRÉ-PROCESSAMENTO .PY

```
import pandas
import itertools
import numpy as np
from patsy import dmatrices
from prettytable import PrettyTable
from matplotlib import pyplot as plt
from scipy import stats
from sklearn import svm, metrics, tree
from sklearn import naive_bayes as nb
from sklearn import linear_model as lm
from sklearn import cross_validation as cv
from sklearn.ensemble import RandomForestClassifier
from sklearn.grid_search import RandomizedSearchCV
from sklearn.neural_network import MLPClassifier
from sklearn.externals.six import StringIO
import pydotplus
from functools import reduce



# Abrindo o arquivo
df = pandas.read_csv('student-por.csv', sep=';')

# Adaptando os dados (Relatorio 1)
del df['reason']
df['school']   =   df['school'].apply(lambda x: 0
    if x == 'GP' else 1)
df['sex']      =      df['sex'].apply(lambda x: 0
    if x == 'F' else 1)
df['address']  =  df['address'].apply(lambda x: 0
    if x == 'R' else 1)
df['famsize']  =  df['famsize'].apply(lambda x: 0
    if x == 'LE3' else 1)
df['Pstatus']  =  df['Pstatus'].apply(lambda x: 0
    if x == 'T' else 1)
df['Mjob']     =     df['Mjob'].apply(lambda x: 0
    if x == 'at_home' else 1)
df['Fjob']     =     df['Fjob'].apply(lambda x: 0
    if x == 'at_home' else 1)
df['guardian'] = df['guardian'].apply(lambda x: 1
    if x == 'other' else 0)
for i in ['schoolsup', 'famsup', 'paid', '
    activities', 'nursery', 'higher',
'internet', 'romantic']:
    df[i] = df[i].apply(lambda x: 1 if x == 'yes'
        else 0)


# Pie Chart
def pie(df, v, title):
    t = df[v].value_counts().to_dict()
    colors = ['gold', 'yellowgreen', 'lightcoral',
        'lightskyblue', 'mediumslateblue']
    plt.pie(t.values(), labels=t.keys(), colors=
        colors, shadow=True, autopct='%1.1f%%')
    plt.axis('equal')
    plt.savefig(title + 'png', transparent=True)
    plt.close()


# Preparando para Relatorio 2


#pie(df, 'Medu', 'Medu-5classes')
#pie(df, 'Fedu', 'Fedu-5classes')
df['Medu'] = df['Medu'].apply(lambda x: 1 if x == 0
     else x)
df['Fedu'] = df['Fedu'].apply(lambda x: 1 if x == 0
     else x)

#pie(df, 'Medu', 'Medu-4classes')
#pie(df, 'Fedu', 'Fedu-4classes')
del df['Fedu']
colunas = reduce(lambda x, y: x if y == 'Medu' else
     x + ' + ' + y, df.columns)
y, X = dmatrices('Medu ~ ' + colunas, df,
    return_type='dataframe')
y = np.ravel(y)
class_names = range(1,5)


# Confusion Matrix
def plot_confusion_matrix(cm, classes, normalize=
    False, title='Matriz de Confusao',
cmap=plt.cm.Blues):
    cm = cm.astype('float') / cm.sum(axis=1)[:, np.
        newaxis]
    for i in range(len(cm)):
        for k in range(len(cm[i])):
            cm[i][k] = round(cm[i][k]*100, 1)
    plt.imshow(cm, interpolation='nearest', cmap=
        cmap, clim=[0,100])
    plt.title(title)
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)
    plt.colorbar(ticks=[0,25,50,75,100])
    thresh = 70
    for i, j in itertools.product(range(cm.shape
        [0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh
                     else "black")
    plt.tight_layout()
    plt.subplots_adjust(bottom=0.15)
    plt.ylabel('Classe correta')
    plt.xlabel('Classe predita')
tabela = PrettyTable(['Modelo', 'f1', 'Mean Squared
     Error'])



# Regressao Logistica
# Modelo
logistic = lm.LogisticRegression().fit(X, y)
predicted = cv.cross_val_predict(logistic, X, y, cv
    =10)
# Cross Validation
scores = cv.cross_val_score(lm.LogisticRegression()
    , X, y, cv=10,
scoring='f1_weighted')
print ('Regressao Logistica')
print (scores.mean())
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'w') as text_file:
    text_file.write(cr)
    text_file.write('\n')
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['Regressao Logistica', scores.mean
    (), mse])
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_rl_n.png', transparent=True)
plt.close()
print ('\n')


# Classificador Bayesiano (Multinomial)
# Modelo
```

```
bayes = nb.MultinomialNB()
bayes = bayes.fit(X, y)
predicted = cv.cross_val_predict(bayes, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(nb.MultinomialNB(), X, y, cv=10, scoring='f1_weighted')
print ('bayesiano multinomial')
print (scores.mean())
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['Bayesiano Multinomial', scores.mean(), mse])
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_bayes_n.png', transparent=True)
plt.close()



##MLP 2,2
## Modelo
mlp = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(2,2), random_state=1)
mlp = mlp.fit(X, y)
predicted = cv.cross_val_predict(mlp, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(2,2), random_state=1), X, y, cv=10, scoring='f1_weighted')
print ('redes neurais 22')
print (scores.mean())
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['RN (2,2)', scores.mean(), mse])
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_rn_n22.png', transparent=True)
plt.close()



##MLP 5,5
## Modelo
mlp = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5,5), random_state=1)
mlp = mlp.fit(X, y)
predicted = cv.cross_val_predict(mlp, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5,5), random_state=1), X, y, cv=10, scoring='f1_weighted')
print ('redes neurais 55')
print (scores.mean())
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['RN (5,5)', scores.mean(), mse])
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')

# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_rn_n55.png', transparent=True)
plt.close()


##MLP 10,10
## Modelo
mlp = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(10,10), random_state=1)
mlp = mlp.fit(X, y)
predicted = cv.cross_val_predict(mlp, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(20,20), random_state=1), X, y, cv=10, scoring='f1_weighted')
print ('redes neurais 1010')
print (scores.mean())
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['RN (20,20)', scores.mean(), mse])
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_rn_n1010.png', transparent=True)
plt.close()



##MLP 20,20

## Modelo
mlp = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(20,20), random_state=1)
mlp = mlp.fit(X, y)
predicted = cv.cross_val_predict(mlp, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(20,20), random_state=1), X, y, cv=10, scoring='f1_weighted')
print ('redes neurais 2020')
print (scores.mean())
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['RN (20,20)', scores.mean(), mse])
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
```

```
    text_file.write(cr)
    text_file.write('\n')
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_rn_n2020.png', transparent=True)
plt.close()


# SVM Linear
# Modelo
svc = svm.SVC(kernel='linear')
svc = svc.fit(X, y)
predicted = cv.cross_val_predict(svc, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(svm.SVC(kernel='linear'
    ), X, y, cv=10,
scoring='f1_weighted')
print ('svm linear')
print (scores.mean())
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['SVM Linear', scores.mean(), mse])
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_svm_l_n.png', transparent=True)
plt.close()


# SVM Polinomial
# Modelo
poly_svc = svm.SVC(kernel='poly', degree=3)
poly_svc = poly_svc.fit(X, y)
predicted = cv.cross_val_predict(poly_svc, X, y, cv
    =10)
# Cross Validation
scores = cv.cross_val_score(svm.SVC(kernel='poly',
    degree=3), X, y, cv=10,
scoring='f1_weighted')
print ('svm polinomial')
print (scores.mean())
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['SVM Polinomial', scores.mean(),
    mse])
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_svm_p_n.png', transparent=True)
plt.close()



# SVM RBF
# Modelo
rbf_svc = svm.SVC(kernel='rbf')
rbf_svc = rbf_svc.fit(X, y)
predicted = cv.cross_val_predict(rbf_svc, X, y, cv
    =10)
# Cross Validation
scores = cv.cross_val_score(svm.SVC(kernel='rbf'),
    X, y, cv=10,
scoring='f1_weighted')
print ('svm rbf')
print (scores.mean())
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['SVM RBF', scores.mean(), mse])
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_svm_rbf_n.png', transparent=True)
plt.close()


# Decision Tree
# Modelo
clf = tree.DecisionTreeClassifier(random_state=1)
clf = clf.fit(X, y)
predicted = cv.cross_val_predict(clf, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(clf, X, y, cv=10)
print ('decision tree')
print (scores.mean())
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['Decision Tree', scores.mean(), mse
    ])
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_tree_n.png', transparent=True)
plt.close()


# Decision Tree 3
# Modelo
clf = tree.DecisionTreeClassifier(max_depth=3,
    random_state=1)
clf = clf.fit(X, y)
predicted = cv.cross_val_predict(clf, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(clf, X, y, cv=10)
print ('decision tree')
print (scores.mean())
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['Decision Tree (D=3)', scores.mean
    (), mse])
```

```
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_tree3_n.png', transparent=True)
plt.close()



# Decision Tree 5
# Modelo
clf = tree.DecisionTreeClassifier(max_depth=5,
    random_state=1)
clf = clf.fit(X, y)
predicted = cv.cross_val_predict(clf, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(clf, X, y, cv=10)
print ('decision tree')
print (scores.mean())
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['Decision Tree (D=5)', scores.mean
    (), mse])
teste = list(df.columns[1::])
teste = [0] + teste

dot_data = tree.export_graphviz(clf,out_file = None
    ,
class_names=['1','2','3','4'], rounded=True,
    feature_names=teste)


graph = pydotplus.graph_from_dot_data(dot_data)
graph.write_pdf('teste5.pdf')


# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('cm_tree5_n.png', transparent=True)
plt.close()



# Random Forest 100
# Modelo
clf = RandomForestClassifier(n_estimators=100)
clf = clf.fit(X, y)
predicted = cv.cross_val_predict(clf, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(clf, X, y, cv=10)
print ('Random Forest 100')
print (scores.mean())
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['Random Forest 100', scores.mean(),
     mse])
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('randomforest100.png', transparent=True
    )
plt.close()



# Random Forest 500
# Modelo
clf = RandomForestClassifier(n_estimators=500)
clf = clf.fit(X, y)
predicted = cv.cross_val_predict(clf, X, y, cv=10)
# Cross Validation
scores = cv.cross_val_score(clf, X, y, cv=10)
print ('Random Forest 500')
print (scores.mean())
# Avaliacao
cnf_matrix = metrics.confusion_matrix(y, predicted)
cr = metrics.classification_report(y, predicted)
print (cr)
with open('cr.txt', 'a') as text_file:
    text_file.write(cr)
    text_file.write('\n')
mse = metrics.mean_squared_error(y, predicted)
tabela.add_row(['Random Forest 500', scores.mean(),
     mse])
# Matriz de Confusao Normalizada
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=
    class_names, normalize=True,
title='Matriz de Confusao Normalizada')
plt.savefig('randomforest500.png', transparent=True
    )
plt.close()
tabela.align = "l"
print (tabela)
with open('tabela.txt', 'w') as text_file:
    text_file.write(tabela.get_string())
```

	nobs	Minimum	Maximum	1. Quartile	3. Quartile	Mean	Median	Stdev
school	649	0	1	0	1	0.348228	0	0.476776
sex	649	0	1	0	1	0.409861	0	0.492187
age	649	15	22	16	18	16.744222	17	1.218138
address	649	0	1	0	1	0.696456	1	0.460143
famsize	649	0	1	0	1	0.29584	0	0.456771
Pstatus	649	0	1	1	1	0.876733	1	0.328996
Medu	649	0	4	2	4	2.514638	2	1.134552
Fedu	649	0	4	1	3	2.306626	2	1.099931
Mjob	649	0	1	1	1	0.791988	1	0.406199
Fjob	649	0	1	1	1	0.935285	1	0.246212
guardian	649	0	1	1	1	0.764253	1	0.424792
traveltime	649	1	4	1	2	1.568567	1	0.74866
studytime	649	1	4	1	2	1.930663	2	0.82951
failures	649	0	3	0	0	0.22188	0	0.593235
schoolsup	649	0	1	0	0	0.104777	0	0.306502
famsup	649	0	1	0	1	0.613251	1	0.487381
paid	649	0	1	0	0	0.060092	0	0.237841
activities	649	0	1	0	1	0.485362	0	0.500171
nursery	649	0	1	1	1	0.802773	1	0.398212
higher	649	0	1	1	1	0.893683	1	0.308481
internet	649	0	1	1	1	0.767334	1	0.422857
romantic	649	0	1	0	1	0.368259	0	0.482704
famrel	649	1	5	4	5	3.930663	4	0.955717
freetime	649	1	5	3	4	3.180277	3	1.051093
goout	649	1	5	2	4	3.1849	3	1.175766
Dalc	649	1	5	1	2	1.502311	1	0.924834
Walc	649	1	5	1	3	2.280431	2	1.28438
health	649	1	5	2	5	3.53621	4	1.446259
absences	649	0	32	0	6	3.659476	2	4.640759
G1	649	0	19	10	13	11.399076	11	2.745265
G2	649	0	19	10	13	11.570108	11	2.913639
G3	649	0	19	10	14	11.906009	12	3.230656

Fig. 36. Apêndice C - Parâmetros Estatísticos do conjunto de dados

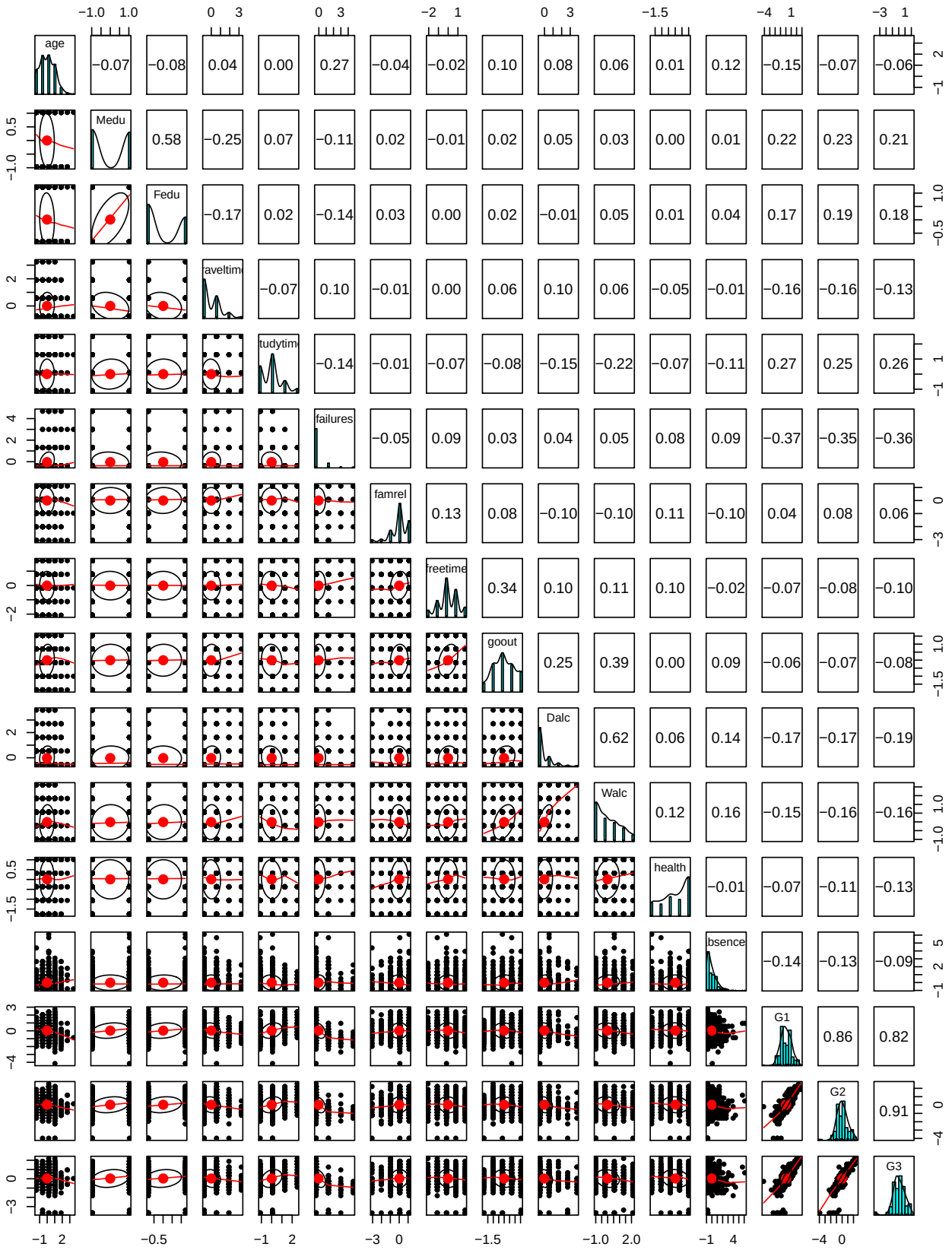

Fig. 37. Apêndice D - Histograma Variáveis numéricas

REFERENCES

[1] Eurostat, 2007. Early school-leavers. http://epp.eurostat.ec.europa.eu/ (Acesso em Abr. 20, 2018).

[2] Ma Y.; Liu B.; Wong C.; Yu P.; and Lee S., 2000. Targeting the right students using data mining. In Proc. of 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston, USA, 457–464.

[3] Luan J., 2002. Data Mining and Its Applications in Higher Education. New Directions for Institutional Research, 113, 17–36.

[4] Kaur P., Singh M., Josan G. S., Classification and prediction based data mining algorithms to predict slow learners in education sector, Procedia Computer Science 57 (2015) 500 – 508.

[5] V.Ramesh, P.Parkavi, K.Ramar(2013), "Predicting student performance: A statistical and datamining approach",International journal of computer applications , Volume 63- no. 8, pp 35-39.

[6] Ahmed M. A., Rizaner A., Ulusoy A. H., Using data mining to predict instructor performance, Procedia Computer Science 102 (2016) 137 – 142.

[7] Minaei-Bidgoli B, Punch WF. Using genetic algorithms for data mining optimization in an educational web based system. Genetic and Evolutionary Computation, Springer Berlin Heidelberg, 2003, p. 2252-2263.

[8] P.Cortez, A.Silva (2008),"Using Data Mining To Predict Secondary School Student Performance", InEUROSIS, A.BritoandJ. Teixeira (Eds.), pp 5-12.

[9] Kotsiantis S.; Pierrakeas C.; and Pintelas P., 2004. Pre- dicting Students' Performance in Distance Learning Using Machine Learning Techniques. Applied Artifi- cial Intelligence (AAI), 18, no. 5, 411–426.

[10] Sorour SE, Mine T, Goda K, Hirokawa S. Estimation of student performance by considering consecutive lesson. 4th International Congress on Advanced Applied Informatics, 2015, p.121-126.

[11] Nguyen Thai-Nghe, Andre Busche,LarsSchmidt Thieme(2009), "Improving Academic Performance Prediction by Dealing with Class Imbalance", Ninth International Conference on Intelligent Systems Design and Applications.

[12] Shahiri A. M, Husain W., Rashid N. A., The Third Information Systems International Conference A Review on Predicting Student's Performance using Data Mining Techniques, Procedia Computer Science 72 (2015) 414 – 422.