Pós-processamento da base *Student Alcohol Consuption*: Análise dos dados

Jean Silva

12 de julho de 2016

Sumário

Capítulo 1

Introdução

A base que está sendo analisada neste trabalho é a *Student Alcohol Consuption* que reúne dados de alunos de duas escolas em relação ao seu rendimento escolar, acompanhamento dos pais e inclusive da própria escola. A base foi obtida por (CORTEZ; SILVA, 2008) para tentar prever a influência do consumo alcoólico no rendimento dos estudantes de escola secundária. Essa análise pode permitir que se tome medidas de prevenção para aumentar o rendimento dos estudantes, uma vez que a causa da queda do mesmo for detectada.

O consumo de bebida alcoólica tem se popularizado cada vez mais nos dias autuais, onde temos uma cultura em que consumir bebida alcoólica é socializar e que incetiva o seu consumo através de comerciais em rádio ou TV ou *merchandising* através de campanhas, promoções, cartazes, e etc.

Capítulo 2

Descrição do Problema e base de dados

2.1 A base

A base *Student Alcohol Consuption* foi extraída do UCI *Machine Leaning Repository* [1]. A base tem 1044 instâncias, 32 atributos diversificados, sendo 21 categóricos e 11 numéricos. A base não possui dados faltosos. A Tabela 2.1 mostra a lista de atributos, a descrição e seu tipo, é possível ver também, no campo tipo, as escalas dos atributos numéricos. A Tabela 2.2 mostra o número de classes existentes para cada atributo categórico. O rendimento escolar (G1, G2 e G3) é classe para essa base (vide Tabela 2.1). Para os experimentos aqui consideramos G1 e G2 como atributos da base e G3 o conjunto de dados da classe.

2.2 Ferramentas Utilizadas

Para os experimentos realizados aqui, foram implementados na linguagem de programação Python 2.7.6 com o auxílio da bibliotecas: Pandas[2], NumPy[3], Scikit-learn[4] e Matplotlib[5]. O Weka 3.8 também foi utilizado[6].

[1] A UCI é um site que armazena bases de uso livre para pesquisas e testes e pode ser acessada pelo link: ⟨http://archive.ics.uci.edu/ml/index.html⟩.

[2] Biblioteca *open source* para manipulação e análise de dados. Pode ser encontrada em: ⟨http://pandas.pydata.org/⟩

[3] Biblioteca para computação científica e manipulação de dados. Pode ser encontrada em: ⟨http://www.numpy.org/⟩

[4] Biblioteca *open source* para mineração de dados e análise de dados com suporte à aprendizado de máquina. Pode ser encontrada em: ⟨http://scikit-learn.org/stable/⟩

[5] Biblioteca *open source* para renderização de gráficos 2D de qualidade. Pode ser encontrada em: ⟨http://matplotlib.org/⟩

[6] Software que reúne uma coleção de algoritmos para mineração de dados. Pode ser baixado em: ⟨http://www.cs.waikato.ac.nz/ml/weka/⟩

Tabela 2.1: Dados sobre os atributos da base.

Atributo	**Descrição**	**Tipo**
school	Escola do estudante	Binário: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira
sex	Sexo do estudante	Binário: 'F' - female or 'M' - male
age	Idade do estudante	Numérico: de 15 à 22
address	Endereço do estudante	Binário: 'U' - urban or 'R' - rural
famsize	Tamanho da família	Binário: 'LE3' - *leq* 3 ou 'GT3' - >3
Pstatus	Estatus de convivência dos pais	Binário: 'T' - vivem juntos or 'A' - seprados
Medu	Eduação da Mãe	Numérico: 0 - nenhum 1 - Primário (4o ano), 2 – 5o à 9o ano, 3 – Educação Secundária ou 4 – Ensino Superior
Fedu	Educação do Pai	Numérico: 0 - nenhum 1 - Primário (4o ano), 2 – 5o à 9o ano, 3 – Educação Secundária ou 4 – Ensino Superior
Mjob	Trabalho da mãe	Nominal: "teacher"(Professor), "health"care related (Saúde), civil "services"(e. g. Administração ou Polícia), "at_home"(em casa) or "other"(outro))
Fjob	Trabalho do pai	Nominal: "teacher"(Professor), "health"care related (Saúde), civil "services"(e. g. Administração ou Polícia), "at_home"(em casa) or "other"(outro))
reason	Por que estuda nessa escola?	Nominal: perto de casa - "home", repultação da escola - "reputation", preferência pelo curso - "course"ou outro - "other")
guardian	Responsável pelo estudante	Nominal: "mother"(Mãe), "father"(Pai) ou "other"(Outro)
traveltime	Tempo que leva de casa à escola	Numérico: 1 - <15 min., 2 - 15 à 30 min., 3 - 30 min. à 1h, ou 4 - >1 h
studytime	Tempo de estudo semanal	Numérico: 1 - <2 h, 2 - 2 à 5 h, 3 - 5 à 10 hs, ou 4 - >10 h
failures	Número de reprovações	Numérico: n se $1 \leq n < 3$, se não 4
schoolsup	Tem suporte extra educacional	Binário: "yes"(Sim) ou "no"(Não)
famsup	Tem suporte educacional por parte da família	Binário: "yes"(Sim) ou "no"(Não)
paid	Aulas extras pagas de Matemática ou Português	Binário: "yes"(Sim) ou "no"(Não)
activities	Atividades extra-curriculares	Binário: "yes"(Sim) ou "no"(Não)
nursery	Frequentou Maternal?	Binário: "yes"(Sim) ou "no"(Não)
higher	Pretender ingressar na faculdade?	Binário: "yes"(Sim) ou "no"(Não)
internet	Tem acesso a internete em casa?	Binário: "yes"(Sim) ou "no"(Não)
romantic	Está envolvido em um relacionamento romêntico?	Binário: "yes"(Sim) ou "no"(Não)
famrel	Qualidade do relacionamento familiar	Numérico: A partir de 1 - Muito ruim à 5 - excelente)
freetime	Tempo livre depois da escola	Numérico: A partir de 1 - Muito baixo até 5 - Muito alto
goout	Sai com os amigos	Numérico: A partir de 1 - Muito baixo até 5 - Muito alto
Dalc	Consumo de álcool diário	Numérico: A partir de 1 - Muito baixo até 5 - Muito alto
Walc	Consumo de álcool no final de semana	Numérico: A partir de 1 - Muito baixo até 5 - Muito alto
health	Estatus atual de saúde	Numérico: A partir de 1 - Muito baixo até 5 - Muito alto
absences	Número de falta na escola	Numérico: 0 à 93
G1	Rendimento no Primeiro ano	Numérico: 0 à 20
G2	Rendimento no Segundo ano	Numérico: 0 à 20
G3	Rendimento no Terceiro ano	Numérico: 0 à 20

Tabela 2.2: Número de classes existentes para cada atributo categórico.

Atributo	**# classes**
school	2
sex	2
address	2
famsize	2
Pstatus	2
Medu	5
Fedu	5
Mjob	5
Fjob	5
reason	4
guardian	3
traveltime	4
studytime	4
schoolsup	2
famsup	2
paid	2
activities	2
nursery	2
higher	2
internet	2
romantic	2

2.3 Pré-processamento dos dados

Foi realizado um pré-processamento nos dados. A base foi reduzida em relação à seus atributos e em relação à dados (número de instâncias).

2.3.1 Redução de Instâncias da base

Os tamanhos das bases analisadas foram 5%, 10%, 20%, ..., 100% do tamanho original. As instâncias removidas foram escolhidas aleatoriamente. O gráfico da Figura 2.1 mostra a execução da Árvore de Decisão (AD) e k-NN (k-ésimo Vizinho mais Próximo do Inglês *k-Nearest Neighbors*), onde o eixo x é o tamanho da base (número de instâncias) e o y é acurácia do classificador.

2.3.2 Redução de atributos da base

Para a redução de instâncias 4 técnicas foram utilizadas, são elas:

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Extração aleatória de atributos

- Extração baseada na matriz de correlação

Na figura 2.3.2 temos os resultados da execução dos classificadores k-NN e da AD para as bases de dados com 5%, 10%, 20%, ..., 100% de atributos da base original para as 4 técnicas. O eixo x é a área de cobertura (*variance covered*) base original e o y é acurácia do classificador.

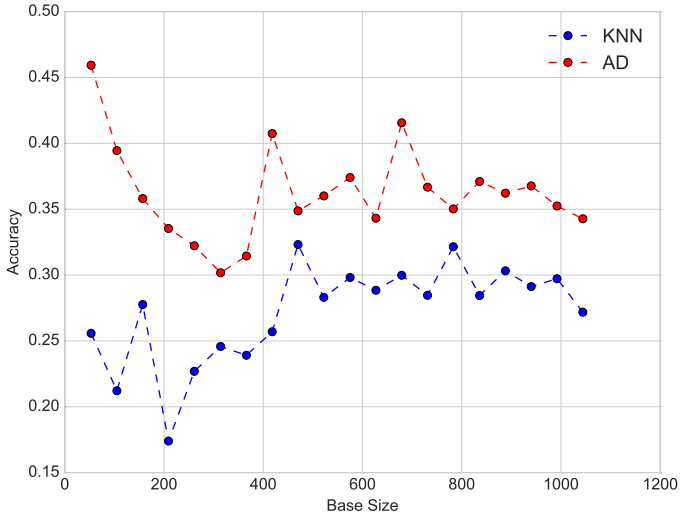


Figura 2.1: Aplicação do k-NN e AD sobre bases geradas através da redução aleatória de instâncias.

As Tabelas 2.3 e 2.4 resumem os experimentos de redução da base reunindo as bases reduzidas que obtiveram os melhores resultados para os classificadores AD e k-NN quando da aplicação do métodos descritos anteriormente sobre as mesmas. A Tabela 2.3 mostra os resultados de redução de atributos, enquanto que a Tabela 2.4 mostra os resultados referentes a redução de instâncias.

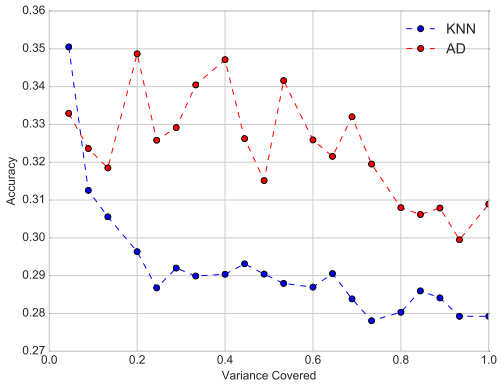
De todos os resultados a base que teve a melhor acurácia dentre todas foi a gerada pela redução de instâncias com 53 instâncias a AD obteve uma acurácia de 47% (vide Tabela 2.3 e 2.4).

Tabela 2.3: Melhores resultados dos experimentos de redução da base pelo número de atributos

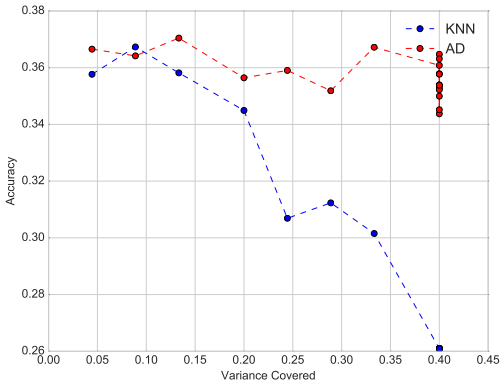
	Técnica							
	PCA		LDA		Aleatório		Correlção	
	Variance Covered	Acurácia	Variance Covered	Acurácia	Variance Covered	Acurácia	Variance Covered	Acurácia
K-NN	5% (2)	35,0%	10% (5)	36,7%	10% (5)	38,0%	100% (45)	35,0%
AD	20% (9)	34,9%	15% (7)	37,2%	10% (5)	43,5%	100% (45)	25,4%

Tabela 2.4: Melhores resultados dos experimentos de redução da base pelo número de instâncias

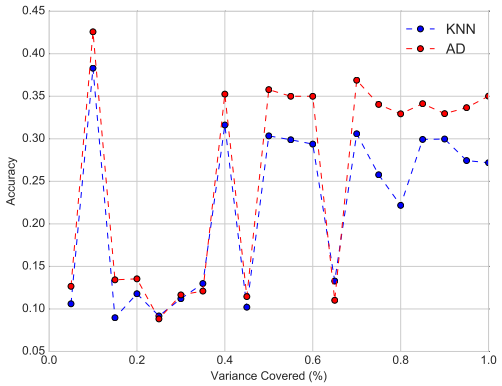
	Acurácia	# instâncias
K-NN	32,5%	470
AD	47,0%	53



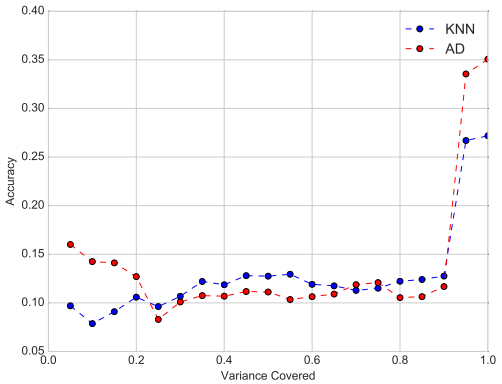
(a) Aplicação do k-NN e AD sobre bases com atributos reduzidos geradas pelo PCA.



(b) Aplicação do k-NN e AD sobre bases com atributos reduzidos geradas pelo LDA.



(c) Aplicação do k-NN e AD sobre bases com atributos reduzidos de forma aleatória.



(d) Aplicação do k-NN e AD sobre bases com atributos reduzidos de acordo com sua correlação média de cada atributo.

Figura 2.2: Aplicação do k-NN e AD sobre as bases reduzidas por cada técnica, respectivamente: PCA (2.2(a)), LDA (2.2(b)), seleção aleatória (2.2(c)) e baseada na correlação média de cada atributo (2.2(d)).

Capítulo 3

Experimentos

Neste capítulo serão analisados os algoritmos supervisionados: redes neurais (*Multi-Layer Perceptron – MLP*), k-NN, Árvores de Decisão (AD) e Naive Bayesian. A base utilizada para os experimentos será a melhor base que foi encontrada na pré-processamento, na seção 2.3, precisamente a que contém 53 instâncias de 1044 e todos os 45 atributos da base original. A justificativa dessa escolha é que base original tinha taxa de acertos de 34% para AD e 22% para o k-NN.

3.1 Métodos de aprendizado supervisionados

3.1.1 Análise sobre o k-NN

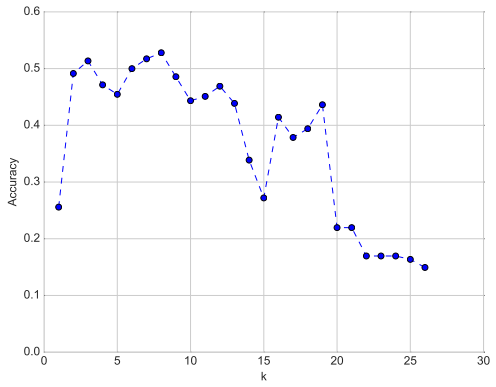
Impacto do valor de k sobre o resultado

A Figura 3.1.1 mostra os resultados de uma análise do valor de k para k-NN. Os resultados gráfico da Figura 3.1(b) são referentes aos dados que foram escalonados no intervalo [0, 1], já na Figura 3.1(a) os dados originais foram usados. É possível perceber que o melhor valor de k é $k = 8$, com uma taxa de acerto de 53%, quando não escalonamos os valores, e quando do escalonamento, o melhor é $k = 25$ ou $k = 26$ com 38% de taxa de acerto.

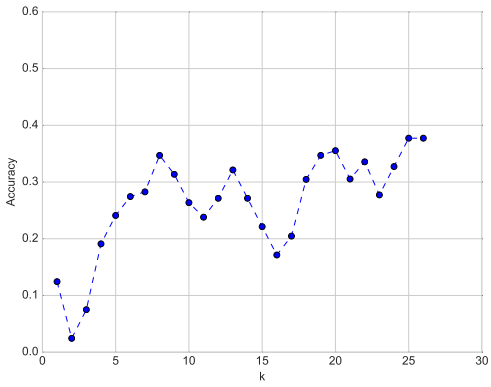
Impacto do parâmetro *distanceweight*

No primeiro experimento, na seção anterior, cada instância tem igual influência, e a mesma é classificada de acordo com aquela classe que possui mais representantes. No caso do k-NN com *distance weight* cada vizinho tem um peso, que é inversamente proporcional à sua distância. Dessa forma os pesos são atribuídos de acordo com sua relevância, podendo trazer uma melhora significativa à taxa de acerto do algoritmo.

A Figura 3.1.1 mostra os resultados do k-NN com *distance weight*. A Figura 3.2(b) mostram os resultados obtidos quando o classificador foi executado com os dados escalonados no intervalo [0, 1], já na Figura 3.1(a) os dados originais foram usados.

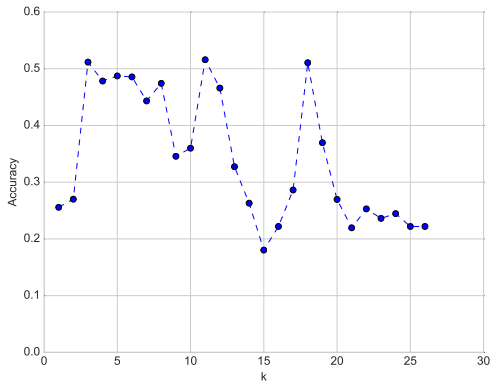


(a) Análise de impacto do valor de k sobre o resultado do k-NN.

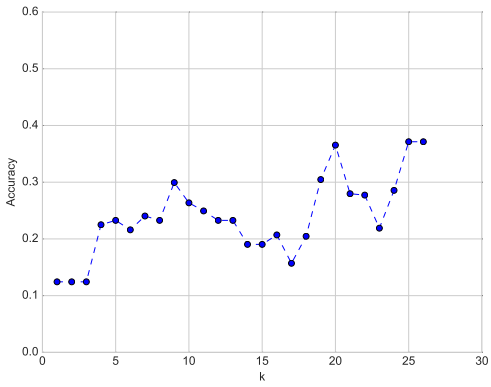


(b) Análise de impacto do valor de k sobre o resultado do k-NN, com os atributos escalonados.

Figura 3.1: Análise de impacto do valor de k sobre o resultado do k-NN. Figura 3.1(a) análise com os valores não escalonados, 3.1(b) com os valores escalonados.



(a) Análise de impacto do valor de k sobre o resultado do k-NN com *distance weight*.



(b) Análise de impacto do valor de k sobre o resultado do k-NN com *distance weight* e os atributos escalonados.

Figura 3.2: Análise de impacto do valor de k sobre o resultado do k-NN com *distance weight*. Figura 3.2(a) análise com os valores não escalonados, 3.2(b) com os valores escalonados.

3.1.2 Análise sobre a Árvore de decisão

Uma vez que a atual versão da lib Python `scikit-learn 0.17` não fornece a função de poda, para esse experimento, usou-se o Weka. Para a base reduzida, podar ou não a árvore não fez diferença no resultado final, a taxa de acerto de 33.96% foi a mesma com e sem poda. Pôde-se perceber que árvore sem poda e com poda fornecida pelo weka foram as mesmas, ambas com 12 folhas e 23 nós (tamanho da árvore). E nesse caso não é possível saber se havia *Overfitting* antes da poda.

3.1.3 Análise sobre a Naive Bayesian Leaning

A tabela 3.1 mostra os resultados referentes aos experimentos, onde variamos os parâmetros: `useKernelEstimator` e `useSupervisedDiscretization`. A partir da análise desses resultados pode-se perceber que dados numéricos estavam obedecendo uma distribuição normal, uma vez que ao setar o parâmetro `useKernelEstimator` = *true*, o resultado piorou. A discretização não impactou na taxa de acerto do Naive Bayes.

Tabela 3.1: Resultados dos experimentos variando valores de parâmetros para o Naive Bayes.

useKernelEstimator	useSupervisedDiscretization	Acurácia
false	false	32.0755%
true	false	30.1887%
false	true	32.0755%

3.1.4 Análise sobre Redes Neurais

Nessa fase de experimentos, o classificador *Multi-Layer Perceptron* (MLP) foi utilizado. Foram feitos treinamentos usando o *backpropagation* padrão e *momentum* = 0.8, variando-se os seguintes parâmetros:

- Quantidade máxima de iterações
- Quantidade de neurônios escondidos da rede, com apenas uma camada
- Taxa de aprendizado.

Três valores foram utilizados para cada parâmetro, a saber: Quantidade máxima de iterações = {100, 1000, 10000}, Taxa de aprendizado = {0.9, 0.05, 0.008} e quantidade de neurônios = {33, 50, 100}. O primeiro valor de número de neurônios na camada intermediária foi obtido pela seguinte fórmula:

$$\frac{n_{atrrs} + n_c}{2}$$ onde n_{atrrs} é o número de atributos e n_c é o número de classes.

Os parâmetros foram combinados formando 27 configurações de parâmetros. A Tabela 3.2 mostra a taxa de acerto obtida para cada conjunto de parâmetros. Cada conjunto de parâmetros foi rodado 30 vezes, e o resultado final é o melhor valor dentre os 30. O método usado no experimento foi o *2-fold-cross-validation*.

A melhor solução presente na Tabela 3.2 é: taxa de aprendizado = 0.008, número de neurônios = 33 e número de iterações = 100, com uma taxa de acerto de $\approx 32\%$. Esse conjunto de parâmetros foi submetido à outro experimento, 30 execuções para cada linha, desta

vez usando o *10-fold-cross-validation*. A Tabela 3.3 mostra os resultados das 30 execuções variando-se o *seed* para inicialização dos pesos na MLP, e a Tabela 3.4 mostra a média, o desvio padrão e o melhor resultado, que foi uma taxa de acerto de $\approx 47\%$, destas últimas 30 execuções sobre o melhor conjunto de parâmetros encontrado.

Tabela 3.2: Resultados dos 27 conjuntos de parâmetros da MLP.

Taxa de aprendizado	**# neurônios**	**# Iterações**	**Acurácia**
0.9	33	100	0.2086956522
0.9	33	1000	0.2086956522
0.9	33	10000	0.2086956522
0.9	50	100	0.1434782609
0.9	50	1000	0.1434782609
0.9	50	10000	0.1434782609
0.9	100	100	0.2355072464
0.9	100	1000	0.2355072464
0.9	100	10000	0.2355072464
0.05	33	100	0.302173913
0.05	33	1000	0.302173913
0.05	33	10000	0.302173913
0.05	50	100	0.2688405797
0.05	50	1000	0.2688405797
0.05	50	10000	0.2688405797
0.05	100	100	0.2239130435
0.05	100	1000	0.2239130435
0.05	100	10000	0.2239130435
0.008	33	100	0.3173913043
0.008	33	1000	0.3123188406
0.008	33	10000	0.3123188406
0.008	50	100	0.3173913043
0.008	50	1000	0.3173913043
0.008	50	10000	0.3173913043
0.008	100	100	0.2739130435
0.008	100	1000	0.2739130435
0.008	100	10000	0.2739130435

A Figura 3.1.4 mostra uma análise da variação dos parâmetros citados anteriormente, fixando 2 e variando 1. Os parâmetros utilizados são os melhores encontrados no inicio do experimento (taxa de aprendizado = 0.008, número de neurônios = 33 e número de iterações = 100).

A Figura 3.3(a) mostra os resultados referente à variação da taxa de aprendizado. Os valores utilizados foram {0.001, 0.002, ..., 0.009} e {0.01, ..., 0.09}. É possível perceber que o melhor valar da taxa de aprendizado é 0.008, portanto sem melhoras.

Tabela 3.3: Resultados da variação do *seed* para reinicialização dos pesos na MLP.

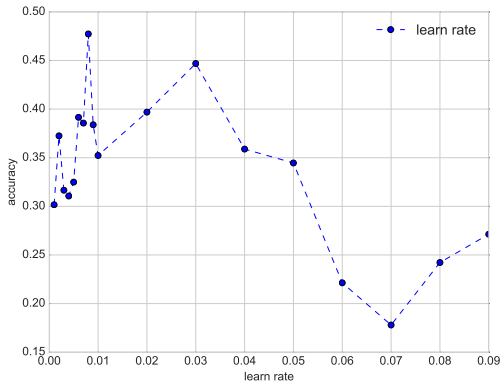
seed	**Acurácia**
0	0.368956044
1	0.2647435897
2	0.2909340659
3	0.3505494505
4	0.2999084249
5	0.3141941392
6	0.268956044
7	0.3356227106
8	0.1987179487
9	0.2665750916
10	0.3576007326
11	0.2898351648
12	0.2981684982
13	0.2546703297
14	0.4242673993
15	0.2112637363
16	0.3999084249
17	0.2558608059
18	0.363003663
19	0.196978022
20	0.2999084249
21	0.2933150183
22	0.2832417582
23	0.2838827839
24	0.3683150183
25	0.4772893773
26	0.0243589744
27	0.2343406593
28	0.4242673993
29	0.304029304

Tabela 3.4: Média, desvio padrão do melhor resultado dentre as 30 execuções da Tabela 3.3.

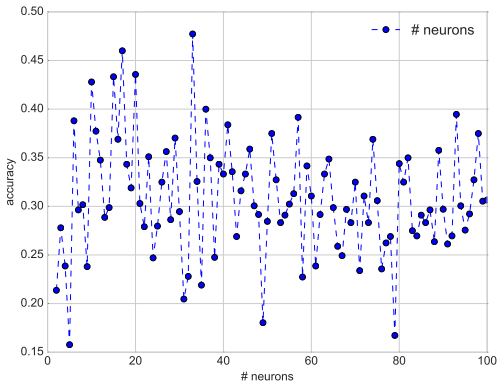
Média	**Desvio Padrão**	**Melhor Resultado**
0.3001221001	0.0837027841	0.4772893773

A Figura 3.3(b) mostra os resultados produzidos pela variação do número de neurônios no intervalo [2, 100]. Pode-se perceber que o melhor valor para o número de neurônios continua sendo 33.

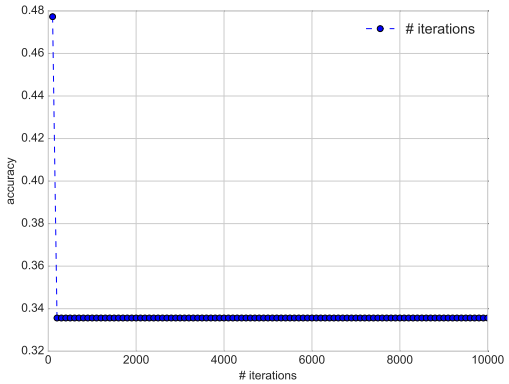
A Figura 3.3(c) mostra os resultados referente à variação do número de iterações. Os valores utilizados foram {100, 200, 300, ..., 10000}. O melhor número de iterações continua sendo 100, e vemos que queda brusca da taxa de acerto a partir do 100, e fica constante.



(a) Taxa de acerto em função da taxa de aprendizado.



(b) Taxa de acerto em função do número de neurônios.



(c) Taxa de acerto em função do número de iterações.

Figura 3.3: Análise de impacto da variação dos parâmetros sobre a taxa de acerto do MLP.

3.2 Comparação entre os métodos supervisionados

A Tabela 3.5 mostra os melhores resultados obtidos dos experimentos sobre os métodos supervisionados, sua taxa de acerto e desvio padrão do *10-fold-cross-validation*. Como se pode ver, o melhor método para esse problema foi k-NN sem o uso do escalonamento dos valores.

Tabela 3.5: Melhores resultado dos experimentos dos métodos supervisionados

Método	**Taxa de acerto**	**Desvio Padrão 10-fold**
K-NN	53.00%	0.3377
AD	33.96%	0.0675
NB	32.07%	0.0704
MLP	48.00%	0.3718

3.3 Métodos de aprendizado não supervisionados

3.3.1 K-Means

O gráfico da Figura 3.4 mostra os resultados da variação do número de grupos (k) do k-means. O intervalo de valores usados para k foi [2, 40]. O melhor valor de índice DB (Davies Bouldin) encontrado foi de 0.55, e com $k = 40$.

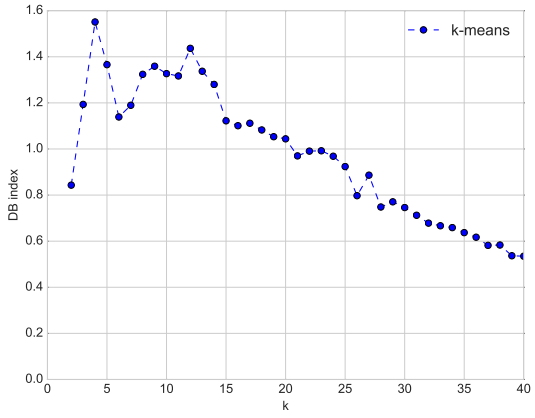


Figura 3.4: Aplicação do k-means com a variação do k (número de grupos).

3.3.2 Hierárquico Aglomerativo

O gráfico da Figura 3.5 mostra os resultados da variação do número de grupos (k) para o método de agrupamento Hierárquico Aglomerativo. O intervalo de valores usados para k também foi [2, 40]. O melhor valor de índice DB encontrado foi de 0.55, e com $k = 40$. O melhor valor de índice DB encontrado foi de 0.57, e com $k = 40$, bem próximo do melhor resultado obtido pelo k-means.

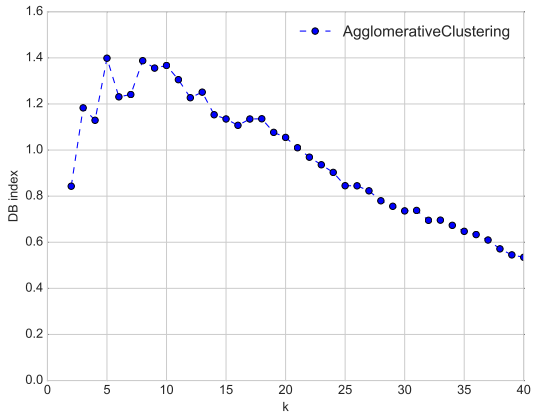


Figura 3.5: Aplicação do método de agrupamento Hierárquico Aglomerativo com a variação do k.

3.4 K-Means versus Hierárquico Aglomerativo

O comportamento dos gráficos das Figuras 3.4 e 3.5 é parecido, mas para verificar o melhor método de agrupamento foi feita uma comparação através do índice CR ou *adjusted rand score* do agrupamento com k = 16 obtido pelo k-means e o agrupamento original, que inclui a classe na base. A mesma coisa foi feita para o método Hierárquico Aglomerativo. O resultado pode ser visto na Tabela 3.6. O melhor método de agrupamento para esta base foi o Hierárquico Aglomerativo, uma vez que seu índice CR se aproxima do 1.0 mais que o do k-means. O valor de $k = 16$ foi escolhido, pois é um bom valor, sendo inferior ao número de classes, no gráfico de ambos os agrupadores, nas seções anteriores é um ponto de vale comum aos dois métodos de agrupamento.

Tabela 3.6: Índices CR obtidos para ambos os métodos de agrupamento.

Método	**Índice CR**
k-means	0.38
Hierárquico Aglomerativo	0.46

Capítulo 4

Sugestões para melhoria dos experimentos

A base atual, tanto a original quando a reduzida conta com 21 classes. A ideia é reduzir o número de classes. As classes representam as categorias de rendimento para um aluno, de 0 à 20. Assim, a sugestão é reduzir de 21 para 4 classes, da seguinte forma:

- Classe 1: 0 à 4
- Classe 2: 5 à 9
- Classe 3: 10 à 14
- Classe 4: 15 à 20

Testes preliminares sobre a basa original foram feitos para avaliar a potencialidade dessa abordagem. Os resultados tiveram melhoras em comparação com abordagem anterior (21 classes). A Tabela 4.1 exibe as taxas de acerto para os classificadores k-NN e AD usando 21 e 4 classes respectivamente.

Tabela 4.1: Tabela comparando os resultado usando o atributo classe com 21 e 4 classes.

	Taxa de aceitação	
	k-NN	AD
21 Classes	27%	34%
4 Classes	72%	82%

De acordo com os experimentos feitos nesta seção, percebe-se que a quantidade de classes usadas nos experimentos dos capítulos anteriores influenciaram fortemente na qualidade dos resultados, dessa forma é possível melhorar ainda mais os mesmos.

Referências Bibliográficas

CORTEZ, P.; SILVA, A. M. G. Using data mining to predict secondary school student performance. EUROSIS, 2008.