LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

Decomposição da densidade estimada de variáveis multimodais utilizando uma função distância: uma nova metodologia para aperfeiçoar a classificação de variáveis binárias

Abner G. de Sá, Giovani Nucci, Mateus P. B. de Sá, Ana Flávia U. S. Macambira
Departamento de Estatística - Universidade Federal da Paraíba
Cidade Universitária, s/n, João Pessoa - PB
abner.sa@posteo.de, giovaninucci@gmail.com,
mateuspbsa@gmail.com,anaflavia@de.ufpb.br

RESUMO

Métodos de classificação estão sendo cada vez mais utilizados nas mais diversas áreas, como por exemplo na área de saúde como auxílio a diagnóstico de imagens médicas e na área de vendas como a identificação de padrão de consumo para a oferta de produtos de acordo com o mesmo. A escolha de um método de classificação não é muito fácil devido ao fato de que muitos são equivalentes em performance. Neste trabalho propomos uma metodologia de decomposição da densidade estimada que pode ser aplicada a qualquer método de classificação da literatura com o objetivo de melhorar a performance de classificação dos mesmos. Os testes foram conduzidos utilizando a distância Euclideana como métrica e foram comparadas as performances dos métodos de árvore de decisão, regressão logística, knn e redes neurais . Os resultados foram bons, aumentando a precisão e/ou elevando o valor do AUC em vários casos.

PALAVRAS CHAVE. Métodos de classificação, decomposição, variáveis binárias.

EST-Estatística

ABSTRACT

Classification methods are being increasingly used in several areas, such as in the health area as an aid to the diagnosis of medical images and in sales as the identification of consumption patterns in order to offer new products according to it . The choice of a classification method is not very easy due to the fact that many are equivalent in performance. In this work we propose a methodology of estimated density decomposition that can be applied to any method of classification of the literature with the objective of improving its classification performance. The tests were conducted using the Euclidean distance as a metric and the performances of the decision tree, logistic regression, knn and neural networks methods were compared. The results were good, increasing the accuracy and / or raising the AUC value in several cases.

KEYWORDS. Classification methods, decomposition, binary variables.

EST

https://proceedings.science/p/106855?lang=pt-br

1. Introdução

A utilização cada vez mais intensa de computadores, celulares, tablets e smartwatches para transações financeiras, entretenimento, formação educacional, como cursos de graduação à distância, cursos de línguas entre outras várias aplicações, gera um volume imenso de dados. Para as empresas é interessante extrair, por exemplo, informações de padrões de consumo dos clientes para que possam oferecer para cada cliente um produto ou serviço mais direcionado ao seu perfil de consumo. A aprendizagem de máquina é um método que está diretamente relacionado ao reconhecimento de padrões e atualmente está sendo amplamente aplicado nas mais diversas áreas, como pode ser visto em: [Tanizaki et al., 2019], [McCoy e Auret, 2019], [Ucci et al., 2019] e em [Gray e Perkins, 2019]. O reconhecimento de padrões faz uso de métodos de classificação de objetos, que os agrupam de acordo com suas características ou atributos. Há vários métodos de classificação disponíveis na literatura, dentre eles o KNN (K-Nearest Neighbours) [Cover e Hart, 2006], Árvores de Decisão [Quinlan, 1986] e Regressão Logística [H. Walker e B. Duncan, 1967], [Cox, 1958].

O KNN é um método não paramétrico que classifica um elemento utilizando alguma função distância como medida de similaridade. Dado um conjunto de teste e uma métrica escolhida, o KNN classifica um novo objeto calculando a distância entre novo objeto e seus k vizinhos mais próximos. O novo objeto será então agrupado no conjunto para o qual obtiver maior proximidade. Um ponto delicado para este método é a escolha do parâmetro k, ou seja, o número de vizinhos que vão ser testados para a classificação do novo objeto. Se muitos vizinhos forem selecionados, a classificação será demorada e se poucos vizinhos forem selecionados, a classificação será rápida, porém a chance de erro será aumentada.

O método de árvores de decisão utiliza um conjunto de regras ou características para classificar os indivíduos até que estejam agrupados em classes consideradas puras, ou seja, somente indivíduos semelhantes em cada classe, ou com baixo nível de impureza. O nível de impureza pode ser medido utilizando-se o índice de Gini. No nó raiz da árvore estão todos os indivíduos novos, ainda não classificados. Uma série de atributos ou características são elencados em ordem de importância. Cada atributo ou característica dá origem a uma ramificação da árvore de decisão até que ao fim do processo, elementos semelhantes pertencem ao mesmo grupo.

Os Modelos Lineares Generalizados ou MLGs, [Nelder e Wedderburn, 1972], são uma classe de modelos baseadas na família exponencial uniparamétrica. Podem ser definidos por uma variável resposta que tenha a distribuição de probabilidade pertencente a família da exponencial, variáveis explicativas independentes que descrevam a estrutura linear do modelo e a função de ligação entre a média da variável resposta e a estrutura linear.

De uma forma geral, podemos dividir a estrutura do MLGs da seguinte forma:

- Componente aleatória que é composta por uma variável aleatória Y com n observações independentes, um vetor de médias μ e uma distribuição pertencente a família das exponenciais;
- Componente sistemático que é composta de variáveis explicativas que produzam um preditor linear η;
- Função de ligação, que é uma função monotônica e diferenciável responsável por ligar as duas componentes acima.

Seja $y = (y_1, ..., y_n)^T$ um vetor de observações referentes às realizações das variáveis aleatórias $Y = (Y_1, ..., Y_n)^T$, independentes identicamente distribuídas (i.i.d.), que tem como a média $\mu = (\mu_1, ..., \mu_n)^T$. Cada Yi segue uma distribuição pertencente a família exponencial definida por:

$$f_Y(y; \theta, \phi) = \exp\left\{\frac{[y\theta - b(\theta)]}{a(\phi)} + c(y, \phi)\right\}, \quad (1)$$

onde $a(.), b(.)$ e $c(.)$ são funções conhecidas, ϕ>0 é denominado parâmetro de dispersão e θ é o parâmetro canônico que caracteriza a distribuição em f_Y. Quando o ϕ é conhecido, a equação (1) representa a família exponencial uniparamétrica indexada por θ.

A função de ligação relaciona o preditor linear η à média μ de um vetor de y. Considere então, a estrutura linear de um modelo de regressão:

$$\eta = X\beta, \tag{2}$$

onde $\eta = (\eta_1, ..., \eta_n)^T$, $\beta = (\beta_1, ..., \beta_n)^T$ e X é uma matriz modelo $n \times p(p < n)$ conhecida de posto p. A função linear η dos parâmetros desconhecidos de β é chamada de preditor linear. Outra característica do componente sistemático de um MLG é que o valor médio μ do vetor y é vinculado ao preditor linear (η,uma função monotônica e diferenciável), e é expressa por:

$$\mu_i = g^{-1}(\eta_i), i = 1, .., n \tag{3}$$

denominando-se g(.) função de ligação. A escolha da função de ligação depende da distribuição de probabilidade usada no modelo.

A ideia da técnica da Regressão Logística é encontrar uma relação entre as características e a probabilidade de um determinado resultado, sendo assim um caso do MLG o *logit*, onde a variável dependente está associada a uma variável aleatória Bernoulli. Se o resultado for binário ou dicotômico, temos a Regressão Logística. Desta forma, se x for uma característica, com base nos exemplos, a técnica dará uma resposta em termos de probabilidade somente de um objeto y satisfazer ou não aquela característica.

Redes neurais são modelos matemáticos inicialmente baseados no comportamento dos neurônios humanos. O neurônio de MacCulloch e Pitts modela um neurônio como uma função que avalia estímulos externos de formas ponderadas, decidindo pela sua ativação ou não.

O método Perceptron consiste em uma rede com neurônios independentes de MacCulloch e Pitts, estimando a ponderação dos neurônios de forma a minimizar o erro de classificação. Sendo demonstrado matematicamente a convergência para problemas linearmente separáveis.

Multi-layer Perceptron (MLP) é uma extensão do modelo Perceptron, desenvolvida por Rumelhard, Hinton e McClelland, introduzindo o conceito de camadas escondidas a rede neural e assim possibilitando a resolução de problemas não linearmente separáveis.

O método consiste em inicializar os pesos, tanto na camada nas camadas de saída e escondida, como valores aleatórios próximos de zero e na repetição de duas fases, forward e *backward*, até a convergência dos pesos.

Representando-se por $v_{i\zeta}$ os pesos na camada escondida e $w_{j\kappa}$ os pesos na camada de saída, na fase *forward*, calcula-se a ativação dos neurônios da camada escondida até a camada de saída. Sendo a função de ativação dada por $g(h) = \tanh(h)$ e $h_\zeta = \sum_i x_i v_{i\zeta}$ a força do estimulo na camada escondida, calcula-se a ativação da camada escondida como $a_\zeta = g(h_\zeta)$ e sendo $h_\kappa = \sum_j a_j w_{j\kappa}$ a força do estimulo na camada de saída, calcula-se a ativação na camada de saída como $y_\kappa = g(h_\kappa)$.

Na fase seguinte, *backward*, calcula-se o erro na ativação do neurônios da camada de saída em direção a camada escondida, atualizando-se os pesos. Onde sendo $\delta_o(\kappa) = (y_\kappa - t_\kappa)y_\kappa(1 - y_\kappa)$ o erro na camada de saída e $\delta_h(\zeta) = a_\zeta(1 - a_\zeta)\sum_k w_\zeta \delta_o(k)$ erro na camada escondida. Ainda na *backward*, atualiza-se os pesos na camada de saída fazendo $w_{\zeta\kappa} \leftarrow w_{\zeta\kappa} - \eta\delta_o(\kappa)a_\zeta$ e os pesos na camada escondida $v_\iota \leftarrow v_\iota - \eta\delta_h(\kappa)x_\iota$, onde η representa uma taxa constante de aprendizagem arbitrária.

https://proceedings.science/p/106855?lang=pt-br

2. Metodologia

A abordagem apresentada neste trabalho consiste em melhorar a qualidade das informações contidas nas variáveis contínuas que apresentam múltiplas modas, ou seja, multimodais. A ideia na qual a abordagem se baseia é a de que várias modas representam vários máximos locais para a função densidade de probabilidade (densidade estimada) definida pelos exemplos de treinamento. O método utilizado para estimar a densidade estimada foi o EDK, apresentado em 2.1. Sendo assim, podemos pensar a densidade estimada como a composição de várias funções densidade de probabilidade, cada uma delas com um ponto de máximo global, ou seja, uma única moda. Desta forma, ao invés de o modelo trabalhar na classificação de uma única variável relacionando-a várias informações misturadas, são criadas novas variáveis, cada uma como fruto da decomposição destas informações para permitir ao modelo ponderar e selecionar qual informação é mais útil. Esta seleção se dá através da função distância. A figura 1 ilustra uma densidade estimada com dois máximos locais definida pela linha contínua e linha pontilhada apresenta a decomposição desta em duas funções densidade de probabilidade, cada uma com seu máximo global, ou seja, com sua única moda.

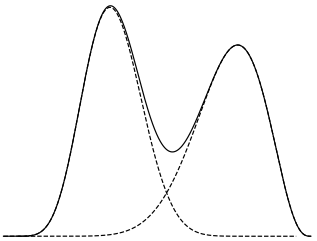


Figura 1: Exemplo de decomposição da densidade estimada

O estudo consiste em em criar uma nova variável para cada moda local observada, individualmente, como a distância dos valores até esta moda, ou seja, sendo $x_i, x_2, \ldots, x_n$, os valores observados da variável x e se seja $\mathbf{M_o}_j$ uma das modas observadas, logo a nova variável x_j^* será $d(x_1, \mathbf{M_o}_j), d(x_2, \mathbf{M_o}_j), \ldots, d(x_n, \mathbf{M_o}_j)$. Por se tratarem de distâncias, o que de fato estamos buscando é testar se a dispersões em torno dessas modas locais trazem mais informação ao ajuste do modelo do que as próprias observações.

O algoritmo 15 apresenta a metodologia proposta.

Para que possamos comparar os resultados das técnicas de classificação sem utilizar o método proposto e utilizando o mesmo, apresentamos o K-fold Cross-Validation.

2.1. Estimativa de Densidade Kernel (EDK)

Sendo $f\left(x\right)$ a densidade da variável aleatória X no ponto x e sendo $S = \{x_1, x_2, \ldots, x_n\}$ o conjunto dos valores observados de $f\left(x\right)$. Dada uma largura de banda h fixa, definindo-se $S_{xh} = \{x_* \in S \mid d(x_*, x) < h\}$ e $n = |S_{xh}|$, a densidade estimada $\hat{f}(x)$ em x pelo método kernel consiste da média aritmética dos valores da funções peso $w_{ni}(x)$ dependentes da função kernel K definida, conforme a equação 4.

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} w_{ni}\left(x\right) \text{ , onde } w_{ni}\left(x\right) = \frac{1}{h}K\left(\frac{x_i - x}{h}\right) \tag{4}$$

Tal método faz as seguintes pressuposições:

(i) as observações $x_1, x_2, \ldots, x_n$ são independentes e identicamente distribuídas – i.i.d.;

Algoritmo 1: Algoritmo de Decomposição

Entrada: $\mathbf{x}$ = vetor de valores observados de dimensão n
Saída: $\mathbf{D}$ = matriz com resultados de dimensão $n \times m$

início
  $EDK_{\mathbf{x}} \leftarrow$ A função Estimativa de Densidade Kernel (EDK) de $\mathbf{x}$
  $\mathbf{M_o} \leftarrow$ O domínio de todas as modas locais na $EDK_{\mathbf{x}}$
  $m \leftarrow dim(\mathbf{M_o})$
  se $m > 1$ **então**
    para $i \in \{1 \dots n\}$ **faça**
      para $j \in \{1 \dots m\}$ **faça**
        $\mathbf{D}_{ij} \leftarrow d(\mathrm{x}_i, \mathbf{M_o}_j)$
      fim
    fim
    retorna ***D***
  senão
    retorna $\mathbf{x}$ pois não há o que ser feito
  fim
fim

(ii) K é uma função simétrica centrada em zero, satisfazendo $\int K(\psi)d\phi = 1$, $\int \psi^2 K(\psi)d\psi = \mu_2 \neq 0$ e $\int K^2(\psi)d\psi < \infty$;

(iii) a segunda derivada de f é continua e limitada para alguma vizinhança de x;

(iv) Sendo h_n a largura de banda para um dado n, $h_n \to 0$ quando $n \to \infty$;

(v) e $nh_n \to \infty$ quando $n \to \infty$.

Existe uma infinidade de funções kernels, entretanto, no presente trabalho estamos utilizando o kernel gaussiano, definido na equação 5. E a distância $d(x_*, x)$ como sendo a distância Euclidiana, onde $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$.

$$K(\psi) = (2\pi)^{-1/2} \exp\left[-\frac{1}{2}(\psi)^2\right] \tag{5}$$

2.2. K-fold Cross-Validation

Esse método estima a taxa de erro verdadeiro de um modelo, quando aplicamos o classificador F a uma amostra de teste independente com a mesma distribuição que o conjunto de treinamento.

O método K-fold Cross-Validation utiliza uma parte dos dados disponíveis para construir o classificador e outra parte diferente dos dados para testá-lo. O procedimento consiste em particionar a amostra em K partes de tamanhos aproximadamente iguais.

Para a k-ésima parte, nós construímos o classificador utilizando as outras $K - 1$ partes, e calculamos a taxa de erro do classificador construído quando predizemos a k-ésima parte. Este procedimento é repetido para $k = 1, \dots, K$ e a taxa de erro final será uma combinação das taxas de erro calculadas em cada uma das K partes.

2.3. Área sob a curva ROC

Como pode ser visto em [Bradley, 1997],o AUC, ou a área sob a curva ROC (Receiver Operating Characteristics), é a forma precisa de avaliar a performance de um método de classificação que "aprende"com base em exemplos apresentados.

ROC é uma curva de probabilidade, de forma que a área sob esta curva, ou seja, o AUC, é a probabilidade do método utilizado ser capaz de identificar um elemento como pertencente a uma entre várias classes. Desta forma, quanto mais próximo de 1 o valor do AUC, mais eficaz é o método de classificação.

Após os dados de treinamento terem sido classificados por algum método, os erros e acertos são contados para cada classe. Para a classificação binária, vamos chamar as classes de positiva e negativa. Para avaliar o método de classificação é utilizada uma tabulação cruzada entre a classe prevista pelo método e a classe a que o elemento realmente pertence no exemplo. Quando um exemplo positivo é classificado como tal, ele é contado como verdadeiro positivo (TP). Caso haja erro e a classificação do exemplo positivo seja dada como negativo, ele é contado como falso negativo (FN). Da mesma forma para os elementos negativos, verdadeiro negativo (TN) e falso positivo (FP).

A curva ROC é desenhada no plano cartesiano e o eixo Y é dado pelos valores TPR (true positive rate), que é a taxa de classificação dos eventos verdadeiramente positivos.

$$TPR = \frac{TP}{TP + FN}$$

Também calculamos a especificidade, dada por

$$\text{Especificidade} = \frac{TN}{TN + FP}$$

O eixo X da curva ROC é dado pelo FPR (false positive rates),

$$FPR = 1 - \text{Especificidade} = \frac{FP}{TN + FP}$$

3. Experimentos Computacionais

Os experimentos computacionais foram realizados em um computador *Intel ® Core™ i7-8750 CPU @ 2.21GHz* com memória *RAM* de 15.8GB. Foram utilizados os bancos de dados [Dua e Graff, 2017a] e [Dua e Graff, 2017b]. A função distância utilizada nos testes foi a distância Euclideana. No método de regressão logística foi utilizada a função de ligação Logit. O método K-fold Cross-Validation foi utilizado com 20 folds. Para cada um deles foi calculada a área sob a curva ROC. As seguintes APIs da linguagem Pyton foram utilizadas: *SciPy*, [Jones et al., 2001–], para a obtenção da EDK, *StatsModels*, [Seabold e Perktold, 2010], para a regressão logística e *scikit-learn*, [Buitinck et al., 2013], para todos os demais métodos.

As Tabelas 1, 2 e 3 estão relacionadas às três variáveis de classificação de cores (R, G e B) do banco de dados [Dua e Graff, 2017a]. As tabelas 4, 5, 6, 7 e 8 estão relacionadas às variáveis de classificação de vinhos do banco de dados [Dua e Graff, 2017b]. Todas as 8 tabelas de resultados tem a mesma configuração. As tabelas possuem três colunas principais. A primeira coluna, **Modelo**, apresenta os nomes dos modelos de classificação existentes na literatura que foram utilizados nos testes. Árvore de decisão, AD, Regressão Logística, representada por Logit sem constante e Logit-C com constante, k-NN, o próprio e MLP A segunda coluna, **Precisão**, apresenta a taxa de acertos obtida na classificação das variáveis em forma de intervalo, onde podemos visualizar a

menor e a maior taxa de acertos obtidos, respectivamente. Esta coluna possui duas subcolunas,que apresentam os resultados dos modelos em seu estado original, subcoluna **Original**, e com a utilização da metodologia de decomposição proposta neste trabalho, subcoluna **Decomposto**. A terceira coluna, **AUC**, do inglês area under the ROC curve, área sob a curva ROC, apresenta as mesmas duas subcolunas **Original**, relativa ao método original e **Decomposto**, que apresenta os resultados obtidos com a utilização da metodologia proposta.

Quando a metodologia proposta neste trabalho e aplicada ao método consegue melhor resultado na classificação das variáveis do que somente o método original apresentado na literatura, os valores estão indicados em negrito nas tabelas. Quando o método com a decomposição proposta atinge a mesma performance que o método original da literatura, os valores estão sublinhados nas tabelas.

Tabela 1: Variável *R* no conjunto de dados *Skin Segmentation*

	Precisão		**AUC**	
Modelo	**Original**	**Decomposto**	**Original**	**Decomposto**
AD	0,8826 ⊢ 0,8944	0,8826 ⊢ 0,8944	0,9260 ⊢ 0,9392	0,9260 ⊢ 0,9392
Logit	0,8644 ⊢ 0,8752	0,7070 ⊢ 0,8853	0,8998 ⊢ 0,9128	0,6188 ⊢ 0,9324
Logit-S	0,7924 ⊢ 0,7925	**0,8759** ⊢ **0,8853**	0,0872 ⊢ 0,1002	**0,8404** ⊢ **0,9324**
k-NN	0,8592 ⊢ 0,8809	0,8533 ⊢ 0,8838	0,8754 ⊢ 0,9046	0,8735 ⊢ 0,8979
MLP	0,7960 ⊢ 0,8829	**0,8747** ⊢ **0,8876**	0,9051 ⊢ 0,9232	**0,9234** ⊢ **0,9350**

Tabela 2: Variável *G* no conjunto de dados *Skin Segmentation*

	Precisão		**AUC**	
Modelo	**Original**	**Decomposto**	**Original**	**Decomposto**
AD	0,7922 ⊢ 0,8944	0,7922 ⊢ 0,8002	0,7959 ⊢ 0,8100	0,7959 ⊢ 0,8100
Logit	0,7924 ⊢ 0,8752	0,7902 ⊢ 0,7968	0,5326 ⊢ 0,5613	**0,7771** ⊢ **0,7910**
Logit-S	0,7590 ⊢ 0,7925	**0,7902** ⊢ **0,7968**	0,4387 ⊢ 0,4674	**0,7771** ⊢ **0,7909**
k-NN	0,7590 ⊢ 0,8809	0,7496 ⊢ 0,7858	0,6906 ⊢ 0,7395	0,7064 ⊢ 0,7333
MLP	0,7960 ⊢ 0,8829	0,7857 ⊢ 0,7967	0,7180 ⊢ 0,7348	**0,7848** ⊢ **0,8027**

Tabela 3: Variável *B* no conjunto de dados *Skin Segmentation*

	Precisão		**AUC**	
Modelo	**Original**	**Decomposto**	**Original**	**Decomposto**
AD	0,7969 ⊢ 0,8027	0,7969 ⊢ 0,8027	0,7973 ⊢ 0,8116	0,7973 ⊢ 0,8116
Logit	0,7924 ⊢ 0,7925	**0,7961** ⊢ **0,8029**	0,5943 ⊢ 0,6152	**0,7787** ⊢ **0,7920**
Logit-S	0,7924 ⊢ 0,7925	**0,7961** ⊢ **0,8029**	0,5943 ⊢ 0,6152	**0,7785** ⊢ **0,7920**
k-NN	0,7540 ⊢ 0,7855	0,7541 ⊢ 0,7846	0,7007 ⊢ 0,7428	**0,7140** ⊢ **0,7432**
MLP	0,7808 ⊢ 0,7979	**0,7907** ⊢ **0,8014**	0,7628 ⊢ 0,7842	**0,7867** ⊢ **0,8040**

Tabela 4: Variável *Alcohol* no conjunto de dados *Wine Quality*

	Precisão		**AUC**	
Modelo	**Original**	**Decomposto**	**Original**	**Decomposto**
AD	0,7477 ⊣ 0,7569	0,7477 ⊣ 0,7569	0,5712 ⊣ 0,6535	0,5694 ⊣ 0,6535
Logit	0,7531 ⊣ 0,7554	0,7531 ⊣ 0,7554	0,4612 ⊣ 0,5659	**0,5506** ⊣ **0,6886**
Logit-S	0,7531 ⊣ 0,7554	0,7531 ⊣ 0,7554	0,4612 ⊣ 0,5659	**0,5506** ⊣ **0,6892**
k-NN	0,6585 ⊣ 0,7538	0,6338 ⊣ 0,7538	0,4737 ⊣ 0,6424	0,4879 ⊣ 0,6061
MLP	0,7531 ⊣ 0,7554	0,7531 ⊣ 0,7554	0,4609 ⊣ 0,5652	**0,5641** ⊣ **0,7174**

Tabela 5: Variável *Citric Acid* no conjunto de dados *Wine Quality*

	Precisão		**AUC**	
Modelo	**Original**	**Decomposto**	**Original**	**Decomposto**
AD	0,7815 ⊣ 0,8338	0,7815 ⊣ 0,8338	0,7306 ⊣ 0,8052	0,7306 ⊣ 0,8052
Logit	0,7531 ⊣ 0,7554	**0,7785** ⊣ **0,8400**	0,5057 ⊣ 0,6968	**0,7088** ⊣ **0,7976**
Logit-S	0,7531 ⊣ 0,7554	**0,7785** ⊣ **0,8369**	0,5057 ⊣ 0,6968	**0,7126** ⊣ **0,8071**
k-NN	0,7323 ⊣ 0,8092	0,6738 ⊣ 0,8215	0,6033 ⊣ 0,7502	**0,6453** ⊣ **0,7614**
MLP	0,7785 ⊣ 0,8338	0,7785 ⊣ 0,8400	0,7188 ⊣ 0,8105	0,7162 ⊣ 0,8051

Tabela 6: Variável *Density* no conjunto de dados *Wine Quality*

	Precisão		**AUC**	
Modelo	**Original**	**Decomposto**	**Original**	**Decomposto**
AD	0,7446 ⊣ 0,7938	0,7385 ⊣ 0,7969	0,7596 ⊣ 0,8337	0,7528 ⊣ 0,8360
Logit	0,6923 ⊣ 0,7600	**0,7292** ⊣ **0,7846**	0,7450 ⊣ 0,8067	**0,7458** ⊣ **0,8237**
Logit-S	0,7531 ⊣ 0,7554	0,7231 ⊣ 0,7754	0,1933 ⊣ 0,2550	**0,7323** ⊣ **0,8154**
k-NN	0,7262 ⊣ 0,8025	0,7292 ⊣ 0,7877	0,7379 ⊣ 0,8169	0,7208 ⊣ 0,8019
MLP	0,7531 ⊣ 0,7554	0,7399 ⊣ 0,7662	0,1933 ⊣ 0,2550	**0,7412** ⊣ **0,8141**

Tabela 7: Variável *Residual Sugar* no conjunto de dados *Wine Quality*

	Precisão		**AUC**	
Modelo	**Original**	**Decomposto**	**Original**	**Decomposto**
AD	0,8092 ⊣ 0,8615	0,8092 ⊣ 0,8615	0,8549 ⊣ 0,9139	0,8549 ⊣ 0,9139
Logit	0,7531 ⊣ 0,7554	**0,7969** ⊣ **0,8615**	0,6225 ⊣ 0,7188	**0,8486** ⊣ **0,9132**
Logit-S	0,7531 ⊣ 0,7554	**0,7969** ⊣ **0,8483**	0,6225 ⊣ 0,7188	**0,8490** ⊣ **0,9124**
k-NN	0,7815 ⊣ 0,8523	0,7815 ⊣ 0,8359	0,8077 ⊣ 0,8871	0,8040 ⊣ 0,8779
MLP	0,8123 ⊣ 0,8677	0,8000 ⊣ 0,8731	0,8611 ⊣ 0,9187	0,8562 ⊣ 0,9180

Tabela 8: Variável *Volatile Acidity* no conjunto de dados *Wine Quality*

	Precisão		AUC	
Modelo	**Original**	**Decomposto**	**Original**	**Decomposto**
AD	0,8369 ⊢ 0,9046	0,8369 ⊢ 0,9046	0,8708 ⊢ 0,9329	0,8715 ⊢ 0,9329
Logit	0,8400 ⊢ 0,9046	0,8308 ⊢ 0,9015	0,8739 ⊢ 0,9317	0,8650 ⊢ 0,9310
Logit-S	0,7531 ⊢ 0,7554	**0,8308 ⊢ 0,9015**	0,0683 ⊢ 0,1261	**0,8733 ⊢ 0,9324**
k-NN	0,7938 ⊢ 0,8954	0,8000 ⊢ 0,8923	0,8178 ⊢ 0,8929	0,7935 ⊢ 0,9161
MLP	0,8431 ⊢ 0,9046	0,8308 ⊢ 0,9046	0,8743 ⊢ 0,9317	0,8740 ⊢ 0,9317

Das Tabelas de 1 a 8, na coluna relativa à precisão da Tabela 3 a metodologia proposta melhorou a solução de 3 dos 5 métodos testados e obteve a mesma solução para um deles. Somente na Tabela 4 a metodologia proposta apenas igualou os resultados obtidos pelos métodos originais da literatura. Em todas as outras Tabelas, para pelo menos um dos métodos a metodologia proposta melhorou os resultados obtidos para a precisão. Já em relação à coluna relativa à área sob a curva ROC, em todas as Tabelas há pelo menos dois métodos cuja resposta foi melhor ao utilizar a metodologia proposta. Vale ressaltar que na Tabela 3 a área sob a curva ROC foi melhor com a metodologia proposta em 4 dos 5 métodos de classificação e igualou o resultado para o outro método. É importante ressaltar também que na Tabela 8, para o modelo AD a coluna AUC do método proposto melhorou o limite inferior e igualou o limite superior em relação ao método original.

Para as Tabelas 6 e 8, as duas referentes ao banco de dados Wine [Dua e Graff, 2017b], a metodologia proposta melhorou muito o AUC em relação ao método Logit-S como pode ser visto respectivamente nas figuras 2. Cada uma das figuras apresenta a curva ROC média para o modelo original, exibida com linha pontilhada, e para o modelo utilizando a metodologia proposta, exibida com a linha sólida.

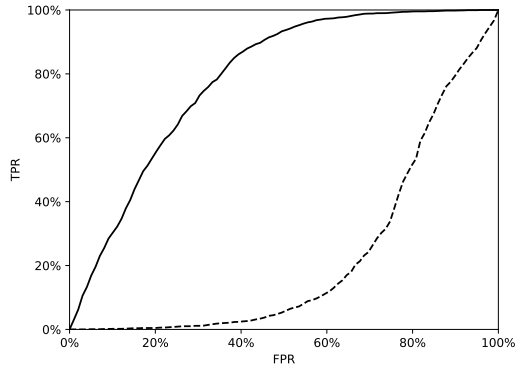


figura (a)

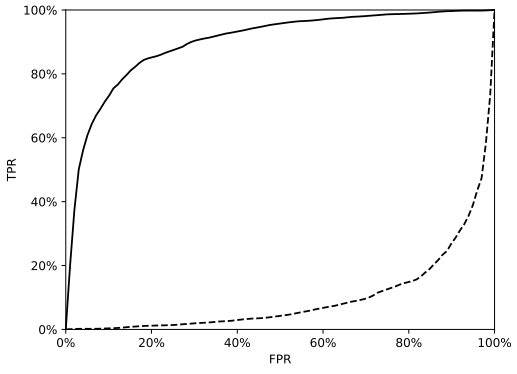


figura (b)

Figura 2: Curva ROC média para a variável density (a) e volatile acidity(b)

4. Conclusões e Trabalhos Futuros

Com base nos resultados obtidos, observamos que a metodologia de decomposição proposta neste trabalho pode proporcionar melhora na precisão e/ou na eficácia dos métodos de classificação já conhecidos na literatura.

Os possíveis caminhos para continuidade deste trabalho são:

- Testar a metodologia de decomposição proposta com mais métodos de classificação;
- Utilizar mais bancos de dados para os testes;
- Testar para o caso multivariado;
- Registrar o tempo computacional de cada um dos experimentos para comprovar que a metodologia proposta se apresenta como uma opção válida;
- Formular matematicamente o modelo.

Referências

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145 – 1159. ISSN 0031-3203. URL http://www.sciencedirect.com/science/article/pii/S0031320396001422.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., e Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, p. 108–122.

Cover, T. e Hart, P. (2006). Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1): 21–27. ISSN 0018-9448. URL https://doi.org/10.1109/TIT.1967.1053964.

Cox, D. R. (1958). The regression analysis of binary sequences (with discussion). *J Roy Stat Soc B*, 20:215–242.

Dua, D. e Graff, C. (2017a). Skin segmentation - UCI machine learning repository. URL http://archive.ics.uci.edu/ml.

Dua, D. e Graff, C. (2017b). Wine quality - UCI machine learning repository. URL http://archive.ics.uci.edu/ml.

Gray, C. C. e Perkins, D. (2019). Utilizing early engagement and machine learning to predict student outcomes. *Computers & Education*, 131:22 – 32. ISSN 0360-1315. URL http://www.sciencedirect.com/science/article/pii/S0360131518303191.

H. Walker, S. e B. Duncan, D. (1967). Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54:167–79.

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python. URL http://www.scipy.org/. [Online; accessed <today>].

McCoy, J. e Auret, L. (2019). Machine learning applications in minerals processing: A review. *Minerals Engineering*, 132:95 – 109. ISSN 0892-6875. URL http://www.sciencedirect.com/science/article/pii/S0892687518305430.

Nelder, J. A. e Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384.

Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1(1):81–106. ISSN 0885-6125. URL http://dx.doi.org/10.1023/A:1022643204877.

Seabold, S. e Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Tanizaki, T., Hoshino, T., Shimmura, T., e Takenaka, T. (2019). Demand forecasting in restaurants using machine learning and statistical analysis. *Procedia CIRP*, 79:679 – 683. ISSN 2212-8271. URL http://www.sciencedirect.com/science/article/pii/S2212827119301568. 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy.

Ucci, D., Aniello, L., e Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. *Computers & Security*, 81:123 – 147. ISSN 0167-4048. URL http://www.sciencedirect.com/science/article/pii/S0167404818303808.