Uma Análise Comparativa de Classificadores da Literatura Para a Base de Dados Wine da Biblioteca SciKit Learn

Enzo B. Cussuol[1]

Vitória, Espírito Santo

Abstract

Esse artigo apresenta uma análise comparativa entre classificadores de aprendizado de máquina clássicos da literatura em cima da base de dados Wine da biblioteca SciKit Learn. Para cada método, foram aplicadas técnicas de treinamento e teste a fim de obter resultados confiáveis, tais como uso de validações cruzadas, variações de hiperparâmetros e padronização dos dados. Os resultados foram orquestrados em tabelas e gráficos, os quais deixaram evidente o desempenho de cada um dos classificadores empregados.

1. Introdução

Com o avanço de pesquisas na área de Inteligência Artificial nos últimos anos, em especial no que se refere ao desenvolvimento de algoritmos de aprendizado de máquina, isto é, algoritmos que se baseiam em treinamentos e testes, surgiu uma série de ferramentas que possibilitaram aos programadores realizarem predições em bases de dados.

Contudo, não é trivial definir qual ferramenta utilizar em cada cenário, uma vez que os algoritmos podem se comportar de forma melhor ou pior de acordo com a situação à qual são submetidos. Urge, portanto, a necessidade de uma análise comparativa entre as técnicas de interesse, a fim de obter o melhor resultado possível.

Neste trabalho, foram analisados os algoritmos ZeroR (ZR), Naive Bayes Gaussiano (NBG) [1], KMeans Centroides (KMC), K Vizinhos Mais Próximos (KNN) [2] e Árvore de Decisão (AD) [3] para realizar predições em cima da base de dados Wine da biblioteca SciKit Learn.

O restante deste artigo está organizado como se segue: A Seção 2 discute a base de dados escolhida. A Seção 3 aborda o desenvolvimento do KMC. A Seção 4 descreve os experimentos realizados e resultados obtidos. As conclusões finais são apresentadas na Seção 5.

[1]enzo.cussuol@edu.ufes.br

2. Base de Dados

Como já informado, a base de dados escolhida para análise foi a Wine do SciKit Learn [4]. Essa base descreve vinhos cultivados em uma região específica da Itália, a qual não é informada, por 3 agricultores diferentes.

2.1. Descrição do Domínio

A base de dados conta com 178 instâncias, as quais possuem 13 características e uma dentre 3 classes associada. Essas características são números reais contínuos e a classe é um número inteiro.

2.2. Definição das Classes e das Características

Os valores das classes variam entre 0, 1 e 2, o que representa qual agricultor cultivou o vinho. Já as características estão associadas à propriedades químicas envolvidas no cultivo do vinho, são elas: álcool, ácido málico, cinza, alcalinidade das cinzas, magnésio, total de fenóis, flavonóides, fenóis não flavonóides, proantocianinas, intensidade da cor, matiz, OD280/OD315 e prolina.

2.3. Número de Instâncias

Como já dito, a base de dados possui 178 instâncias, as quais estão distribuídas da seguinte forma: 59 instâncias da classe 0, 71 da classe 1 e 48 da classe 2.

3. O Método KMC

Os métodos utilizados na análise já estão prontos dentro da própria biblioteca SciKit Learn, contudo, para o KMC, foi realizada uma implementação própria. O SciKit Learn fornece ferramentas para que o próprio programador implemente o seu classificador, basta que ele crie uma classe que herde da classe BaseEstimator, o que foi exatamente o que foi realizado, criando-se a classe KMC.

O classificador KMC consiste em utilizar um algoritmo de agrupamento, que nesse caso foi o KNN, para gerar K grupos dentro de cada classe da base de dados. Feito isso, um centróide será gerado para cada grupo, o qual estará associado à classe do grupo. Por fim, para realizar a predição, calcula-se a menor distância da instância para um dos centróides, dando à instância a classe associada ao centróide mais próximo dela.

Dito isso, para implementar a classe KMC basta adicionar os atributos desejados e implementar os métodos fit e predict. Com relação aos atributos, basta guardar o valor de k e uma lista de centróides, a qual é preenchida pelo método fit. Já o método predict, como o nome sugere, realiza a predição, executando o procedimento de cálculo das distâncias descrito no parágrafo anterior.

4. Descrição dos Experimentos Realizados e seus Resultados

A fim de obter resultados confiáveis, os experimentos, isto é, os treinos e testes realizados para cada classificador, foram realizados utilizando técnicas de validação cruzada estratificada. Para os métodos que possuíam hiperparâmetros (KMC, KNN e AD), foi ainda adicionada um ciclo interno de validação que variava esses hiperparâmetros.

Os valores escolhidos foram 3 rodadas de validação cruzada com 10 camadas externas e, quando possível, 4 camadas internas. Os valores dos hiperparâmetros foram: KMC: [k = 1, 3, 5, 7], KNN: [n_neighbors = 1, 3, 5, 7] e AD: [max_depth = None, 3, 5, 10].

Além disso, foi realizado um pré-processamento na base de dados, no qual utilizou-se uma padronização z-score. Essa padronização consiste em normalizar os dados de forma que a média de todos os valores é 0 e o desvio padrão é 1. Essa padronização se faz necessária pois ela reduz drasticamente o efeito de *outliers no resultado dos testes. O procedimento segue a fórmula abaixo, na qual* x' *é o novo valor da característica,* x *é o valor original,* α *é a média dos valores da característica e* β *é o desvio padrão desses valores:*

$$x' = \frac{x - \alpha}{\beta}$$

Com os experimentos realizados, podemos analisar algumas métricas para fundamentar a discussão a respeito de qual classificador foi mais bem sucedido. A Tabela 4 apresenta os resultados obtidos em função de métricas estatísticas básicas, e a Figura 1 ilustra a distribuição desses resultados para cada classificador.

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
ZR	0.399346	0.024707	0.390505	0.408188
NBG	0.973420	0.048205	0.956171	0.990670
KMC	0.966122	0.042954	0.950751	0.981493
KNN	0.960458	0.053188	0.941425	0.979490
AD	0.889760	0.075837	0.862623	0.916898

Table 1: Métricas estatísticas referentes aos resultados obtidos por cada método

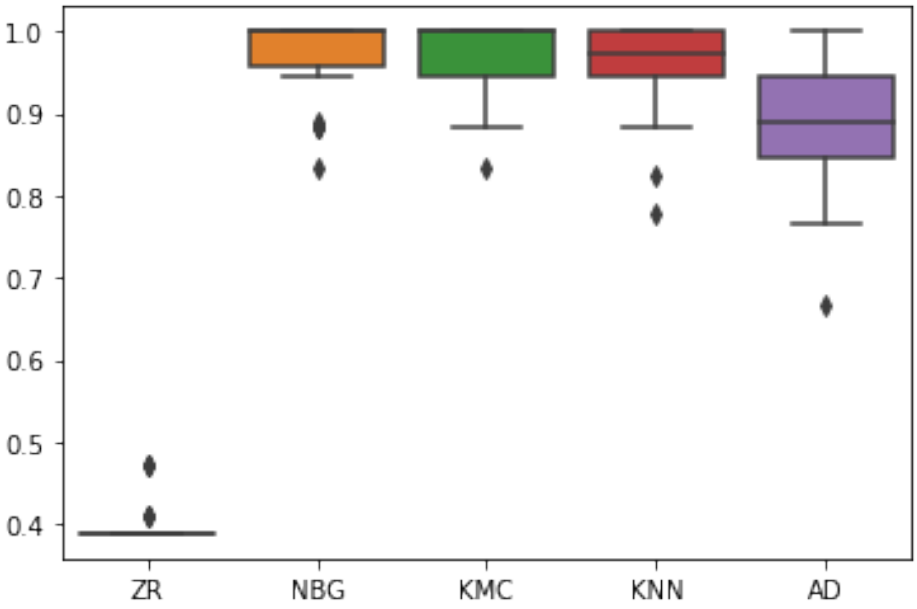

Figure 1: Boxplot referente aos resultados obtidos por cada método

Além disso, foi elaborada a Tabela 4, a qual mostra os resultados dos testes de hipótese entre os pares de métodos. Na matriz triangular superior estão os resultados do teste t pareado e na matriz triangular inferior estão os resultados do teste não paramétrico de Wilcoxon [5]. Os valores em negrito rejeitaram a hipótese nula para um nível de significância de 95%, isto é, para aquele par de métodos, pode-se dizer que eles apresentaram resultados significativamente diferentes.

ZR	**0**	**0**	**0**	**0**
0	NBG	0.17667	0.05271	**0**
0	0.30493	KMC	0.956171	**0**
0	0.11458	0.36571	KNN	**0**
0	**0.00001**	**0.00003**	**0.00001**	AD

Table 2: Tabela pareada dos resultados dos testes de hipótese entre os pares de métodos

5. Conclusões

5.1. Análise geral dos resultados

Claramente o classificador ZR foi o pior, enquanto que os classificadores NBG, KMC e KNN aparecem praticamente empatados como os que obtiveram melhores resultados. O classificador AD teve desempenho relativamente bom, mas não tanto quanto esses 3 últimos.

Além disso, a partir da Tabela 4, podemos dizer que os classificadores NBG, KMC e KNN não obtiveram diferenças significativas em seus resultados, o que não é válido para os pares que incluem os classificadores ZR e AD.

Portanto, a partir da análise fornecida, é possível inferir que, dentre os métodos analisados, os classificadores NBG, KMC e KNN são os melhores para predizer resultados referentes à base de dados Wine.

5.2. Contribuições do Trabalho

Este trabalho forneceu uma análise comparativa entre diferentes classificadores utilizados na área de aprendizado de máquina. O contexto estudado foi a classificação de um certo vinho à um certo agricultor. Foram discutidos quais os melhores e piores métodos para se obter melhores resultados.

5.3. Melhorias e trabalhos futuros

Como trabalhos futuros, convém a análise de um maior número de classificadores e eventualmente a obtenção de mais dados referentes aos vinhos na Itália.

References

[1] K. M. Leung, Naive bayesian classifier, Polytechnic University Department of Computer Science/Finance and Risk Engineering 2007 (2007) 123–156.

[2] L. E. Peterson, K-nearest neighbor, Scholarpedia 4 (2) (2009) 1883.

[3] S. R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE transactions on systems, man, and cybernetics 21 (3) (1991) 660–674.

[4] O. Kramer, Scikit-learn, in: Machine learning for evolution strategies, Springer, 2016, pp. 45–53.

[5] R. F. Woolson, Wilcoxon signed-rank test, Wiley encyclopedia of clinical trials (2007) 1–3.