31/03/2021

Segundo Trabalho de Inteligência Artificial e Sistemas Inteligentes

Prof. Flávio Miguel Varejão

1. *Descrição*

Este trabalho consiste em realizar uma comparação experimental entre um conjunto pré-definido de técnicas de aprendizado e classificação automática aplicadas a alguns problemas de classificação. As técnicas escolhidas são: ZeroR, Aleatório, Aleatório Estratificado, OneR Probabilístico, Naive Bayes Gaussiano, KmeansCentroides, KGACentroides, Knn, DistKnn, Árvore de Decisão e Florestas de Árvores. As bases de dados a serem utilizadas são iris, digits, wine e breast cancer, todas disponíveis em https://scikit-learn.org/stable/datasets/toy_dataset.html.

Para cada base, o procedimento experimental será dividido em duas etapas.

A primeira etapa consiste no treino e teste com 3 rodadas de validação cruzada estratificada de 10 folds dos classificadores que não possuem hiperparâmetros, isto é, os classificadores ZeroR, Aleatório, Aleatório Estratificado, OneR Probabilístico e Naive Bayes Gaussiano. Os resultados de cada classificador devem ser apresentados numa tabela contendo a média das acurácias obtidas em cada fold, o desvio padrão e o intervalo de confiança a 95% de significância dos resultados, e também através do boxplot dos resultados de cada classificador em cada fold.

A segunda etapa consiste no treino, validação e teste dos classificadores que precisam de ajuste de hiperparâmetros, isto é, os classificadores KmeansCentroides, KGACentroides, Knn, DistKnn, Árvore de Decisão e Florestas de Árvores. Neste caso o procedimento de treinamento, validação e teste será realizado através de 3 rodadas de ciclos aninhados de validação e teste, com o ciclo interno de validação contendo 4 folds e o externo de teste com 10 folds. A busca em grade (grid search) do ciclo interno deve considerar os seguintes valores de hiperparâmetros de cada técnica de aprendizado:

KMeansCentroides: [k = 1, 3, 5, 7]

KGACentroides: [k = 1, 3, 5, 7]

Knn: [n_neighbors = 1, 3, 5, 7]

DistKnn: [n_neighbors = 1, 3, 5, 7]

Árvore de Decisão: [max_depth = None, 3, 5, 10]

Florestas de Árvores: [n_estimators = 10, 20, 50, 100]

Os resultados de cada classificador devem ser apresentados numa tabela contendo a média das acurácias obtidas em cada fold do ciclo externo, o desvio padrão e o intervalo de confiança a 95% de significância dos resultados, e também através do boxplot dos resultados de cada classificador em cada fold.

Um exemplo de uma tabela para uma base de dados hipotética é mostrado a seguir.

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
ZeroR	0.95	0.04	0.94	0.96
Aleatório	0.92	0.01	0.91	0.93
Estratificado	0.96	0.08	0.90	0.99
OneR Prob	0.93	0.01	0.93	0.94
Naive Bayes	0.91	0.04	0.87	0.95
KMeans	0.95	0.02	0.94	0.96
KGA	0.95	0.03	0.92	0.98
KNN	0.96	0.07	0.88	0.99
DistKNN	0.98	0.01	0.96	0.99
Árv Desc	0.97	0.02	0.95	0.98
Floresta	0.96	0.04	0.92	0.99

Os métodos OneR Probabilístico, KmeansCentroides e KGACentroides devem ser implementados. Os métodos KmeansCentroides e KGACentroides devem ser implementados como variações do método KCentroides. Os métodos ZeroR, Aleatório, Aleatório Estratificado, Naive Bayes Gaussiano, Knn, DistKnn, Árvores de Decisão e Florestas de Árvores estão disponíveis no scikit-learn. Os métodos ZeroR, Aleatório, Aleatório Estratificado são variações do DummyClassifier. Os métodos Knn e DistKnn são variações do KneighborsClassifier. As descrições dos métodos implementados no sklearn podem ser acessadas respectivamente em:
https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier

Os dados utilizados no conjunto de treino em cada rodada de teste devem ser padronizados (normalização com z-score). Os valores de padronização obtidos nos dados de treino devem ser utilizados para padronizar os dados do respectivo conjunto de teste.

Além das tabelas e dos gráficos bloxplot, será necessário apresentar também a tabela pareada dos resultados (p-values) dos testes de hipótese entre os pares de métodos. Na matriz triangular superior devem ser apresentados os resultados do teste t pareado (amostras dependentes) e na matriz triangular inferior devem ser apresentados os resultado do teste não paramétrico de wilcoxon. Os valores da célula da tabela rejeitarem a hipótese nula para um nível de significância de 95% devem ser escritos em negrito.

Um exemplo de uma tabela pareada para uma base de dados hipotética é mostrado a seguir.

ZeroR	0.0941	0.095	0.096	0.085	**0.045**	0.096	0.065	0.096	0.089	0.096
0.096	Aleat	0.074	0.084	0.096	0.064	0.084	0.075	0.084	0.096	0.084
0.084	0.096	Estrat	0.105	0.084	0.096	0.105	0.096	0.105	0.084	0.105
0.105	0.084	0.096	OneR	0.105	0.084	0.095	0.084	0.074	0.105	**0.032**
0.045	0.105	0.084	0.096	NB	0.105	0.085	0.105	0.097	0.076	0.096
0.096	**0.045**	0.105	0.084	0.096	KM	0.075	0.085	0.084	0.096	0.084
0.084	0.096	0.145	0.105	0.084	0.065	KGA	0.094	0.105	0.084	0.105
0.105	0.084	0.096	**0.034**	0.105	0.096	0.075	KNN	0.083	0.105	**0.023**
0.043	0.105	0.084	0.096	0.074	0.084	0.096	0.096	DKNN	0.011	**0.034**
0.054	**0.049**	0.105	0.084	0.094	0.105	0.084	0.084	0.074	AD	0.075
0.024	0.074	0.053	0.105	0.059	**0.015**	0.105	0.105	**0.044**	**0.042**	Flor

2. *OneR Probabilístico*

O classificador OneR Probabilístico escolhe na etapa de treino a característica que tenha maior poder de predição individual dentre todas. Para fazer essa escolha é criada uma tabela de contigência para cada característica da base de dados contando-se a quantidade de ocorrências de cada par (valor da característica, valor da classe) na base de dados de treino. valores desse atributo e as colunas são as classificações possíveis.

Considere abaixo as tabelas de contingência das características Cor e Manga em um exemplo hipotético de decisão da Classe de um tipo de camisa:

Característica (Cor)	Classe A	Classe B	Classe C
Verde	**12**	1	2
Azul	7	7	**8**
Amarelo	0	1	**10**

Característica (Manga)	Classe A	Classe B	Classe C
Curta	**12**	9	7
Longa	7	0	**13**

A base de treino possui 19 exemplos da Classe A, 9 da Classe B e 20 da Classe C, perfazendo um total de 48 exemplos. Pode-se observar na tabela de Cor 15 camisas são da cor Verde, sendo que 12 destes são da Classe A, 1 da Classe B e 2 da classe C. De modo análogo, pode-se observar na tabela de Manga que 20 camisas são de manga Longa, sendo que 7 são da Classe A e 13 da Classe C.

Após montar as tabelas de contigência da cada característica, o método OneR Probabilístico selecionará aquela que tem maior poder de diferenciação das classes. Para isso, é necessário somar os valores das células que contém o maior número de ocorrências de cada valor da característica. Esses valores estão escritos em negrito nas tabelas acima. Assim, para a características Cor soma-se 12 + 8 + 10, perfazendo um total de 30. E para a característica Manga soma-se 12 + 13, perfazendo um total de 25. A característica com maior poder de diferenciação é aquela que possui a maior soma. Caso a base somente tivesse as características Cor e Manga, a característica selecionada seria Cor.

Uma vez selecionada a característica de maior poder de diferenciação é necessário obter a distribuição de classes de cada valor da característica. No caso da cor Verde, 12/15 é a fração de exemplos da Classe A, 1/15 é a fração de exemplos da Classe B e 2/15 é a fração de exemplos da Classe C.

Ao realizar a classificação, o OneR Probabilístico verifica qual o valor da Cor da camisa que precisa ter sua classe predita. Se a cor é Verde, ele faz um sorteio ponderado levando em conta a distribuição de classes deste valor, portanto, o sorteio daria uma chance de 12/15 para escolher a Classe A, 1/15 para a Classe B e 2/15 para a Classe C.

Um requisito para se construir as tabelas de contigências das características é que elas possuam um conjunto moderado de valores discretos, o que não é o caso das características existentes nas bases de dados utilizadas neste trabalho. Portanto, será necessário realizar uma discretização nas características das bases de dados. Os valores discretizados obtidos nos dados de treino devem ser utilizados para discretizar os dados do respectivo conjunto de teste. O método de discretização a ser utilizado é o de intervalos definidos pelo método Kmeans. Esse método de discretização está implementado no sklearn e sua descrição pode ser acessada em
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.KBinsDiscretizer.

O parâmetro strategy deve ser definido como *kmeans*.

Para simplificação, em vez de se definir um hiperparâmetro para este método, será adotado que o número de intervalos de cada característica será o dobro do número de classes da base.

3. *KCentroides*

O classificador KCentroides utiliza um algoritmo de agrupamento para definir K grupos de exemplos de cada classe na base de treino. Assumindo que uma base de dados possui ncl classes, o algoritmo KCentroides forma inicialmente K*ncl grupos, sendo K grupos em cada uma das ncl classes. Em seguida, são calculados os centróides de cada um dos grupos e este centróide é associado a classe do grupo a partir do qual foi gerado. O método possui como hiperparâmetro o valor de K.

Para realizar uma classificação, o KCentroides verifica qual o centróide mais próximo do elemento a ser classificado e retorna a sua classe.

Para se criar o método KmeansCentroides, o método Kmeans deve ser passado para o Kcentroides em sua criação. De forma análoga, para se criar o método KGACentroides, o

método de agrupamento GA (Genetic Algorithm) deve ser passado para o Kcentroides em sua criação.

4. *Informações Complementares*

a. Use o valor 36851234 para o parâmetro random_state (`random_state=36851234`) nas chamadas a RepeatedStratifiedKFold para que os resultados sejam reproduzíveis.

b. Os classificadores ZeroR, Aleatório e Aleatório Estratificado correspondem ao uso do DummyClassifier estabelecendo respectivamente o valor do parâmetro strategy como *most_frequent, uniform* e *stratified.*

c. O classificador DistKnn corresponde ao uso do KneighborsClassifier estabelecendo o valor do parâmetro weights como *distance.*

d. Os gráficos bloxplot requeridos no treino e no teste devem ser gerados usando função específica do pacote seaborn (ver instruções de instalação e uso no apêndice A deste enunciado).

e. O apêndice B deste enunciado apresenta instruções de instalação e uso do overleaf para a escrita do artigo.

5. *Artigo*

Após a realização dos experimentos, um artigo descrevendo todo o processo experimental realizado deverá ser escrito em latex usando o software overleaf. O artigo deve ser estruturado contendo os seguintes componentes:

1. Título
2. Resumo
3. Seção 1. Introdução
4. Seção 2. Descrição dos Métodos Implementados (OneR Probabilístico e KCentróides)
5. Seção 3. Descrição dos Experimentos Realizados
    a. Seção 3.1 Iris
    b. Seção 3.2 Digits
    c. Seção 3.3 Wine
    d. Seção 3.4 Breast Cancer
6. Seção 4. Conclusões
    a. Análise geral dos resultados
    b. Contribuições do Trabalho
    c. Melhorias e trabalhos futuros
7. Referências Bibliográficas

Na subseção de análise geral dos resultados é importante discutir, dentre outras coisas, se houve diferença estatística significativa entre quais métodos em quais bases e responder se teve um método que foi superior em mais de uma base.

6. Condições de Entrega

O trabalho deve ser feito individualmente e submetido pelo sistema da sala virtual até a data limite (28 de abril de 2021).

O trabalho deve ser submetido em três arquivos: um arquivo pdf com o artigo produzido no trabalho, um arquivo ipynb com o notebook jupyter para ser carregado e executado no jupyter e todos os arquivos com código fonte em python utilizados em um arquivo zip. Tanto o arquivo pdf quanto os arquivos ipynb e zip devem possuir o mesmo nome Trab2_Nome_Sobrenome. Note que a data limite já leva em conta um dia adicional de tolerância para o caso de problemas de submissão via rede. Isso significa que o aluno deve submeter seu trabalho até no máximo um dia antes da data limite. Se o aluno resolver submeter o trabalho na data limite, estará fazendo isso assumindo o risco do trabalho ser cadastrado no sistema após o prazo. Em caso de recebimento do trabalho após a data limite, o trabalho não será avaliado e a nota será ZERO. Aluno que receber zero por este motivo e vier pedir para o professor considerar o trabalho não será considerado. Plágio ou cópia de trabalhos serão verificadas automaticamente por sistemas como o moss. Trabalhos em que se configure cópia receberão nota zero independente de quem fez ou quem copiou.

7. Requisitos da implementação

a. Modularize seu código adequadamente.

b. Crie códigos claros e organizados. Utilize um estilo de programação consistente, Comente seu código.

c. Os arquivos do programa devem ser lidos e gerados na mesma pasta onde se encontram os arquivos fonte do seu programa.

Observação importante

Caso haja algum erro neste documento, serão publicadas novas versões e divulgadas erratas em sala de aula. É responsabilidade do aluno manter-se informado, freqüentando as aulas ou acompanhando as novidades na página da disciplina na Internet.

Apêndice A. Boxplots usando seaborn

```
def example1():
      mydata=[1,2,3,4,5,6,12]
      sns.boxplot(y=mydata) # Also accepts numpy arrays
      plt.show()

def example2():
      df = sns.load_dataset('iris')
      #returns a DataFrame object. This dataset has 150 examples.
      #print(df)
      # Make boxplot for each group
```

```
        sns.boxplot( data=df.loc[:,:] )
        # loc[:,:] means all lines and all columns
        plt.show()

example1()
example2()
```

Apêndice B. Artigo em Latex usando Overleaf

Juntamente com este enunciado foi disponibilizado um arquivo zip com o template de latex para confecção do artigo. O primeiro passo a ser feito é criar uma conta pessoal no Overleaf (https://www.overleaf.com/register). Uma vez criada sua conta, deve-se entrar nela. Para incluir o template no overleaf, basta apenas selecionar "New Project>Upload Project" e selecionar o arquivo zip, como mostrado na figura abaixo. Não é necessário descompactar, faça o upload do zip direto. Lembrar de renomear o artigo após o upload do arquivo.