05/05/2022

Primeiro 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 um problema de classificação. As técnicas escolhidas são: ZeroR (ZR), Bagging (BA), AdaBoost (AB), RandomForest (RF) e Heterogeneous Pooling (HP). 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 do classificador que não possui hiperparâmetros, isto é, o classificadores ZR.

A segunda etapa consiste no treino, validação e teste dos classificadores que precisam de ajuste de hiperparâmetros, isto é, os classificadores BA, AB, RF e HP. 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:

Bagging: [n_estimators = 3, 9, 15, 21]

AdaBoost: [n_estimators = 3, 9, 15, 21]

RandomForest: [n_estimators = 3, 9, 15, 21]

HeterogeneousPooling: [n_samples = 1, 3, 5, 7]

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.

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

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
ZR	0.95	0.04	0.94	0.96
BA	0.91	0.04	0.87	0.95
AB	0.95	0.02	0.94	0.96
RF	0.96	0.07	0.88	0.99
HP	0.97	0.02	0.95	0.98

O método HP deve ser implementado. Os métodos ZR, BA, AB e RF estão disponíveis

no scikit-learn. 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.ensemble.BaggingClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

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	0.085	**0.045**	0.065	0.089
0.045	BA	0.105	0.105	0.076
0.096	**0.036**	AB	0.085	0.096
0.105	0.105	0.096	RF	0.105
0.024	0.094	0.105	0.084	HP

2. HP

O classificador Heterogeneous Pooling é um combinado de classificadores heterogêneos que usa como classificadores base: Árvore de Decisão (DT), Naive Bayes Gaussiano (NB) e K Vizinhos Mais Proximo (KNN), sempre com valores default do sklearn para seus hiperparâmetros. O único parâmetro do método Heterogeneous Pooling é o *n_samples*, que indica o número de vezes que os classificadores base serão usados para gerar o combinado. Por exemplo, se *n_samples* é igual a 3, o combinado será composto por 9 classificadores: 3 árvores de decisão, 3 naive bayes e 3 vizinhos mais próximos. Para diferenciar os classificadores de mesmo tipo em um combinado, o primeiro deles será treinado com a base de treino original e os demais serão treinados com uma base de treino diferente, obtida a partir da base de treino original através da função resample do sklearn. Ela pode ser acessada em:
https://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html

As descrições dos métodos KNN, NB e DT usados no HP e implementados no sklearn podem ser acessadas respectivamente em:
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

O critério de decisão para classificar uma instância é votação majoritária, ou seja, deve-se escolher a classe mais escolhida dentre os classificadores que compõem o combinado. Em caso de empate, a classe escolhida deve ser a mais frequente na base de dados de treino original dentre as que empataram na votação.

O pseudo código a seguir mostra como o HP é obtido a partir de uma base de dados de treino:

- Obter e armazenar a ordenação das classes de acordo com a ocorrência nos exemplos na base de treino (ordenar decrescentemente da mais frequente para a menos frequente)
- Para cada um dos *n_samples* faça
    - Se for a primeira iteração então
        - Usar a base original para treino dos classificadores
    - Senão
        - Montar uma base de treino de mesmo tamanho da original coletando aleatoriamente exemplos da base original com reposição
    - Fim-se
    - Treinar os classificadores NN, NB, DT na base de treino corrente e incluí-los no combinado de classificadores
- Fim-para

O pseudo código seguinte mostra como o combinado HP é usado para classificar um exemplo da base de dados de teste:

- Para cada um dos classificadores individuais do combinado faça
    - Obter a classificação do exemplo usando o classificador individual e armazenar a classe selecionada
- Fim-para
- Contar quantas vezes cada classe foi selecionada e obter a(s) mais votada(s)
- Se mais de uma classe for a mais votada então
    - Retornar a classe mais votada mais frequente na base de treino dentre as que empataram
- Senão
    - Retornar a classe mais votada
- Fim-se

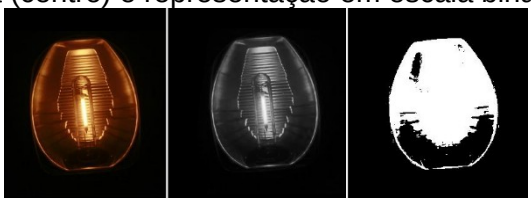
---

3. Base de Dados

A base de dados usada no trabalho foi obtida de um projeto de pesquisa que visa utilizar informações de imagens de lâmpadas e luminárias de iluminação pública para atualização de cadastro das concessionárias de energia. A base completa contém 297 exemplos de imagens de lâmpadas e luminárias.

A Figura 1 apresenta um exemplo de uma foto originalmente colorida de um dispositivo de iluminação pública, sua representação em escala de cinza e em preto e branco.

Figura 1 – Foto original colorida de uma lâmpada de iluminação pública (esquerda), imagem em escala de cinza (centro) e representação em escala binária (direita).



A partir da imagem colorida do dispositivo e suas representações, é realizado o processamento das imagens e são extraídas características para compor a base de dados. Três tipos de técnicas foram aplicadas nessa extração. Descritores de Fourier utilizam a imagem binária e são usados para descrever o contorno do objeto. Descritores de Hu usam a imagem em escala de cinza e definem um conjunto de momentos invariantes para descrever a imagem. Os descritores de Haralick usam uma abordagem estatística para descrever as texturas da imagem com base na distribuição e relacionamento da escala de cinza da imagem. A base completa obtida contém 10 descritores de Fourier, 7 descritores de Hu e 6 descritores de Haralick, totalizando 23 características.

Os trabalhos utilizarão diferentes subconjuntos de características: alunas(os) cuja matrícula terminam em 0 ou 1 utilizarão a base completa (com os 10 descritores de Fourier, 7 descritores de Hu e 6 descritores de Haralick), aquelas(es) que terminam em 2 usarão apenas os 10 descritores de Fourier, as(os) que terminam em 3 usarão apenas os 7 descritores de Hu, as(os) que terminam em 4 usarão apenas os 6 descritores de Haralick, as(os) que terminam em 5 ou 6 usarão os 10 descritores de Fourier e os 7 descritores de Hu, as(os) que terminam em 7 ou 8 usarão os 10 descritores de Fourier e os 6 descritores de Haralick, as(os) que terminam em 9 usarão os 7 descritores de Hu e os 6 descritores de Haralick. Juntamente com o enunciado do trabalho é fornecido um notebook jupyter com código que pode ser usado para fazer a leitura dos dados conforme a necessidade da(o) aluna(o).

As classes são caracterizadas pelo tipo: Lâmpada de vapor de mercúrio (*Mercury Vapor* - MV), lâmpada de vapor de sódio de alta pressão (*High Pressure Sodium* - HPS) e lâmpada de vapor metálico (*Metal Halide* – MH) e pela potência: 70, 100, 125, 150, 250 ou 400W da lâmpada. A Tabela 1 mostra a quantidade e distribuição de exemplos por classe. Nota-se que a base é razoavelmente balanceada.

Tabela 1 – Distribuição das exemplos por classe

Classe	Tipo	Potência (W)	Quantidade	(%)
HPS070	HPS	70	30	10,1%
HPS100	HPS	100	32	10,8%
HPS150	HPS	150	35	11,8%
HPS250	HPS	250	33	11,1%
HPS400	HPS	400	37	12,5%
MH150	MH	150	23	7,7%
MH250	MH	250	49	16,5%
MH400	MH	400	37	12,5%
MV125	MV	125	21	7,1%
Total			297	100,0%

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. Use o valor 11 para o parâmetro random_state (`random_state=11`) na inicialização de todos os classificadores que tenham não determinismo (isto é, que tenham um parâmetro random_state) para que chamadas sucessivas não retornem valor diferente e, portanto, tornar os resultados reproduzíveis.

c. Para que os resultados do método HeterogeneousPooling sejam reproduzíveis use o valor 0 para o parâmetro random_state (`random_state=0`) na primeira chamada a resample. A partir daí, use o valor corrente incrementado de 1 nas chamadas sucessivas de resample (random_state = 1, random_state = 2, ...).

d. Use as seguintes funções do scipy.stats para obter os resultados dos testes de hipóteses:

from scipy import stats

stat, p = stats.ttest_rel(scores1, scores2)

...

stat, p = stats.wilcoxon(scores1, scores2)

d. Os gráficos bloxplot requeridos no treino e no teste devem ser gerados usando função específica do pacote seaborn.

e. O apêndice 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 ter um máximo de 5 páginas e ser estruturado da seguinte forma:

1. Título
2. Resumo
3. Seção 1. Introdução
4. Seção 2. Base de Dados
    a. Descrição do Domínio
    b. Definição das Classes e das Características
    c. Número de Instâncias
5. Seção 3. O Método Heterogeneous Pooling
6. Seção 4. Descrição dos Experimentos Realizados e seus Resultados
7. Seção 5. Conclusões
    a. Análise geral dos resultados
    b. Contribuições do Trabalho
    c. Melhorias e trabalhos futuros
8. Referências Bibliográficas

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

6. Condições de Entrega

O trabalho deve ser feito individualmente e submetido pelo sistema da sala virtual até a data limite (04 de junho de 2022).

O trabalho deve ser submetido em dois arquivos: um arquivo pdf com o artigo produzido no trabalho e um arquivo ipynb com o notebook jupyter com o código do trabalho. Tanto o arquivo pdf quanto o arquivo ipynb devem possuir o mesmo nome Trab1_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. Plágio ou cópia de trabalhos serão verificadas. 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

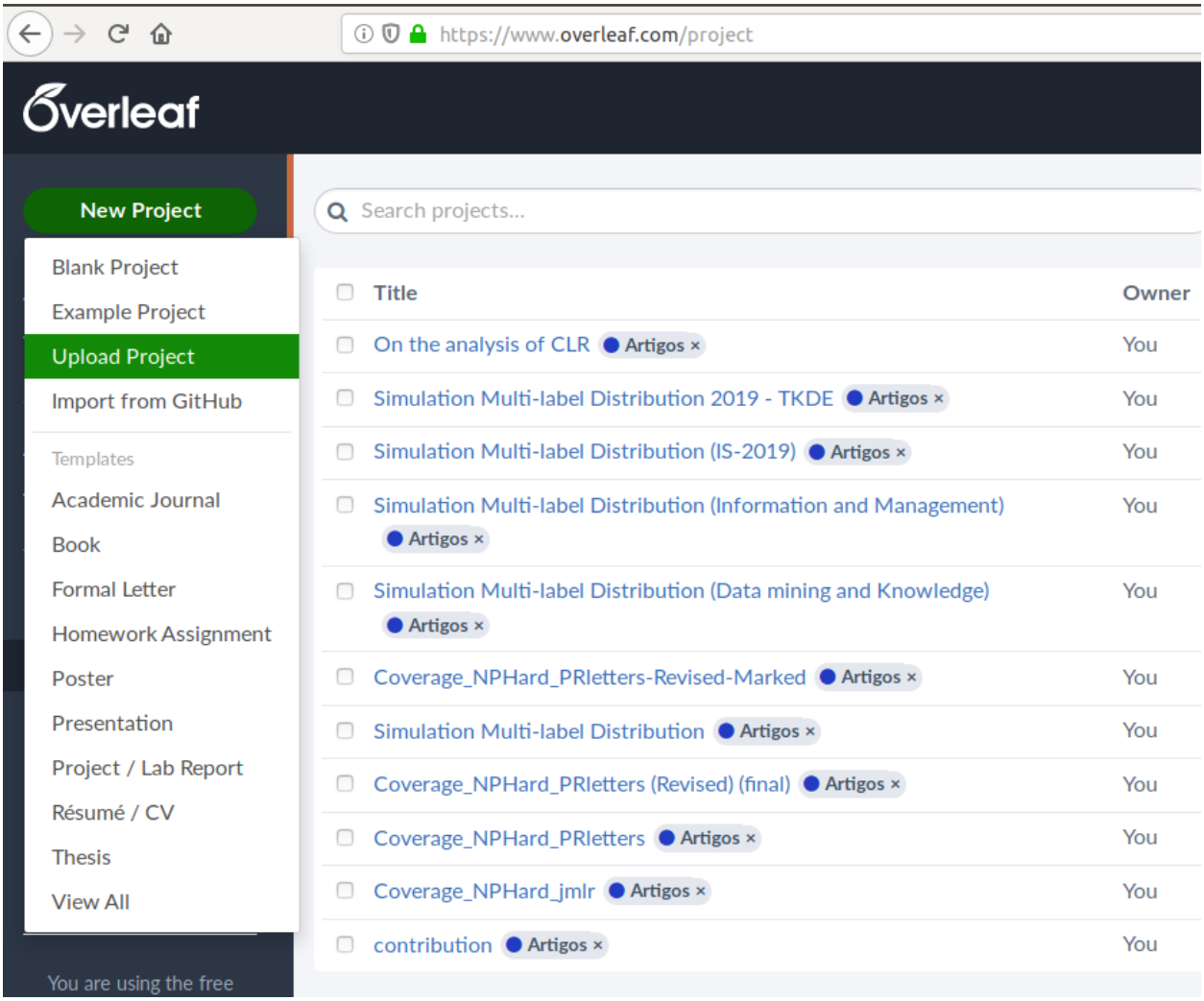
```
def example1():
      mydata=[1,2,3,4,5,6,12]
      sns.boxplot(y=mydata) # Also accepts numpy arrays
      plt.show()

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.


https://www.overleaf.com/project
Overleaf
New Project
Blank Project
Example Project
Upload Project
Import from GitHub
Templates
Academic Journal
Book
Formal Letter
Homework Assignment
Poster
Presentation
Project / Lab Report
Résumé / CV
Thesis
View All
You are using the free
Search projects...
Title
Owner
On the analysis of CLR Artigos × You
Simulation Multi-label Distribution 2019 - TKDE Artigos × You
Simulation Multi-label Distribution (IS-2019) Artigos × You
Simulation Multi-label Distribution (Information and Management) Artigos × You
Simulation Multi-label Distribution (Data mining and Knowledge) Artigos × You
Coverage_NPHard_PRletters-Revised-Marked Artigos × You
Simulation Multi-label Distribution Artigos × You
Coverage_NPHard_PRletters (Revised) (final) Artigos × You
Coverage_NPHard_PRletters Artigos × You
Coverage_NPHard_jmlr Artigos × You
contribution Artigos × You