13/05/2024

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), Naive Bayes (NB), Decision Tree (DT), K Nearest Neighbors (KNN), Multi-layer Perceptron (MLP), Random Forest (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 dos classificadores que não necessitam de ajuste de valores de hiperparâmetros, isto é, os classificadores ZR e NB.

A segunda etapa consiste no treino, validação e teste dos classificadores que precisam de ajuste de hiperparâmetros, isto é, os classificadores DT, KNN, MLP, 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:

Decision Tree: *{'criterion': ['gini','entropy'],*
'max_depth': range(5, 10, 15, 25)}

K Nearest Neighbors: {‘n_neighbors’:[1,3,5,7,9,11,13,15]}

Multi Layer Perceptron: {'hidden_layer_sizes': [(100,),(10,)],
'alpha': [0.0001, 0.005],
'learning_rate': ['constant','adaptive']}

Random Forest: {'n_estimators': [50, 100],
'max_depth': [10, None],
'max_features': ['sqrt', None]}

HeterogeneousPooling: {‘n_samples’: [1, 2, 3, 4, 5, 6, 7, 9]}

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
NB	0.91	0.04	0.87	0.95

DT	0.95	0.02	0.94	0.96
KNN	0.90	0.05	0.89	0.95
MLP	0.96	0.07	0.88	0.99
RF	0.97	0.02	0.95	0.98
HP	0.93	0.01	0.92	0.95

O método HP deve ser implementado. Os métodos ZR, NB, DT, KNN, MLP e RF estão disponíveis no scikit-learn.

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 corrigido** (amostras dependentes) e na matriz triangular inferior devem ser apresentados os resultados do **teste não paramétrico de wilcoxon**. Os valores da célula da tabela que 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	NB	0.105	0.105	0.076
0.096	**0.036**	MLP	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 de um método para seleção de características. O método consiste em selecionar aleatoriamente um número de características, variando de 2 até o (número de características - 1) e, a partir desse número, devem ser selecionadas as N características através de uma roleta onde a chance de uma característica ser escolhida é baseada no seu ANOVA F-value. **Atenção, uma característica não deve ser escolhida mais de uma vez.** Assim, toda vez que uma característica é selecionada a roleta deve ser recalculada somente com as características restantes.

O critério de decisão para classificar uma instância é a 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 métodos (NB, DT, KNN) faça
    - Para cada um dos *n_samples* faça
        - Se for a primeira iteração então
            - Usar a base original para treino dos classificadores
        - Senão
            - Escolher o número de características aleatoriamente
            - Selecionar as características da base utilizando o método da roleta.
        - Fim-se
        - Treinar o classificador na base de treino e incluí-lo no combinado
    - Fim-para
- Fim-para

O pseudo código seguinte mostra como o HP é usado para classificar um exemplo:

- 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 pacientes para classificação da severidade da hemofilia de pacientes hemofílicos. A base completa contém 415 exemplos. O conjunto de dados contém uma variável categórica e 17 numéricas, além da classificação. As classes são hemofilia suave (*Mild)*, moderada (*Moderate)* e severa (*Severe*). A base de dados é razoavelmente balanceada. A classe suave possui 46.2% dos exemplos, a classe moderada possui 17.1%), e a classe severa

possui 36.6%.

Existem duas tarefas distintas que foram delineadas com base nos dados coletados no projeto:

- Tarefa I – Diferenciação entre as 3 classes: Esta tarefa tem como objetivo a classificação dos exemplos nas classes suave, moderada e severa.
- Tarefa II – Identificação de casos severos: Esta tarefa tem como objetivo identificar se a hemofilia é severa, portanto, a classificação dos exemplos é binária, podendo ser severa ou não severa, isto é, as classes suave e moderada são unificadas na classe não severa.

Os alunos serão designados para implementar uma das duas tarefas com base no último dígito de sua matrícula. Alunos cujos números de matrícula terminam de 0 a 4 serão responsáveis pela Tarefa I e aqueles com números finais de 5 a 9 pela Tarefa II.

Faz parte do trabalho fazer o pré-processamento dos dados, o qual envolve a codificação da característica categórica, a codificação da classe e a padronização das características numéricas.

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. Use a seguinte função do scipy.stats para obter os resultados do teste de hipóteses de wilcoxon:

from scipy import stats

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

d. O teste t corrigido deve ser calculado conforme função apresentada na aula, tal como descrito no apêndice A deste enunciado.

e. Os gráficos bloxplot requeridos devem ser gerados usando função específica do pacote seaborn, conforme apêndice B deste enunciado.

f. O apêndice C 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 (17 de junho de 2024).

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. Implementação do teste t corrigido

```
from scipy.stats import t
from math import sqrt
from statistics import stdev

def corrected_dependent_ttest(data1, data2, n_training_samples,
n_test_samples):
    n = len(data1)
    differences = [(data1[i]-data2[i]) for i in range(n)]
    sd = stdev(differences)
    divisor = 1 / n * sum(differences)
    test_training_ratio = n_test_samples / n_training_samples
    denominator = sqrt(1 / n + test_training_ratio) * sd
    t_stat = divisor / denominator
    # degrees of freedom
    df = n - 1

    # calculate the p-value
    p = (1.0 - t.cdf(abs(t_stat), df)) * 2.0
    # return everything
    return t_stat, p

print('Corrected T Test')
s,p =  corrected_dependent_ttest (scores, dTScores,  (len(iris_X)*3)/4,
len(iris_X)/4)
print("t: %0.2f p-value: %0.2f\n" % (s,p))
```

Apêndice B. 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.
```

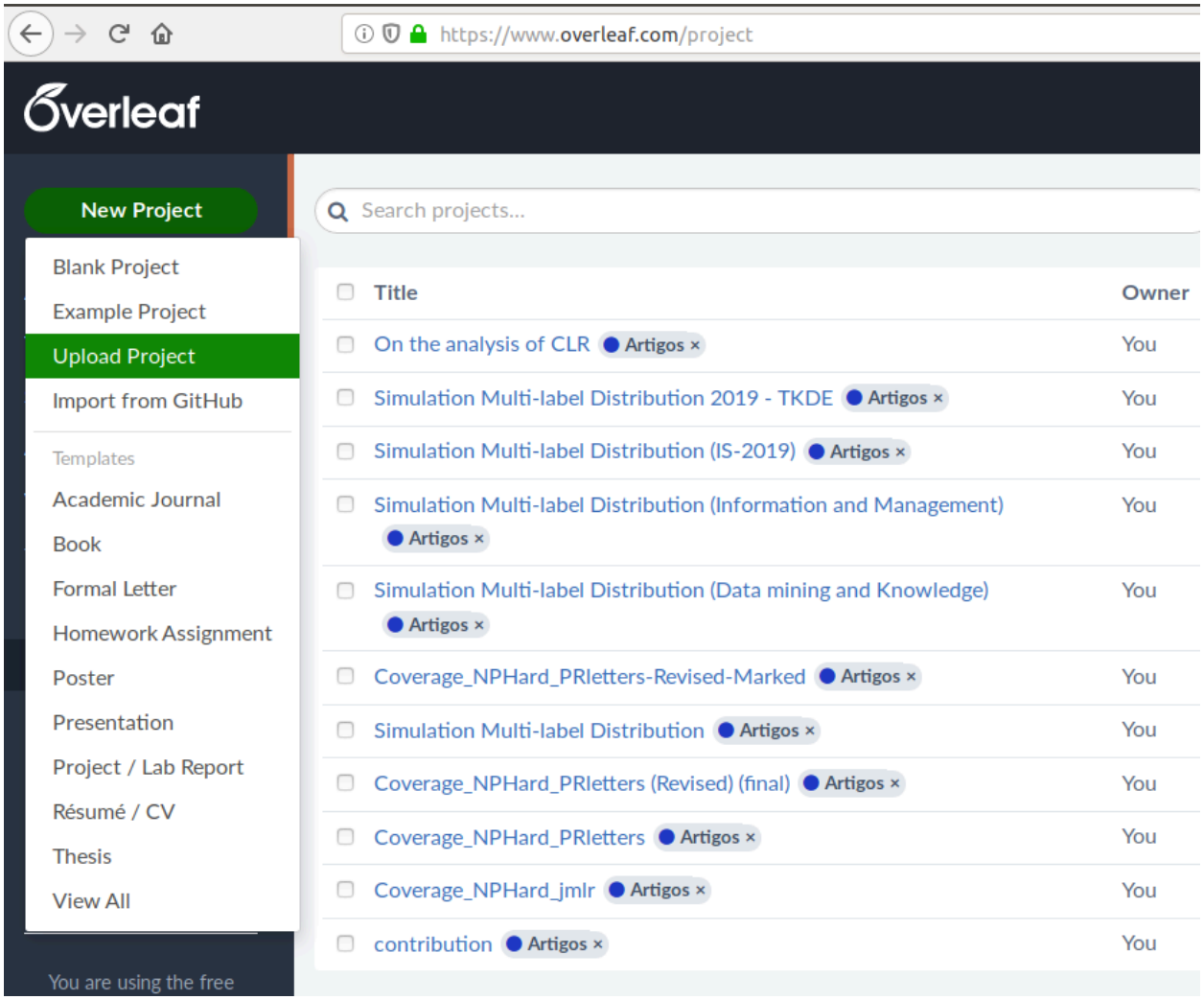
```
        #print(df)
        # Make boxplot for each group
        sns.boxplot( data=df.loc[:,:] )
        # loc[:,:] means all lines and all columns
        plt.show()

example1()
example2()
```

Apêndice C. One Hot Encode

```
for ohe_enc in ['NOME DA SUA COLUNA']:

  unique_values = df[ohe_enc].unique()

  for value in unique_values:
    df[ohe_enc+"_"+value.replace("
","").lower()]=df[ohe_enc].str.contains(value)

  df.drop([ohe_enc], axis=1, inplace=True)
```

Apêndice D. 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
Search projects...
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
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