09/06/2025

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: Decision Tree (DT), K Nearest Neighbors (KNN), Multi-layer Perceptron (MLP), Random Forest (RF) e Heterogeneous Boosting (HB). O procedimento experimental 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': [5, 10, 15, 25]}
K Nearest Neighbors: {'n_neighbors':[1,3,5,7,9]}
Multi Layer Perceptron: {'hidden_layer_sizes': [(100,),(10,)],
'alpha': [0.0001, 0.005],
'learning_rate': ['constant','adaptive']}
Random Forest: {'n_estimators': [5, 10, 15, 25],
'max_depth': [10, None]}
HeterogeneousBoosting: {'n_estimators': [5, 10, 15, 25, 50]}

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
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
HB	0.93	0.01	0.92	0.95

O método HB deve ser implementado. Os métodos 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** (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.

DT	0.085	**0.045**	0.065	0.089
0.045	KNN	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	HB

2. HB

O classificador Heterogeneous Boosting (HB) é um combinado de classificadores heterogêneos que usa como classificadores base: DT, Naive Bayes Gaussiano (NB), MLP e KNN, sempre com valores default do sklearn para seus hiperparâmetros. O único parâmetro do método HB é o *n_estimators*, que indica o número de classificadores base que comporão o combinado. Por exemplo, se *n_estimators* é igual a 9, o combinado será composto por 9 classificadores. 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 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 HB é obtido a partir de uma base de dados de treino:

- Obter e guardar a classe mais frequente da base de treino
- Inicializar o vetor de pesos dos exemplos da base de treino com valor 1 para todos os exemplos
- Repita *n_estimators* vezes
    - Selecionar com reposição os exemplos da base de treino utilizando o método da roleta
    - Para cada um dos métodos de classificação (NB, DT, MLP, KNN) faça
        - Treinar o classificador com todos os exemplos selecionados
        - Testar o classificador em todos os exemplos selecionados
    - Fim-para
    - Escolher o classificador com melhor desempenho para compor o combinado
        - Em caso de empate, usar a seguinte preferência: MLP, DT, KNN, NB

- Dobrar o peso dos exemplos classificados de forma errônea pelo classificador incorporado ao combinado
    - Fim-para
- Fim-repita

O pseudo código seguinte mostra como o HB 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 frequente na base de treino dentre as que empataram como mais votada
- Senão
    - Retornar a classe mais votada
- Fim-se

3. Base de Dados

A base de dados usada no trabalho foi obtida do trabalho de conclusão de curso (TCC) do egresso do curso de Ciência da Computação Lucas Frinhani. No TCC foi desenvolvido um sistema de predição de resultados de apostas esportivas para o jogo *Leagle of Legends (LoL)*. A base completa contém 8152 exemplos de partidas da temporada 2021. O conjunto de dados contém trinta e oito características numéricas, uma característica binária indicando a equipe vencedora (usada como classe), além do identificador numérico (*id*) da partida, que deve ser descartado pois não interfere no resultado.

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

- Tarefa I – Predição de vencedor usando apenas dados de pré-jogo
- Tarefa II – Predição de vencedor usando apenas dados coletados aos 10 minutos de jogo
- Tarefa III – Predição de vencedor usando dados de pré-jogo e dados coletados aos 10 minutos de jogo
- Tarefa IV – Predição de vencedor usando apenas dados coletados aos 15 minutos de jogo
- Tarefa V – Predição de vencedor usando dados de pré-jogo, dados coletados aos 10 minutos de jogo e dados coletados aos 15 minutos de jogo

Os alunos serão designados para implementar uma das tarefas com base no último dígito de sua matrícula. Alunos cujos números de matrícula terminam em 1 e 6 serão responsáveis pela

Tarefa I, aqueles com números finais 2 e 7 fazem a Tarefa II, aqueles com números finais 3 e 8 fazem a Tarefa III, aqueles com números finais 4 e 9 fazem a Tarefa IV e aqueles com números finais 5 e 0 fazem a Tarefa V.

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 36854321 para o parâmetro random_state (`random_state=36854321`) nas chamadas a RepeatedStratifiedKFold para que os resultados sejam reproduzíveis.

b. Use o valor 11 para o parâmetro random_state (`random_state=13`) 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 as seguinte funções do scipy.stats para obter os resultados do testes de hipóteses:

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 boxplot 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
    d. Distribuição de Classes
5. Seção 3. O Método Heterogeneous Boosting
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 (9 de julho de 2025).

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 por Nadeau e Bengio

```
def t_corrigido_nadeau_bengio(data1, data2, X, n_folds_externos):
    """
    Parâmetros:
        data1, data2: listas ou arrays com as acurácias
```

```
        X: conjunto de dados original
        n_folds_externos: número de folds no loop externo
    Retorna:
        t_stat: valor da estatística t
        p_valor: valor-p do teste bilateral
    """

    N = len(X)  # número total de amostras no dataset
    n = len(data1)  # número de execuções

    # Tamanhos dos conjuntos de treino/teste em cada fold externo
    n_test = N // n_folds_externos
    n_train = N - n_test

    # Cálculo da estatística t com correção
    diffs = np.array(data1) - np.array(data2)
    mean_diff = np.mean(diffs)
    std_diff = np.std(diffs, ddof=1)

    se_corrigido = std_diff * np.sqrt(1/n + n_test/n_train)
    t_stat = mean_diff / se_corrigido
    p_valor = 2 * (1 - t.cdf(abs(t_stat), df=n - 1))

    return t_stat, p_valor

print('\nCorrected T Test')
s,p = t_corrigido_nadeau_bengio(scores,scoresWN,iris_X,10)
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. 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.