Introdução a Inteligência Artificial DCC024

Aluno: Paulo Henrique Moreira Melo

ATENÇÃO
PARA FAZER ESTE TP EU RENOMEEI OS CSVs recebidos
Iris aleatório virou iris.csv
Iris treino virou treino.csv
Iris teste virou teste.csv

Fiz isso pq o python não lida bem com arquivos com espaços. Inclui os arquivos renomeados no zip de entrega

K NEAREST NEIGHBOR

KNN.py

Para implementação do algoritmo criou-se a **classe KNN** que possui 3 atributos:

- **KNN.k** guarda o valor de quantos vizinhos serão analisados.
- **KNN.X_train** guarda o array com as colunas 4 correspondente às larguras e comprimentos das pétalas e sépalas
- **KNN.y_train** guarda um array composto por inteiros 0,1 ou 2 com os valores correspondendo respectivamente iris setosa, iris virginica e iris versicolor

A **classe KNN** possui 4 métodos:

- **__init__(self, k=3):** o construtor que recebe qual valor de k
- **ler_dados(self, X, y)** um método "setter" que recebe os valores de X_train e y_train
- **prever(self, X)** método que recebe o array teste de 4 colunas e passa linha por linha para o método preveja(self, x). Então uma lista com os rótulos previstos é montada e então retornada
- **preveja(self, x)** recebe a linha com o numpy.array com os 4 números (que representa um ponto em um espaço de 4 dimensões. Gera-se uma lista com as distâncias euclidianas de cada ponto com os pontos já rotulados. Depois, organiza as distâncias em ordem crescente e retorna os índices dos primeiros k vizinhos. Depois, extrai os rótulos dos k vizinhos mais próximos. Depois, pega-se o rótulo mais comum. Este rótulo é retornado

Outros métodos:

- **rotulos_para_numeros(mylist)** transforma os rótulos strings em inteiros
- **estatistica(y_true,y_pred)** calcula a revocação, f1 e precisão e acurácia

Discussão:

Introdução a Inteligência Artificial DCC024

Aluno: Paulo Henrique Moreira Melo

```
com K=2
                precisao  revocacao        f1
iris setosa     1.000000   1.000000  1.000000
iris virginica  1.000000   0.777778  0.875000
iris versicolor 0.857143   1.000000  0.923077
valor da acuracia: 0.9333333333333333
                iris setosa  iris virginica  iris versicolor
iris setosa               9               0                0
iris virginica            0               7                2
iris versicolor           0               0               12
```

```
com K=3
                precisao  revocacao        f1
iris setosa     1.000000   1.000000  1.000000
iris virginica  1.000000   0.777778  0.875000
iris versicolor 0.857143   1.000000  0.923077
                iris setosa  iris virginica  iris versicolor
iris setosa               9               0                0
iris virginica            0               7                2
iris versicolor           0               0               12
valor da acuracia: 0.9333333333333333
```

```
com K=5
                precisao  revocacao   f1
iris setosa          1.0        1.0  1.0
iris virginica       1.0        1.0  1.0
iris versicolor      1.0        1.0  1.0
                iris setosa  iris virginica  iris versicolor
iris setosa               9               0                0
iris virginica            0               9                0
iris versicolor           0               0               12
valor da acuracia: 1.0
```

Introdução a Inteligência Artificial DCC024

Aluno: Paulo Henrique Moreira Melo

```
com K=8
                precisao  revocacao        f1
iris setosa     1.000000   1.000000  1.000000
iris virginica  1.000000   0.888889  0.941176
iris versicolor 0.923077   1.000000  0.960000
                iris setosa  iris virginica  iris versicolor
iris setosa               9               0                0
iris virginica            0               8                1
iris versicolor           0               0               12
valor da acuracia: 0.9666666666666667
```

```
com K=32
                precisao  revocacao        f1
iris setosa     1.000000   1.000000  1.000000
iris virginica  1.000000   0.888889  0.941176
iris versicolor 0.923077   1.000000  0.960000
                iris setosa  iris virginica  iris versicolor
iris setosa               9               0                0
iris virginica            0               8                1
iris versicolor           0               0               12
valor da acuracia: 0.9666666666666667
```

Vê-se que com K muito pequeno é difícil de definir bem em que grupo qual ponto pertence, aumentando o K, tem-se um resultado melhor. A melhora máxima ocorreu quando K=5, ou seja, valendo-se do ponto mais comum dos 5 vizinhos mais próximos a determinado ponto para conseguir definir a qual grupo este ponto pertence. Nota-se que eu também plotei a matriz de confusão de cada valor k.

K-MEANS:

Kmeans.py

Para implementação do algoritmo criou-se a **classe Kmeans** que possui 3 atributos:

- **Kmeans.interacoes_maximas** que define um valor máximo para iterações do algoritmo
- **Kmeans.n_amostras** que é o número de pontos (amostras de flor) no espaço 4 dimensional

Introdução a Inteligência Artificial DCC024

Aluno: Paulo Henrique Moreira Melo

- **Kmeans.n_features** que é o número de dimensões (o número de colunas), neste caso, 4.
- **Kmeans.n_clusters** número de clusters
- **Kmeans.centroides** lista com os centroides

A **classe Kmeans** possui 8 métodos, sendo **prever(self, X)** o método principal:

prever(self, X) onde o algoritmo se desenrola. O algoritmo se desenrola em um loop de até **Kmeans.interacoes_maximas** de vezes. O algoritmo começa inicializando os centróides com valores e índices aleatórios. Depois ele otimiza os clusters dentro do loop: ele começa assinalando cada amostra a seu centróide mais próximo. Depois ele calcula novos centróides usando os novos clusters com a função **pegar_centroide(self, clusters),** depois usando a função **sao_iguais(self, centroides_antigos, centroides)** checa-se os centróides antigos são iguais aos novos centróides. Se sim, a função deixou de melhorar, e ele já retorna. A função retorna com o valor do cluster que cada ponto foi assinalado (com seu "rótulo" de mentirinha)

Outros métodos:

- **rotulos_para_numeros(mylist)** transforma os rótulos strings em inteiros

Discussão:

```
Valor dos centroides com K=2
   sepal_length   sepal_width  petal_length  petal_width
0      5.005660      3.360377      1.562264     0.288679
1      6.301031      2.886598      4.958763     1.695876


Valor dos centroides com K=3
   sepal_length   sepal_width  petal_length  petal_width
0      6.853846      3.076923      5.715385     2.053846
1      5.883607      2.740984      4.388525     1.434426
2      5.006000      3.418000      1.464000     0.244000
```

```
 agora vamos calcular os centroides de cada conjunto de dados separadamente! Primeiro o centroide da setosa, depois o centroide da v
irginica e depois da versicolor!
[[5.006 3.418 1.464 0.244]]
[[6.588 2.974 5.552 2.026]]
[[5.936 2.77  4.26  1.326]]
```

Não é possível calcular acurácia, pois não sabemos quais são os supostos rótulos que poderiam ser sabidos. Porém, separei os dados originais de em 3 arrays separados, um array para setosa, um array para virginica e um array para versicolor. E o que se viu é que, com k=3, o centróide da setosa correspondeu ao index terceiro index, a virginica ao primeiro index e a

versicolor ao segundo index. Outra observação muito interessante fazemos com k=2, observa-se que praticamente a iris_setosa se organiza em um centróide com valores principalmente dela, enquanto o segundo centróide é uma mistura dos valores da iris_virginica e iris_versicolor. Com estes dados, poderíamos imaginar que geneticamente as iris_verginica e iris_versicolor são mais parecidas entre si e iris_setosa é uma parte mais isolada.