Uma imagem com texto

Descrição gerada automaticamente

Supervised Learning – a4rl

André Filipe Frade Guerra

Novembro/2022

Conteúdo

[Exercício 1 3](#_Toc120453657)

[Exercício 2 5](#_Toc120453658)

[Exercício 3 9](#_Toc120453659)

[Exercício 4 9](#_Toc120453660)

## Exercício 1

No exercício 1 começamos por inicializar as variáveis com os valores necessários para correr o algoritmo: Uma imagem com texto, interior

Descrição gerada automaticamente

De seguida, o inicio do algoritmo que começa por repetir o algoritmo tendo em conta o numero de alfas no array da variavél e o numero de ciclos pretendido, neste caso 30, atribuindo sempre novos valores aleatorios aos w’s:

Uma imagem com texto, monitor, ecrã, captura de ecrã

Descrição gerada automaticamente

Inicia-se o algoritmo, que vai atualizar os valores w’s em cada interação, até que o resultado do erro seja 0:

Uma imagem com texto

Descrição gerada automaticamente

Por fim podemos visualizar o resultado do exercício:

Uma imagem com texto, captura de ecrã, ecrã, prateado

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Podemos concluir que quando o alfa aumenta de valor, o número de ciclos para chegar ao resultado é inferior.

## Exercício 2

No exercício 2 começamos por inicializar as variáveis com os valores necessários para correr o algoritmo e ver o ficheiro iris.data que contem os dados a serem utilizados no algoritmo, de seguida, o inicio do algoritmo que começa por repetir o algoritmo tendo em conta o numero de “k”’s no array da variavél:Uma imagem com texto

Descrição gerada automaticamente

Neste algoritmo foi criada uma classe “Iris” que guarda os valores ao ler o ficheiro e a distância que é usada mais a frente pelo algoritmo:

Uma imagem com texto, ecrã

Descrição gerada automaticamente

O método “ReadFile” lê o ficheiro e cria os objetos em arrays especificos, e tambem retorna 1 array com todos os objetos “dataSet”, array que é usado no algoritmo:

Uma imagem com texto

Descrição gerada automaticamente

O algoritmo começa com o metodo Evaluate, que corre o numero de vezes que pretendemos, neste caso 10, e no fim mostra a média do numero de previsoes corretas:

Uma imagem com texto, captura de ecrã, ecrã

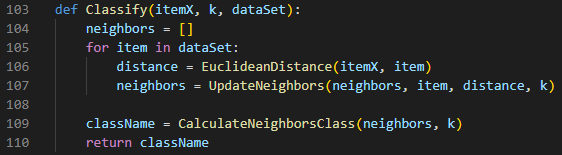
Descrição gerada automaticamente

O método “K\_FoldValidation” separa os dados em 2 arrays, o de testes e o de treino, percorre todos os dados no array de teste que são “treinados” tendo em conta o array de treino, e calcula no fim a precisão, verificando se a classe do dado a ser treinado é igual ao encontrado pelo algoritmo:

Uma imagem com texto

Descrição gerada automaticamente

O método “Classify” para cada dado de teste a ser classificado, vai definir a classe a que pertence “versicolor”, “setosa” ou “virginica”, tendo em conta os dados de treino:



O método “EuclideanDistance” calcula a distância euclidiana tendo em conta os 4 parâmetros de cada objeto:

Uma imagem com texto

Descrição gerada automaticamente

O método “UpdateNeighbors” escolhe o numero “k” ([3, 4, 7, 11]) de vizinhos mais próximos do ponto a ser avaliado, tendo em conta a distância anteriormente calculada:

Uma imagem com texto

Descrição gerada automaticamente

Por fim o método “CalculateNeighborsClass”, retorna a classe a que o ponto em avaliação pode pertencer, tendo em conta a classe do maior número de vizinhos mais próximos, anteriormente calculado:

Uma imagem com texto

Descrição gerada automaticamente

Ao corrermos o algoritmo obtemos o seguinte resultado:

Uma imagem com texto

Descrição gerada automaticamente

Assim, quando o valor de k é maior o número de previsões corretas é maior, até certo k. Podemos verificar que quando o k é 11, o algoritmo tem mais eficácia na previsão da classe.

## Exercício 3



## Exercício 4

No exercício 4 foram utilizadas bibliotecas externas que implementam o algoritmo:

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto, eletrónica, teclado

Descrição gerada automaticamente

Este algoritmo é mais consistente nos resultados obtendo sempre 3 falhas na previsão.