KNN演算法

資料集：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sepallength | sepalwidth | petallength | petalwidth | class |
| 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 4.9 | 3 | 1.4 | 0.2 | Iris-setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 7 | 3.2 | 4.7 | 1.4 | Iris-versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | Iris-versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | Iris-versicolor |
| 5.5 | 2.3 | 4 | 1.3 | Iris-versicolor |
| 6.3 | 3.3 | 6 | 2.5 | Iris-virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | Iris-virginica |
| 7.1 | 3 | 5.9 | 2.1 | Iris-virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | Iris-virginica |
| 5 | 3.6 | 1.4 | 0.2 | ? |
| 5.7 | 2.8 | 4.5 | 1.3 | ? |
| 7.6 | 3 | 6.6 | 2.1 | ? |

實驗目標：找出上述三個?的class種類

實作程式：

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  //鳶尾花結構  struct IRIS{  double sepallength;  double sepalwidth;  double petallength;  double petalwidth;  short iris\_class;  };  //鳶尾花類別字串  char irisClassName[3][30] = {"Iris-setosa","Iris-versicolor","Iris-virginica"};  //鳶尾花資料集  IRIS iris[] = { {5.1,3.5,1.4,0.2,0},  {4.9,3,1.4,0.2,0},  {4.7,3.2,1.3,0.2,0},  {4.6,3.1,1.5,0.2,0},  {7,3.2,4.7,1.4,1},  {6.4,3.2,4.5,1.5,1},  {6.9,3.1,4.9,1.5,1},  {5.5,2.3,4,1.3,1},  {6.3,3.3,6,2.5,2},  {5.8,2.7,5.1,1.9,2},  {7.1,3,5.9,2.1,2},  {6.3,2.9,5.6,1.8,2}};  //鳶尾花報表  void Print(IRIS x){  cout << "Sepallength:" << x.sepallength << "\tSepalwidth:" << x.sepalwidth;  cout << "\tPetallength:" << x.petallength << "\tPetalwidth:" << x.petalwidth;  cout << "\t鳶尾花種類:" << irisClassName[x.iris\_class] << endl;  }  //KNN計算並回傳測資的鳶尾花種類  int KNN(IRIS in,int K){  double weight[2][sizeof(iris)/sizeof(iris[0])],temp0,temp1;  short class\_tmp[3];  class\_tmp[0] = 0;  class\_tmp[1] = 0;  class\_tmp[2] = 0;  //測資與資料集的向量計算  for(int i = 0;i < sizeof(iris)/sizeof(iris[0]);i++){  weight[0][i]=i;  weight[1][i]=(iris[i].petallength\*in.petallength+iris[i].petalwidth\*in.petalwidth+iris[i].sepallength\*in.sepallength+iris[i].sepalwidth\*in.sepalwidth)/  (sqrt(iris[i].petallength\*iris[i].petallength+iris[i].petalwidth\*iris[i].petalwidth+iris[i].sepallength\*iris[i].sepallength+iris[i].sepalwidth\*iris[i].sepalwidth)+  sqrt(in.petallength\*in.petallength+in.petalwidth\*in.petalwidth+in.sepallength\*in.sepallength+in.sepalwidth\*in.sepalwidth));  }  //氣泡排序取得最近的K個鳶尾花  for(int x=0;x < sizeof(iris)/sizeof(iris[0])-1;x++){  for(int y=0;y < sizeof(iris)/sizeof(iris[0])-1-x;y++){  if (weight[1][y] < weight[1][y + 1]) {  temp1 = weight[1][y];  weight[1][y] = weight[1][y+1];  weight[1][y+1] = temp1;  temp0 = weight[0][y];  weight[0][y] = weight[0][y+1];  weight[0][y+1] = temp0;  }  }  }  for(int i=0;i<K;i++){  class\_tmp[iris[(int)weight[0][i]].iris\_class]++;  }  //鄰居最多的判斷式  if(class\_tmp[0] > class\_tmp[1]){  if(class\_tmp[0]>class\_tmp[2]){  return 0;  }else{  return 2;  }  }else{  if(class\_tmp[1] > class\_tmp[2]){  return 1;  }else{  return 2;  }  }  return 0;  }  int main(){  IRIS input;  int K;  cout << "鳶尾花資料集：" << endl;  for(int i = 0;i < sizeof(iris)/sizeof(iris[0]);i++){  Print(iris[i]);  }  cout << "請輸入K值(不可大於" << sizeof(iris)/sizeof(iris[0]) << ")：";  cin >> K;  cout << "請輸入測資(輸入0 0 0 0結束):";  while(cin >> input.petallength >> input.petalwidth >> input.sepallength >> input.sepalwidth){  if((input.petallength == input.petalwidth) && (input.petalwidth == input.sepallength) && (input.sepallength == input.sepalwidth)){  return 0;  }  cout << "你所輸入的鳶尾花為：" << endl;  input.iris\_class = KNN(input,K);  Print(input);  }  return 0;  } |