

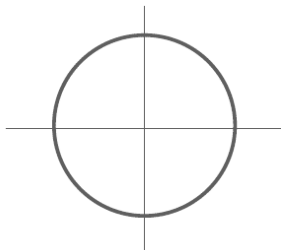
Nama : Muhammad Alfisar Rachman
NIM : 1301164036
Kelas : IF-40-08

Problem :

Melakukan klasifikasi terhadap 200 data apakah 200 data tersebut termasuk kedalam label 0,1,2, atau 3 , dimana sudah diberikan 800 data yang sudah di ketahui labelnya dan 200 data yang belum di ketahui labelnya. Dalam tugas ini pencarian klasifikasi 200 data diharuskan menggunakan dan mengimplementasikan ***k-Nearest Neighbors***.

Strategi Penyelesaian Masalah :

Strategi yang digunakan yaitu dengan mengimplementasikan ***k-Nearest Neighbors*** kedalam sebuah program sehingga program tersebut dapat menentukan label dari 200 data yang sudah di berikan berdasarkan jumlah K yang didapat. Dalam program yang saya buat, saya mengambil data train sebanyak 320 untuk di cari kembali labelnya dan hasil label baru dari 320 data tersebut akan saya bandingkan dengan label lama dari 320 data tersebut, hasil dari perbandingan itu akan mendapatkan K ke berapa yang mempunyai nilai akurasi paling tinggi. Untuk pencarian label tersebut harus melakukan pencarian jarak dari masing-masing data pada 320 data tersebut akan di hitung jaraknya ke 480 data sisanya, berikut rumus pencarian jarak yang saya pakai dan pengimplementasinya di algoritma program saya.



L2 (Euclidean) distance

$$d_1(x_1, y_2) = \sqrt{\sum_p (x_1^p - y_2^p)^2}$$

```
def jarak(v1,w1,x1,y1,z1,v2,w2,x2,y2,z2):  
    distance = ((v2-v1)**2) + ((w2-w1)**2) + ((x2-x1)**2) + ((y2-y1)**2) + ((z2-z1)**2)  
    return math.sqrt(distance)
```

Setelah jarak yang di cari sudah dapat, maka program akan melakukan sorting menurut jarak yang sudah di cari tadi, setelah itu program akan mengelompokkan data tersebut memiliki label apa menurut jumlah data yang diambil berdasarkan jumlah K nya. Setelah menemukan K yang paling optimum maka K tersebut akan di pakai pada program yang nantinya akan di gunakan untuk mencari label dari 200 data yang di berikan.

Nama : Muhammad Alfisar Rachman
NIM : 1301164036
Kelas : IF-40-08

Analisis :

Berdasarkan penjelasan yang saya jabarkan maka dapat diketahui bahwa pengambilan data train yang di jadikan validasi sangat berpengaruh terhadap pencarian K optimumnya, selain itu pemakaian rumus untuk menghitung jarak ternyata berperan penting terhadap pencarian Knya. Selain pengambilan data dan juga rumus untuk menghitung jarak ternyata K lah yang sangat berperan penting terhadap tingkat akurasi data validasi dan K lah yang akan menentukan seberapa bagus hasil label dari 200 data tersebut

Hasil Percobaan :

K - Optimum

```
(12, 12)
88.125
(13, 13)
88.125
(14, 14)
88.125
('K optimum : ', 14)
('Akurasi : ', 88.125)
```

Hasil Running program

```
PROBLEMS OUTPUT
mylinux@mylinux:~$ python her.py --client 1
clearcd: command not found
(1, 1)
(2, 1)
(3, 1)
(4, 1)
(5, 1)
(6, 1)
(7, 1)
(8, 1)
(9, 1)
(10, 1)
(11, 1)
(12, 0)
(13, 1)
(14, 1)
(15, 0)
(16, 1)
(17, 0)
(18, 0)
(19, 0)
(20, 0)
(21, 1)
(22, 0)
(23, 0)
(24, 1)
(25, 1)
(26, 0)
(27, 0)
(28, 0)
(29, 1)
(30, 0)
(31, 0)
(32, 1)
(33, 0)
(34, 1)
(35, 1)
(36, 1)
(37, 1)
(38, 1)
(39, 0)
(40, 1)
(41, 0)
(42, 0)
(43, 2)
(44, 0)
(45, 0)
(46, 0)
(47, 0)
(48, 1)
(49, 0)
(50, 1)
(51, 1)
(52, 1)
(53, 0)
(54, 1)
(55, 1)
(56, 3)
(57, 1)
(58, 1)
(59, 1)
(60, 1)
(61, 1)
(62, 1)
(63, 1)
(64, 1)
(65, 0)
(66, 1)
(67, 0)
(68, 1)
(69, 3)
(70, 1)
(71, 0)
(72, 0)
(73, 1)
(74, 1)
(75, 1)
(76, 1)
(77, 2)
(78, 0)
(79, 1)
(80, 1)
(81, 1)
(82, 0)
(83, 0)
(84, 1)
(85, 1)
(86, 1)
(87, 0)
(88, 0)
(89, 0)
(90, 0)
(91, 0)
(92, 0)
(93, 1)
(94, 1)
(95, 0)
(96, 0)
(97, 0)
(98, 1)
(99, 0)
(100, 2)
(101, 2)
(102, 2)
(103, 3)
(104, 3)
(105, 2)
(106, 2)
(107, 3)
(108, 3)
(109, 2)
(110, 3)
(111, 3)
(112, 0)
(113, 3)
(114, 3)
(115, 2)
(116, 2)
(117, 2)
(118, 2)
(119, 2)
(120, 2)
(121, 2)
(122, 3)
(123, 2)
(124, 2)
(125, 3)
(126, 3)
(127, 2)
(128, 2)
(129, 3)
(130, 2)
(131, 2)
(132, 3)
(133, 2)
(134, 3)
(135, 3)
(136, 2)
(137, 2)
(138, 2)
(139, 2)
(140, 2)
(141, 2)
(142, 0)
(143, 3)
(144, 2)
(145, 2)
(146, 2)
(147, 3)
(148, 3)
(149, 3)
(150, 2)
(151, 2)
(152, 3)
(153, 3)
(154, 3)
(155, 3)
(156, 2)
(157, 3)
(158, 0)
(159, 2)
(160, 3)
(161, 2)
(162, 3)
(163, 3)
(164, 2)
(165, 3)
(166, 2)
(167, 3)
(168, 2)
(169, 2)
(170, 2)
(171, 2)
(172, 3)
(173, 0)
(174, 2)
(175, 2)
(176, 2)
(177, 2)
(178, 2)
(179, 3)
(180, 3)
(181, 3)
(182, 3)
(183, 2)
(184, 2)
(185, 2)
(186, 2)
(187, 3)
(188, 2)
(189, 3)
(190, 3)
(191, 2)
(192, 3)
(193, 2)
(194, 3)
(195, 3)
(196, 3)
(197, 3)
(198, 3)
(199, 3)
(200, 3)
```