

Nama =Farhan Fadhlullah
Nim=1301164123
Kelas=IF 40-08

DESKRIPSI MASALAH

Dari Tupro 3 ini terdapat 2 data data yaitu data train dan test,digunakan untuk menentukan klaisfikasi dari suatu class yang mempunyai beberapa atribut untuk menentukan klasifikasinya. Dalam tugas program 3 ini membahas tentang k nearest neighbor(knn), tahapan untuk menyelesaikan knn yaitu: harus menentukan parameter k terlebih dahulu, berapa nilai k yang akan digunakan,lalu menghitung jaraknya,dalam knn penghitungan jarak menggunakan 2 metode yaitu Manhattan dan Euclidean, dalam penghitungan jarak ini yang di ambil ialah data train untuk menghitung jarak kuadrat, setelah itu akan mendapatkan hasil yang di inginkan sesuai dengan metode yang digunakan manhattan atau Euclidean, hasil yang didapat diurutkan secara ascending,lalu mengumpulkan kategori y (klasifikasi knn berdasarkan nilai k), dengan menggunakan kategori knn yang paling mayoritas maka dapat di prediksi kategori objek tersebut.terdapat DataTrain.csv yang memiliki 5 atribut (X1, X2, X3, X4, X5) dan memiliki 1 output yang memiliki 4 kelas/label (0, 1, 2, 3). Dalam tupro 3 ini masalah yang diberikan adalah membangun sebuah sistem klasifikasi dengan metode K-NN untuk menentukan klasifikasi atau label data testing dalam file DataTest_Tugas3.csv. Dalam Data train terdapat 800 himpunan data. Sistem akan membaca masukan file DataTrain_Tugas3ai.csv dan DataTest_Tugas3ai.csv dan mengeluarkan output berupa file TebakanTugas3ai berupa baris 200 baris angka bernilai integer(0,1,2,3)yang menjadi kelas/label baris yang sesuai pada file DataTest_Tugas3_AI

ANALISIS DAN STRATEGI

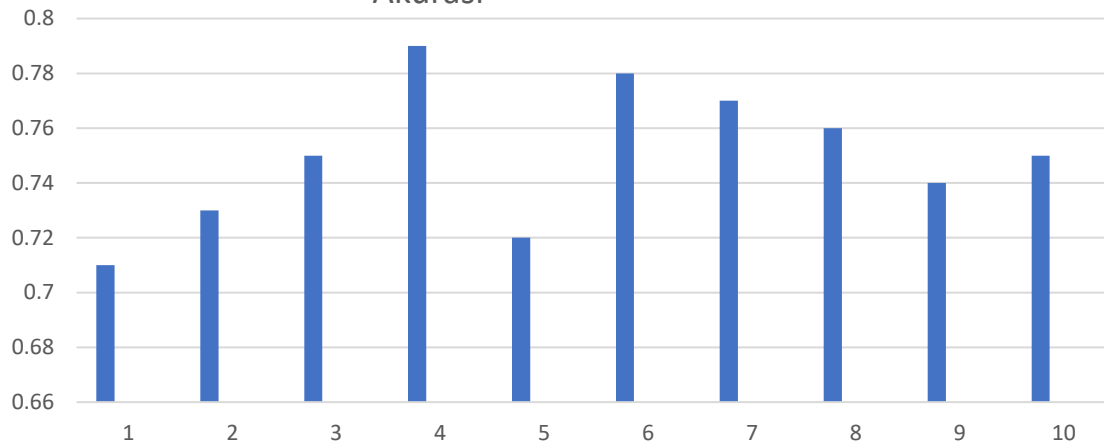
Dalam menyelesaikan permasalahan yang diberikan di Tupro 3 ini adalah membuat program dengan menggunakan metode knn dengan cara membaca file Data_Test_Tugas3AI.csv dan DataTrain_Tugas3_AI.csv dan mengeluarkan 200 baris TebakanTugas3.csv.

Dalam metode ini program akan membaca file Data train dulu,lalu akan membaca data test yang digunakan untuk bisa membandingkan jarak dari setiap data test dengan data train. Dalam menghitung jarak terdapat 2 metode yaitu metode Manhattan dan metode Euclidean. Setiap baris data test dibandingkan dengan seluruh data train.lalu dicari jarak terdekat dengan nilai k yang telah ditentukan.

Pada tahapan knn harus menentukan parameter k, menghitung kuadrat jarak baik menggunakan Euclidean atau manhattan terhadap objek data training,lalu di urutkan hasilnya secara ascending, mengumpulkan kategori y, dengan menggunakan kategori nearest neighbor yang paling mayoritas maka dapat diprediksi kategori objek.pada system knn ini biasanya akan membaca file data test, pada setiap data test yang dibaca akan dicari jaraknya dengan data train ,jarak yang di cari pada program saya menggunakan metode euclidean, lalu di urutkan berdasarkan data dari yang terkecil ke terbesar lalu dicari data terdekatnya sebanyak k yang telah di tentukan di awal .pada program saya kompleksitasnya nya adalah $200 \times (800 \times 800 \times 800) = 128.160.000$.

Untuk menentukan nilai K yang paling optimal diperlukan validasi dari data train ,lalu data train di bagi 2 pertama 35% untuk validasi dan bagian pertama 65% sebagai data train,data pertama berisi $35\% \times 800 = 280$ data yang dijadikan sebagai data validasi terhadap data train yang 65%.data validasi berisi 80 baris dari data train no urut 1-80,80 bari berikutnya dari data train no urut 301-380 ,sedangkan 80 baris terakhir dari 720-800

Akurasi



Series 1 Column1 Column2

Output - TuproAi3 (run) x

```

184 0.25324 3.546517 3.056972 0.779376 -0.490221 2.0
185 1.751636 1.538861 3.351011 1.155168 0.269289 1.0
186 0.878858 0.253358 -0.523204 1.477101 1.987675 2.0
187 0.84656 -0.722583 4.058765 0.694228 -0.896636 3.0
188 0.178394 3.477135 3.051401 0.43709 -0.50888 2.0
189 0.696275 4.374711 -1.514462 1.154267 3.349603 3.0
190 0.980117 2.742951 -1.308336 2.376409 2.686247 3.0
191 0.092755 -0.112141 -0.510224 0.253273 2.340795 2.0
192 0.996165 -0.550859 4.688104 0.863038 -0.47857 3.0
193 0.259038 2.164298 2.221758 0.040045 -1.643159 2.0
194 0.619725 -0.67492 2.058558 1.815732 -0.596283 3.0
195 2.53175 -1.164131 2.859301 1.468797 -1.041466 3.0
196 0.662597 2.447717 -3.110399 1.770591 1.580017 3.0
197 1.044456 -0.275026 3.882778 -0.722511 -0.201997 3.0
198 0.586493 3.969911 -0.094108 -0.280996 4.097631 3.0
199 0.222837 -1.731954 3.16993 0.81254 -1.894653 3.0
200 1.643372 2.683902 -0.676011 1.927427 3.051704 3.0
BUILD SUCCESSFUL (total time: 0 seconds)

```