

Viontina Dea Ivoni Y. P.

1301164643

IFIK-40-01

LAPORAN TUGAS 2 AI

K-NEAREST NEIGHBOR (K-NN)

Diberikan file Data Train ([2019] **DataTrain Tugas 2 AI.csv**) yang berisi 4000 data dengan 4 atribut (atribut 1, atribut 2, atribut 3, atribut 4) bilangan riil yang memiliki 2 kelas (0,1). Bangunlah sebuah sistem klasifikasi yang menerapkan algoritma k-Nearest Neighbor untuk mengklasifikasikan 1000 Data Test ([2019] **DataTest Tugas 2 AI.csv**).

Sistem yang dibangun harus mampu membaca masukan file Data Test dan mengeluarkan output berupa file berupa satu vektor berisi 1000 baris prediksi kelas 0/1 dari setiap data pada file Data Test (sesuai urutan).

Untuk running file **Tugas2_KNN_1301164643.py** diuji dengan $K = 5$. Setiap 1 data test dicari Euclidean Distance nya dengan 4000 data train. Langkah tersebut diulang sebanyak 1000 kali sampai semua data test dicari Euclidean Distance-nya dengan 4000 data train. Output dari running file tersebut adalah data berupa prediksi kelas yang disimpan pada file **Prediksi_Tugas2AI_[1301164643].csv**

Sedangkan untuk running file **Tugas2_FindBestK_1301164643.py** di bandingkan nilai K untuk menguji data train.

Desain Program

Untuk mencari class pada file data test :

1. Open file data train dan file data test.
2. Menentukan nilai K.
3. Lakukan looping untuk mencari Euclidean Distance antara data test dan data train.
4. Lakukan sorting pada hasil perhitungan Euclidean Distance.
5. Ambil data terdekat berdasarkan hasil sorting dan tentukan nilai prediksi class nya berdasarkan data yang diambil berdasarkan jumlah K.
6. Simpan data prediksi class pada file prediksi class.

Untuk mencari akurasi nilai K yang diuji dengan data train :

1. Open file data train dan file data test.
2. Menentukan nilai K, K yang dipakai disini yaitu dari 2 sampai 43 dengan selisih 3.
3. Lakukan split data train dengan nilai fold.
4. Lakukan looping untuk mencari Euclidean Distance antara data set dan data train.
5. Lakukan sorting pada hasil perhitungan Euclidean Distance.

6. Ambil data terdekat berdasarkan hasil sorting dan tentukan nilai prediksi class nya berdasarkan data yang diambil berdasarkan jumlah K.
7. Hitung akurasi dari setiap K yang diuji.
8. Tampilkan akurasinya.
9. Bandingkan nilai akurasinya, dan ambil yang paling besar nilai akurasinya sebagai nilai K terbaik.

Hasil Running Program **Tugas2_FindBestK_1301164643.py**

```
C:\Users\yonandaputri\Documents\AI\Tugas 2>python Tugas2_FindBestK_1301164643.py
Accuracy K = 2 : 0.5417500000000002
Accuracy K = 5 : 0.5417500000000002
Accuracy K = 8 : 0.5417500000000002
Accuracy K = 11 : 0.5417500000000002
Accuracy K = 14 : 0.5417500000000002
Accuracy K = 17 : 0.5417500000000002
Accuracy K = 20 : 0.5417500000000002
Accuracy K = 23 : 0.5417500000000002
Accuracy K = 26 : 0.5417500000000002
Accuracy K = 29 : 0.5417500000000002
Accuracy K = 32 : 0.5417500000000002
```

Menurut hasil running, akurasi nilai K sama semua. Jadi bisa dipilih nilai K mana saja. Namun saat running dari 3 sampai 41 dengan selisih 2 akan terdapat perbedaan akurasi.