

LAPORAN TUGAS PEMROGRAMAN 3
SISTEM CERDAS



Disusun oleh :

Daffa Satriarga - 1303202040

Gregorio B N Alvito - 1303202099

Muhammad Haikal- 1303202131

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
BANDUNG
2022

DAFTAR ISI

DAFTAR ISI	2
ABSTRAK	4
BAB I	
PENDAHULUAN	5
Latar Belakang	5
Metode KNN yang digunakan	5
Supremum	5
Minkowski	5
Observasi	5
Teknik/strategi pemrosesan/evaluasi data pada sheet data dengan melakukan 5-Fold Cross Validation	5
Strategi penggunaan algoritma kNN yang meliputi pemilihan ukuran jarak, nilai k dan cara mengambil keputusan	5
Library	5
BAB II	
PEMBAHASAN	6
Kode Program	6
Import Library pandas, numpy, sklearn. sebagai bahan utama dari perhitungan KNN	6
Membaca file dataset.xlsx, lalu mengambil sebagian kolom untuk dipisahkan data kolom label dan pixel	6
Setelah itu merubah data frame menjadi array	6
Membuat fungsi convert array menjadi 1 dimensi	6
Lalu membuat fungsi membagi data latih dan data uji menggunakan train_test_split.	7
Program fungsi normalisasi data menggunakan standar scaler	7
Membuat program penampungan nilai akurasi	7
Kalkulasi KNN	7
Kalkulasi KNN menggunakan metric Supremum	7
Membuat dataframe dengan tiap kolom yang berisi kalkulasi KNN Supremum	8
Menggabungkan data frame menjadi satu file excel	9
Output dari KNN Supremum	10
Hasil kalkulasi k = 1	10
Hasil kalkulasi k = 3	11
Hasil kalkulasi k = 5	12
Hasil kalkulasi k = 7	13
Hasil kalkulasi k = 9	14
Kesimpulan Supremum	14
Kalkulasi KNN menggunakan metric Minkowski	14

Membuat data frame dengan tiap kolom yang berisi kalkulasi KNN Minkowski	15
Menggabungkan data frame menjadi satu file excel	16
Output dari KNN Minkowski	17
Hasil kalkulasi $k = 1$	17
Hasil kalkulasi $k = 3$	18
Hasil kalkulasi $k = 5$	19
Hasil kalkulasi $k = 7$	20
Hasil kalkulasi $k = 9$	21
Kesimpulan Minkowski	21
BAB III	
KESIMPULAN	22
DAFTAR PUSTAKA	23

ABSTRAK

Terdapat dataset.xlsx yang berisi kasus *hand-writing digit recognizer*. Dari data tersebut digunakan suatu metode pembelajaran *K-Nearest Neighbor* (K-NN). *K-Nearest Neighbor* (KNN) adalah suatu metode yang menggunakan algoritma supervised dimana hasil dari query instance yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. KNN dibangun dalam bahasa pemrograman *python* dengan beberapa *library* yaitu *numpy*, *pandas* dan *sci-kit learn*. Perhitungan KNN dilakukan menggunakan 2 *metric* yaitu *euclidean* dan *minkowski* yang dilakukan secara iteratif berdasarkan k yang berbeda yaitu rentang 1 sampai dengan 5. kemudian dilakukan *5-Fold Cross Validation* sebagai metode validasi. Setelah kita melakukan kalkulasi dataset.xlsx menggunakan KNN dengan *metric* supremum dan *minkowski*, dan kita sudah memiliki 2 nilai akurasi tertinggi pada setiap *metric*nya. Terdapat nilai akurasi 0,535 untuk *metric* Supremum dan terdapat nilai akurasi 0,8125 untuk *metric* Minkowski. dan dapat kita simpulkan bahwa *metric* Minkowski lebih baik daripada *metric* Supremum

Kata kunci : K-Nearest Neighbor, Minkowski, Supremum

BAB I

PENDAHULUAN

1. Latar Belakang

Terdapat dataset.xlsx yang berisi kasus *hand-writing digit recognizer*. Data tersebut merepresentasikan gambar *grayscale* dari tulisan tangan dengan dimensi 28 x 28 pixel (total 784 pixel). Setiap pixel dapat berisi nilai 0-255 yang merepresentasikan derajat keabuan dari pixel, semakin tinggi semakin gelap. Sheet data terdiri atas 1501 baris dan 786 kolom, dimana baris pertama merupakan label attribute yaitu idData, label-kelas dan idPixel. Sheet Submit terdiri atas 786 kolom dengan kolom pertama adalah idData, kolom kedua adalah label kelas dari data 0-9 dan kolom lainnya merepresentasikan nilai pixel dengan rentang 0-255 (dalam 1 vektor). Dari data tersebut digunakan suatu metode pembelajaran *K-Nearest Neighbor* (KNN).

K-nearest neighbor (KNN) merupakan salah satu algoritma yang digunakan dalam machine learning. Jika kamu ingin mempelajari machine learning, kamu perlu memahami bagaimana algoritma ini bekerja, berikut contoh aplikasi dan kelebihan dan kekurangannya.

1.1. Metode KNN yang digunakan

- Supremum
- Minkowski

1.2. Observasi

Berikut adalah hal-hal yang diobservasi dalam pemecahan masalah.

- Teknik/strategi pemrosesan/evaluasi data pada *sheet* data dengan melakukan *5-Fold Cross Validation*
- Strategi penggunaan algoritma kNN yang meliputi pemilihan ukuran jarak, nilai k dan cara mengambil keputusan

1.3. Library

- *Pandas*
- *Numpy*
- *Sklearn*

BAB II PEMBAHASAN

2. Kode Program

2.1. Import Library pandas, numpy, sklearn. sebagai bahan utama dari perhitungan KNN

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
import numpy as np
```

2.2. Membaca file dataset.xlsx, lalu mengambil sebagian kolom untuk dipisahkan data kolom label dan pixel

```
#mengambil sebagian kolom untuk dipisahkan data kolom label dan pixel
def input_excel():
    file_excel = pd.read_excel('dataset.xlsx')
    label = file_excel[file_excel.columns[2:786]]
    pixel = file_excel[file_excel.columns[1]]
    return label, pixel
```

2.3. Setelah itu merubah data frame menjadi array

```
#mengubah dataframe menjadi array
def ubah_array(kolom):
    kolom = np.array(kolom)
    return kolom
```

2.4. Membuat fungsi convert array menjadi 1 dimensi

```
#convert array 1 dimensi
def flat_array(flatted):
```

```
flatted = flatted.flatten()
return flatted
```

- 2.5. Lalu membuat fungsi membagi data latih dan data uji menggunakan `train_test_split`.

```
def split_data(label, pixel):
    X_train, X_test, y_train, y_test = train_test_split(label, pixel, test_size=0.2,
random_state=35)
    return X_train, X_test, y_train, y_test
```

- 2.6. Program fungsi normalisasi data menggunakan standar scaler

```
def normalisasi(X_train, X_test):
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)
    return X_train, X_test
```

- 2.7. Membuat program penampungan nilai akurasi

```
#list accuracy_arr
accuracy_arr = []
```

3. Kalkulasi KNN

- 3.1. Kalkulasi KNN menggunakan metric Supremum

```
#perhitungan KNN (1, 3, 5, 7, 9) dengan 5 Fold Cross Validation
for i in range(1,10,2):
    knn = KNeighborsClassifier(n_neighbors=i, metric='chebyshev').fit(X_train,
y_train)
    y_pred = knn.predict(X_test)
    result = cross_val_score(knn, X_train, y_train, cv=5, scoring='accuracy')
    accuracy_arr.append(result.mean())
```

3.1.1.1. Membuat dataframe dengan tiap kolom yang berisi kalkulasi KNN Supremum

```
#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN  
k=1  
data1 = pd.DataFrame({  
    'idData' : np.arange(1, len(y_test) + 1),  
    'label aktual' : y_test,  
    'prediksi' : y_pred,  
    'label luaran' : np.NaN  
})  
data1.at[0,'label luaran'] = accuracy_arr[0]  
  
#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN  
k=3  
data2 = pd.DataFrame({  
    'idData' : np.arange(1, len(y_test) + 1),  
    'label aktual' : y_test,  
    'prediksi' : y_pred,  
    'label luaran' : np.NaN  
})  
data2.at[0,'label luaran'] = accuracy_arr[1]  
  
#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN  
k=5  
data3 = pd.DataFrame({  
    'idData' : np.arange(1, len(y_test) + 1),  
    'label aktual' : y_test,  
    'prediksi' : y_pred,  
    'label luaran' : np.NaN  
})  
data3.at[0,'label luaran'] = accuracy_arr[2]
```



```

#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN
k=7
data4 = pd.DataFrame({
    'idData' : np.arange(1, len(y_test) + 1),
    'label aktual' : y_test,
    'prediksi' : y_pred,
    'label luaran' : np.NaN
})
data4.at[0,'label luaran'] = accuracy_arr[3]

#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN
k=9
data5 = pd.DataFrame({
    'idData' : np.arange(1, len(y_test) + 1),
    'label aktual' : y_test,
    'prediksi' : y_pred,
    'label luaran' : np.NaN
})
data5.at[0,'label luaran'] = accuracy_arr[4]

```

3.1.2. Menggabungkan data frame menjadi satu file excel

```

with pd.ExcelWriter(r"Supremum\OutputValidasi.xlsx") as hasil:
    data1.to_excel(hasil, sheet_name="k=1", index=False)
    data2.to_excel(hasil, sheet_name="k=3", index=False)
    data3.to_excel(hasil, sheet_name="k=5", index=False)
    data4.to_excel(hasil, sheet_name="k=7", index=False)
    data5.to_excel(hasil, sheet_name="k=9", index=False)

```

3.1.3. Output dari KNN Supremum

3.1.3.1. Hasil kalkulasi k = 1

	A	B	C	D
1	idData	label aktua	prediksi	label luaran
2	1	0	3	0.535
3	2	8	0	
4	3	5	9	
5	4	8	4	
6	5	9	7	
7	6	2	2	
8	7	4	2	
9	8	2	2	
10	9	9	9	
11	10	6	6	
12	11	4	0	
13	12	2	3	
14	13	4	7	
15	14	2	2	
16	15	9	9	
17	16	2	2	
18	17	3	3	
19	18	4	0	
20	19	2	0	
21	20	2	1	
22	21	8	1	
23	22	5	7	
24	23	2	2	
25	24	6	6	
26	25	5	4	
27	26	7	7	
28	27	6	1	
29	28	4	0	
30	29	2	6	
31	30	3	3	
32	31	9	9	
33	32	0	0	
< > k=1 k=3 k=5 k=7 k=9				

3.1.3.2. Hasil kalkulasi k = 3

	A	B	C	D	
1	idData	abel aktua	prediksi	abel luaran	
2	1	0	3	0.506667	
3	2	8	0		
4	3	5	9		
5	4	8	4		
6	5	9	7		
7	6	2	2		
8	7	4	2		
9	8	2	2		
10	9	9	9		
11	10	6	6		
12	11	4	0		
13	12	2	3		
14	13	4	7		
15	14	2	2		
16	15	9	9		
17	16	2	2		
18	17	3	3		
19	18	4	0		
20	19	2	0		
21	20	2	1		
22	21	8	1		
23	22	5	7		
24	23	2	2		
25	24	6	6		
26	25	5	4		
27	26	7	7		
28	27	6	1		
29	28	4	0		
30	29	2	6		
31	30	3	3		
32	31	9	9		
33	32	0	0		
<div> <div><</div> <div>></div> <div>k=1</div> <div>k=3</div> <div>k=5</div> <div>k=7</div> <div>k=9</div> </div>					

3.1.3.3. Hasil kalkulasi k = 5

	A	B	C	D
1	idData	abel aktua	prediksi	abel luaran
2	1	0	3	0.524167
3	2	8	0	
4	3	5	9	
5	4	8	4	
6	5	9	7	
7	6	2	2	
8	7	4	2	
9	8	2	2	
10	9	9	9	
11	10	6	6	
12	11	4	0	
13	12	2	3	
14	13	4	7	
15	14	2	2	
16	15	9	9	
17	16	2	2	
18	17	3	3	
19	18	4	0	
20	19	2	0	
21	20	2	1	
22	21	8	1	
23	22	5	7	
24	23	2	2	
25	24	6	6	
26	25	5	4	
27	26	7	7	
28	27	6	1	
29	28	4	0	
30	29	2	6	
31	30	3	3	
32	31	9	9	
33				

< > | k=1 | k=3 | **k=5** | k=7 | k=9 |

3.1.3.4. Hasil kalkulasi k = 7

	A	B	C	D
1	idData	label aktua	prediksi	label luaran
2	1	0	3	0.529167
3	2	8	0	
4	3	5	9	
5	4	8	4	
6	5	9	7	
7	6	2	2	
8	7	4	2	
9	8	2	2	
10	9	9	9	
11	10	6	6	
12	11	4	0	
13	12	2	3	
14	13	4	7	
15	14	2	2	
16	15	9	9	
17	16	2	2	
18	17	3	3	
19	18	4	0	
20	19	2	0	
21	20	2	1	
22	21	8	1	
23	22	5	7	
24	23	2	2	
25	24	6	6	
26	25	5	4	
27	26	7	7	
28	27	6	1	
29	28	4	0	
30	29	2	6	
31	30	3	3	
32	31	9	9	
33	32	0	0	

< >
k=1 k=3 k=5 k=7 k=9

3.1.3.5. Hasil kalkulasi k = 9

	A	B	C	D
1	idData	abel aktua	prediksi	abel luaran
2	1	0	3	0.516667
3	2	8	0	
4	3	5	9	
5	4	8	4	
6	5	9	7	
7	6	2	2	
8	7	4	2	
9	8	2	2	
10	9	9	9	
11	10	6	6	
12	11	4	0	
13	12	2	3	
14	13	4	7	
15	14	2	2	
16	15	9	9	
17	16	2	2	
18	17	3	3	
19	18	4	0	
20	19	2	0	
21	20	2	1	
22	21	8	1	
23	22	5	7	
24	23	2	2	
25	24	6	6	
26	25	5	4	
27	26	7	7	
28	27	6	1	
29	28	4	0	
30	29	2	6	
31	30	3	3	
32	31	9	9	
33	32	0	0	

3.1.4. Kesimpulan Supremum

Melihat hasil kalkulasi di atas, bisa disimpulkan bahwa k = 1 dengan nilai 0,535 adalah nilai akurasi tertinggi dibandingkan dengan nilai k lainnya.

3.2. Kalkulasi KNN menggunakan metric Minkowski

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
for i in range(1,10,2):
    knn = KNeighborsClassifier(n_neighbors=i, metric='minkowski').fit(X_train,
y_train)
    y_pred = knn.predict(X_test)
    result = cross_val_score(knn, X_train, y_train, cv=5, scoring='accuracy')
    accuracy_arr.append(result.mean())
```

- 3.2.1. Membuat data frame dengan tiap kolom yang berisi kalkulasi KNN Minkowski

```
#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN
k=1
data1 = pd.DataFrame({
    'idData' : np.arange(1, len(y_test) + 1),
    'label aktual' : y_test,
    'prediksi' : y_pred,
    'label luaran' : np.NaN
})
data1.at[0,'label luaran'] = accuracy_arr[5]

#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN
k=3
data2 = pd.DataFrame({
    'idData' : np.arange(1, len(y_test) + 1),
    'label aktual' : y_test,
    'prediksi' : y_pred,
    'label luaran' : np.NaN
})
data2.at[0,'label luaran'] = accuracy_arr[6]

#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN
k=5
data3 = pd.DataFrame({
    'idData' : np.arange(1, len(y_test) + 1),
    'label aktual' : y_test,
    'prediksi' : y_pred,
    'label luaran' : np.NaN
})
data3.at[0,'label luaran'] = accuracy_arr[7]

#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN
k=7
data4 = pd.DataFrame({
```

```

'idData' : np.arange(1, len(y_test) + 1),
'label aktual' : y_test,
'prediksi' : y_pred,
'label luaran' : np.NaN
})
data4.at[0,'label luaran'] = accuracy_arr[8]

#Membuat dataframe dengan tiap kolom yang berisi hasil kalkulasi KNN
k=9
data5 = pd.DataFrame({
'idData' : np.arange(1, len(y_test) + 1),
'label aktual' : y_test,
'prediksi' : y_pred,
'label luaran' : np.NaN
})
data5.at[0,'label luaran'] = accuracy_arr[9]

```

3.2.2. Menggabungkan data frame menjadi satu file excel

```

with pd.ExcelWriter(r"Minkowski\OutputValidasi.xlsx") as hasil:
    data1.to_excel(hasil, sheet_name="k=1", index=False)
    data2.to_excel(hasil, sheet_name="k=3", index=False)
    data3.to_excel(hasil, sheet_name="k=5", index=False)
    data4.to_excel(hasil, sheet_name="k=7", index=False)
    data5.to_excel(hasil, sheet_name="k=9", index=False)

```


3.2.3. Output dari KNN Minkowski

3.2.3.1. Hasil kalkulasi k = 1

	A	B	C	D
1	idData	label aktua	prediksi	label luaran
2	1	0	0	0.8125
3	2	8	8	
4	3	5	5	
5	4	8	8	
6	5	9	9	
7	6	2	2	
8	7	4	4	
9	8	2	2	
10	9	9	9	
11	10	6	6	
12	11	4	4	
13	12	2	3	
14	13	4	4	
15	14	2	8	
16	15	9	9	
17	16	2	1	
18	17	3	3	
19	18	4	4	
20	19	2	1	
21	20	2	2	
22	21	8	8	
23	22	5	5	
24	23	2	3	
25	24	6	6	
26	25	5	8	
27	26	7	7	
28	27	6	6	
29	28	4	4	
30	29	2	1	
31	30	3	3	
32	31	9	9	
33	32	0	0	
< > k=1 k=3 k=5 k=7 k=9				

3.2.3.2. Hasil kalkulasi k = 3

	A	B	C	D	
1	idData	abel aktua	prediksi	abel luaran	
2	1	0	0	0.81	
3	2	8	8		
4	3	5	5		
5	4	8	8		
6	5	9	9		
7	6	2	2		
8	7	4	4		
9	8	2	2		
10	9	9	9		
11	10	6	6		
12	11	4	4		
13	12	2	3		
14	13	4	4		
15	14	2	8		
16	15	9	9		
17	16	2	1		
18	17	3	3		
19	18	4	4		
20	19	2	1		
21	20	2	2		
22	21	8	8		
23	22	5	5		
24	23	2	3		
25	24	6	6		
26	25	5	8		
27	26	7	7		
28	27	6	6		
29	28	4	4		
30	29	2	1		
31	30	3	3		
32	31	9	9		
33	32	0	0		
<div> <div><</div> <div>></div> <div>k=1</div> <div>k=3</div> <div>k=5</div> <div>k=7</div> <div>k=9</div> </div>					

3.2.3.3. Hasil kalkulasi k = 5

	A	B	C	D
1	idData	label aktual	prediksi	label luaran
2	1	0	0	0.808333
3	2	8	8	
4	3	5	5	
5	4	8	8	
6	5	9	9	
7	6	2	2	
8	7	4	4	
9	8	2	2	
10	9	9	9	
11	10	6	6	
12	11	4	4	
13	12	2	3	
14	13	4	4	
15	14	2	8	
16	15	9	9	
17	16	2	1	
18	17	3	3	
19	18	4	4	
20	19	2	1	
21	20	2	2	
22	21	8	8	
23	22	5	5	
24	23	2	3	
25	24	6	6	
26	25	5	8	
27	26	7	7	
28	27	6	6	
29	28	4	4	
30	29	2	1	
31	30	3	3	
32	31	9	9	
33				

3.2.3.4. Hasil kalkulasi k = 7

	A	B	C	D
1	idData	label aktua	prediksi	label luaran
2	1	0	0	0.790833
3	2	8	8	
4	3	5	5	
5	4	8	8	
6	5	9	9	
7	6	2	2	
8	7	4	4	
9	8	2	2	
10	9	9	9	
11	10	6	6	
12	11	4	4	
13	12	2	3	
14	13	4	4	
15	14	2	8	
16	15	9	9	
17	16	2	1	
18	17	3	3	
19	18	4	4	
20	19	2	1	
21	20	2	2	
22	21	8	8	
23	22	5	5	
24	23	2	3	
25	24	6	6	
26	25	5	8	
27	26	7	7	
28	27	6	6	
29	28	4	4	
30	29	2	1	
31	30	3	3	
32	31	9	9	
33	32	0	0	

< > | k=1 | k=3 | k=5 | **k=7** | k=9 |

3.2.3.5. Hasil kalkulasi k = 9

	A	B	C	D		
1	idData	label aktual	prediksi	label luaran		
2	1	0	0	0.794167		
3	2	8	8			
4	3	5	5			
5	4	8	8			
6	5	9	9			
7	6	2	2			
8	7	4	4			
9	8	2	2			
10	9	9	9			
11	10	6	6			
12	11	4	4			
13	12	2	3			
14	13	4	4			
15	14	2	8			
16	15	9	9			
17	16	2	1			
18	17	3	3			
19	18	4	4			
20	19	2	1			
21	20	2	2			
22	21	8	8			
23	22	5	5			
24	23	2	3			
25	24	6	6			
26	25	5	8			
27	26	7	7			
28	27	6	6			
29	28	4	4			
30	29	2	1			
31	30	3	3			
32	31	9	9			
33	32	2	2			
	<	k=1	k=3	k=5	k=7	k=9

3.2.4. Kesimpulan Minkowski

Berdasarkan hasil kalkulasi di atas, bisa disimpulkan bahwa k = 1 dengan nilai 0,8125 adalah nilai akurasi tertinggi dibanding nilai k yang lainnya.

BAB III

KESIMPULAN

Setelah kita melakukan kalkulasi dataset.xlsx menggunakan KNN dengan metric supremum dan minkowski, dan kita sudah memiliki 2 nilai akurasi tertinggi pada setiap metricnya. Terdapat nilai akurasi 0,535 untuk metric Supremum dan terdapat nilai akurasi 0,8125 untuk metric Minkowski. dan dapat kita simpulkan bahwa metric Minkowski lebih baik daripada metric Supremum

DAFTAR PUSTAKA

- Lindawati. Data Mining dengan teknik Clustering dalam pengklasifikasian data mahasiswa studi kasus prediksi lama studi mahasiswa Universitas Bina Nusantara. 2008.
- Kesumadewi, Sri. Klasifikasi Status Gizi Menggunakan Naive Bayes Classification. 2009.
- Kumalasari, Noviana Ayu; Marji, Candra Dewi. Implementasi algoritma Modified k-Nearest Neighbor untuk menentukan tingkat resiko penyakit lemak darah. 2014.