

LAPORAN OBSERVASI

Tugas Pemrograman 03

CTI-2G3 Sistem Cerdas



Oleh:

Alfan Cahyo Wicaksono (1303184065)

Deni Saputra Hermawan (1303180074)

Mohammad Rizqi Akmaludin (1303184031)

PROGRAM STUDI S1 TEKNOLOGI INFORMASI

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

2021/2022

Hal yang dapat diobservasi/dipaparkan:

1. Deskripsi studi kasus yang minimal berisikan tujuan, deskripsi masukan dan deskripsi luaran dari penerapan kNN pada kasus yang diambil.

1.1. Tujuan

Tujuan yang akan dihasilkan pada kode program yang ada adalah untuk mendapatkan nilai akurasi dan hasil prediksi berupa klasifikasi terhadap id data.

1.2. Deskripsi Masukan

Masukan atau input yang digunakan pada program ini berupa 2 sheet dalam 1 file input yakni Data dan Submit. Untuk sheet Data terdiri atas iddata, label dan pixel. Untuk sheet Submit terdiri atas iddata dan pixel.

1.3. Deskripsi Luaran

Luaran atau output yang dihasilkan pada penerapan Knn adalah 2 file output yakni file outputLatih sebagai hasil dari inputan sheet Data dan output dengan file file outputSubmit sebagai hasil dari inputan sheet Submit. Dimana pada file outputLatih memiliki atribut iddata, Klasifikasi dan akurasi sementara file outputSubmit hanya akan memiliki atribut iddata dan klasifikasi.

2. Membaca Data

Kami menggunakan pandas read excel dengan 2 dataframe yang menampung masing masing sheet.

a. Read sheet Data

```
# Input file dan memasukkan kedalam variabel
df_Data = pd.read_excel('DataSetTB3_SHARE.xlsx', sheet_name="Data")
X = df_Data.iloc[:, 2:]
Y = df_Data['label']
```

Variabel X untuk menyimpan colom seluruh pixel. Variabel Y menyimpan colom label.

b. Read sheet Submit

```
# Input sheet submit
df_submit = pd.read_excel('DataSetTB3_SHARE.xlsx', sheet_name="Submit")

x_submit = df_submit.iloc[:, 1:]
```

Variabel x_submit digunakan untuk menyimpan colom seluruh pixel.

3. Split Data untuk Dataset Data

Kami menggunakan library `train_test_split` untuk membagi dataset Data sesuai permintaan soal yakni 70% train dan 30% test.

```
# split isi dari sheet Data menjadi 70 : 30
X_train, X_test, y_train, y_test = train_test_split(
    X, Y, test_size=0.3, random_state=42)
```

4. Eksperimen KNN dengan mencoba metrics dan k yang berbeda

Kami melakukan percobaan untuk menyari nilai terbaik dari akurasi dengan beberapa metrics dan k, didapat hasil sebagai berikut.

```
PS D:\PROGRAMMING\Github\Tupro-Sistem-Cerdas\Tupro3_knn> python -u "d:\PROGRAMMING\Github\Tupro-Sistem-Cerdas\Tupro3_knn\tupro3_knn.py"
Akurasi euclidean dengan k = 5 adalah 0.8333333333333334
Akurasi euclidean dengan k = 8 adalah 0.8033333333333333
Akurasi manhattan dengan k = 5 adalah 0.81
Akurasi manhattan dengan k = 8 adalah 0.7966666666666667
Akurasi minkowski dengan k = 5 adalah 0.8333333333333334
Akurasi minkowski dengan k = 8 adalah 0.8033333333333333
Akurasi chebyshev dengan k = 5 adalah 0.4566666666666667
Akurasi chebyshev dengan k = 8 adalah 0.4566666666666667
```

5. Teknik KNN Classifier

Berdasarkan hasil nomor diatas, kami memutuskan untuk menggunakan Metric Euclidean dan panjang atau nilai $K = 5$. Berikut rumus yang kami gunakan untuk melakukan fit dan prediksi.

```
# Analisis Data Latih dan Penetapan KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5, weights='uniform',
                           algorithm='auto', leaf_size=30, metric='euclidean')
knn.fit(X_train, y_train)
```

6. Prediksi untuk menghasilkan klasifikasi pada output

Kami menggunakan library `predict` pada `knn` untuk mendapatkan nilai klasifikasi.

a. Prediksi pada sheet Data

```
y_pred_knn = knn.predict(X_test)
```

b. Prediksi pada sheet Submit

Untuk fit sendiri, kami menggunakan parameter `X` dan `Y`. Karena pada submit itu tidak memiliki label, sehingga untuk melakukan `predict` perlu ada perbandingan, maka solusi yang diberikan adalah menggunakan semua data pada sheet data. (Referensi bu Azka)

```
# Analisis Data Submit
knn.fit(X, Y)
y_pred_knn_submit = knn.predict(X_submit)
print(y_pred_knn_submit)
```

7. Membuat file output

Berikut adalah code dan ss hasil output tiap data :

a. OutputLatih.xlsx

```
## Membuat file excel "OutputLatih.xlsx"
df1 = pd.DataFrame(X_test, columns=['idData'])
numpy_array = np.array(y_pred_knn)
print(df1.index)
workbook = xlswriter.Workbook('OutputLatih.xlsx')
worksheet = workbook.add_worksheet("Data")
worksheet.write(0, 0, 'idData')
worksheet.write(0, 1, 'Klasifikasi')
worksheet.write('C1', 'Akurasi')
worksheet.write('C2', accuracy_score(y_test, y_pred_knn))
start = 1
for i in range(300):
    worksheet.write(start, 0, df1.index[i])
    worksheet.write(start, 1, numpy_array[i])
    start += 1
workbook.close()
```

AutoSave

Off

OutputLatih ▾

File

Home

Insert

Page Layout

Formulas

Data

Review

Paste ▾

Cut

Copy ▾

Format Painter

Clipboard

Calibri ▾

11 ▾

A[^]

A^v

B

I

U ▾

▾

▾

▾

▾

▾

▾






Font

K12 ▾




	A	B	C	D	E	F	G
1	idData	Klasifikasi	Akurasi				
2	521	5	0.833333				
3	737	7					
4	740	4					
5	660	6					
6	411	4					
7	678	6					
8	626	6					
9	513	5					
10	859	1					
11	136	1					
12	811	9					
13	76	0					
14	636	6					
15	973	9					
16	938	9					
17	899	8					

- Karena menggunakan train split yang membagi 30% test maka hasil yang didapat berupa 300 row data.
- b. OutputSubmit.xlsx








```
# Membuat file excel "OutputSubmit.xlsx"
df2 = pd.DataFrame(X_submit, columns=['idData'])
numpy_array_submit = np.array(y_pred_knn_submit)
print(df1.index)
workbook = xlswriter.Workbook('OutputSubmit.xlsx')
worksheet = workbook.add_worksheet("Submit")
worksheet.write(0, 0, 'idData')
worksheet.write(0, 1, 'Klasifikasi')
start = 1
for i in range(500):
    worksheet.write(start, 0, df2.index[i]+1)
    worksheet.write(start, 1, numpy_array_submit[i])
    start += 1
workbook.close()
```




AutoSave ☐ Off     OutputSubmit 

File **Home** Insert Page Layout Formulas Data Review





Paste  Cut  Copy  Format Painter

Clipboard

Calibri 11       

B *I* U   

Font

L11    

	A	B	C	D	E	F	G
1	idData	Klasifikasi					
2	1	0					
3	2	0					
4	3	0					
5	4	0					
6	5	0					
7	6	0					
8	7	0					
9	8	0					
10	9	0					
11	10	0					
12	11	0					
13	12	0					
14	13	0					
15	14	0					
16	15	0					
17	16	0					
18	17	0					

Untuk outputSubmit karena tidak ada split data maka yang digunakan adalah seluruh data yakni 500 row.

8. Kesimpulan dan Hasil Analisis

Berdasarkan percobaan yang dilakukan, kami menemukan bahwa metric yang terbaik adalah euclidean dengan nilai K adalah 5. Hasil tersebut telah diuji coba pada data train dan test pada sheet Data. Dimana saat menggunakan hal tersebut pada sheet Submit didapatkanlah hasil range sampai 5.

Link Github : <https://github.com/AwanSaputra/Tupro-Sistem-Cerdas>

Link Video :

<https://drive.google.com/file/d/1qisBJGtdi0WvYKWaJHjk5BsCx8CBJubq/view?usp=sharing>

Referensi :

1. Slide RESPONSI TUPRO 3 SC IT
2. <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>
3. <https://towardsdatascience.com/knn-in-python-835643e2fb53>
4. Responsi private bersama Bu Azka