

Tugas Besar 1 Mathematical Tools For Data Science
Resume Model Classifier



Disusun oleh :
Muhamad Irvan Dandung
(41519110136)

Dosen Pengampu :
Ida Nurhaida, Dr,MT.

Universitas Mercu Buana – Menteng
Jl. Menteng Raya No.29, RT.1/RW.10, Kb. Sirih, Kec. Menteng, Kota Jakarta
Pusat, Daerah Khusus Ibukota Jakarta 10340.

BAB I

PENDAHULUAN

Clustering adalah salah satu proses data mining yang bertujuan untuk mempartisi data yang ada menjadi satu atau lebih objek cluster berdasarkan karakteristik yang dimilikinya. Data dengan karakteristik yang sama dikelompokkan dalam satu cluster dan data dengan karakteristik yang berbeda dikelompokkan ke dalam cluster yang lain. Pada penelitian ini akan dilakukan analisis algoritma Clustering KNN untuk mengkategorikan bunga dengan menggunakan dataset iris. Dalam penelitian ini akan dilakukan proses penentuan jenis bunga iris, dalam penentuan bunga iris tersebut digunakan 4 faktor yang merupakan karakteristik sendiri dari bunga iris tersebut. Keempat faktor karakteristik bunga iris yang digunakan untuk penelitian yaitu Id, sepal length, sepal width, petal length, petal width dan species. Sehingga dengan adanya banyak faktor yang diteliti dan record data yang ada dalam penentuan jenis bunga iris, maka diperlukan sebuah metode yang dapat digunakan untuk menghasilkan secara tepat dan akurat dalam menentukan jenis bunga pada umumnya, termasuk penentuan jenis bunga iris pada khususnya. Sudah banyak algoritma clustering yang dipakai dalam penelitian ini tapi kita lebih memfokuskan menggunakan algoritma clustering KNN.

1. Tujuan Penelitian

Tujuan adanya penelitian atau pengerjaan tugas besar 1 ini yaitu:

1. Mengklasifikasi data bunga Iris ini untuk menghitung variasi morfologis bunga iris dari tiga spesies yang terkait, yakni Iris Sentosa, Iris Virginica, dan Iris Versicolor.
2. Menerapkan metode algoritma K-Nearest Neighbor (KNN) untuk mengatasi permasalahan di atas.

2. Metode Penelitian

Adapun metode yang digunakan untuk mengklasifikasi data bunga Iris ini untuk menghitung variasi morfologis bunga iris dari tiga spesies yang terkait, yakni Iris Sentosa, Iris Virginica, dan Iris Versicolor. Algoritma K-Nearest Neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek yang berdasarkan dari data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN merupakan algoritma supervised learning dimana hasil dari query instance yang baru diklasifikasi berdasarkan mayoritas dari kategori pada algoritma KNN. Dimana kelas yang paling banyak muncul yang nantinya akan menjadi kelas hasil dari klasifikasi.

Algoritma metode k-NN sangatlah sederhana, bekerja dengan berdasarkan pada jarak terdekat dari sampel uji ke sample latih untuk menentukan k-NN nya. Setelah mengumpulkan k-NN, kemudian diambil mayoritas dari k-NN untuk dijadikan prediksi dari sampel uji. Data untuk algoritma k-NN terdiri dari beberapa atribut multivariat X_i yang akan digunakan untuk mengklasifikasikan Y . Data dari k-NN dapat dalam skala ukuran apapun, dari ordinal ke nominal.

Algoritma metode k-NN sangatlah sederhana, bekerja dengan berdasarkan pada jarak terpendek dari sampel uji ke sample latih untuk menentukan k-NN nya. Setelah mengumpulkan k-NN, kemudian diambil mayoritas dari k-NN untuk dijadikan prediksi dari sampel uji. Data untuk algoritma k-NN terdiri dari beberapa atribut multivariat X_i yang akan digunakan untuk mengklasifikasikan Y . Data dari k-NN dapat dalam skala ukuran apapun, dari ordinal ke nominal.

Kelebihan algoritma KNN yaitu:

- Sangat Non-linear.
- Mudah dipahami dan diimplementasikan.
- Tangguh terhadap data training sample yang noisy.
- Efektif apabila data training sample-nya besar.
- Memiliki konsistensi yang kuat.
- Asymptotically correct.

Kekurangan algoritma KNN yaitu:

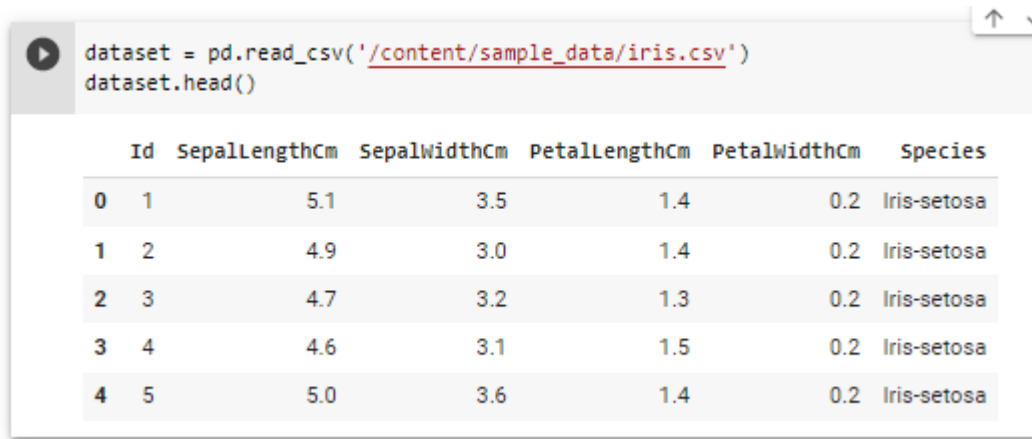
- Perlu Menentukan Parameter k (Jumlah Tetangga Terdekat).
- Tidak Menangani Nilai Hilang (Missing Value) Secara Implisit.
- Sensitif Terhadap Data Pencilan (Outlier).
- Rentan Terhadap Variabel Yang Non-Informatif.
- Rentan Terhadap Dimensionalitas Yang Tinggi.
- Rentan Terhadap Perbedaan Rentang Variabel.
- Pembelajaran Berdasarkan Jarak Tidak Jelas.
- Nilai Komputasi yang Tinggi.

BAB II

HASIL & PEMBAHASAN

1. Pembacaan Dataset dan Cleansing Data

Dalam pengambilan data kita menggunakan dataset yang telah disediakan di dalam file iris.csv. Pada file tersebut memiliki 150 row dataset dengan 6 column yaitu Id, Sepal Length Cm, Sepal Width Cm, Petal Length Cm, Petal Width Cm, Species.



```
dataset = pd.read_csv('/content/sample_data/iris.csv')
dataset.head()
```

	Id	SepallengthCm	SepalwidthCm	PetalLengthCm	PetalwidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Setelah berhasil melakukan pembacaan data, lalu langkah selanjutnya menghapus data yang tidak diperlukan. Untuk melakukan pengklasifikasian data di atas itu hanya diperlukan dataset dari 5 column yang ada yaitu Sepal Length Cm, Sepal Width Cm, Petal Length Cm, Petal Width Cm, dan Spesies dimana data species nanti akan digunakan sebagai data label klasifikasi dan data Sepal Length Cm, Sepal Width Cm, Petal Length Cm, Petal Width Cm adalah data yang digunakan sebagai data train atau pengklasifikasian.

```
[ ] dataset.drop(['Id'], axis=1, inplace=True)
dataset
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

2. Merubah Dataset Menjadi Array List

Supaya dataset di atas dapat diolah maka data di atas harus diubah dari data frame menjadi array list dimana data dari column Sepal Length Cm, Sepal Width Cm, Petal Length Cm, Petal Width Cm akan digunakan sebagai data kategori X dan data dari column Species akan digunakan sebagai kategori y.

```
[ ] x = dataset.iloc[:, :-1].values
x[0:5]
```

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2]])
```

```
[ ] y = dataset.iloc[:, 4].values
y[0:5]
```

```
array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa'], dtype=object)
```

3. Menyiapkan Data Train dan Data Test

Setelah kita memisahkan kategori X dan Y lalu data tersebut akan dijadikan argumen / input ke dalam function train_test_split() untuk menghasilkan data train dan data set.

- Train Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2, random_state=4)
print ('Train set:', X_train.shape, y_train.shape)
print (('Test set:', X_test.shape, y_test.shape))
```

Train set: (120, 4) (120,)
Test set: (30, 4) (30,)

Dari proses ini dihasilkan nilai train set yaitu: (120, 4) (120,) dan test set yaitu: (30, 4) (30,).

4. Proses Klasifikasi Data yang Sudah Diolah Menggunakan KNN

Sebagai langkah pertama kita akan menentukan nilai k yaitu dengan nilai 5 karena kita belum mengetahui berada dimana / angka berapa nilai k yang cocok. Setelah itu siapkan training modelnya.

```
k = 4
#Train Model
neigh = KNeighborsClassifier(n_neighbors = k).fit(X_train,y_train)
neigh
```

KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
metric_params=None, n_jobs=None, n_neighbors=4, p=2,
weights='uniform')

5. Melakukan Prediksi Train Model.

Setelah model dibuat langkah selanjutnya yaitu melakukan prediksi pada data set test.

```
[14] yhat = neigh.predict(X_test)
yhat[0:5]
```

array(['Iris-virginica', 'Iris-setosa', 'Iris-virginica',
 'Iris-virginica', 'Iris-virginica'], dtype=object)

6. Menghitung Akurasi Subset

```
[15] from sklearn import metrics
print("Train set Accuracy: ", metrics.accuracy_score(y_train, neigh.predict(X_train)))
print("Test set Accuracy: ", metrics.accuracy_score(y_test, yhat))
```

Train set Accuracy: 0.975
Test set Accuracy: 0.9666666666666667

Dari inputan data-data yang telah diinput. Nilai akurasi train set ada di 0.975 dan nilai akurasi test set ada di 0.966 yang mana nilai keduanya hampir mendekati 1 dan itu merupakan score yang cukup tinggi dan baik untuk training pertama.

7. Training Model Sebanyak K yang Ingin Diuji

```
✓ [38] Ks = 11
0d mean_acc = np.zeros((Ks-1))
std_acc = np.zeros((Ks-1))
ConfusionMx = [];
for n in range(1,Ks):

    #Train Model and Predict
    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train,y_train)
    yhat=neigh.predict(X_test)
    mean_acc[n-1] = metrics.accuracy_score(y_test, yhat)
    std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

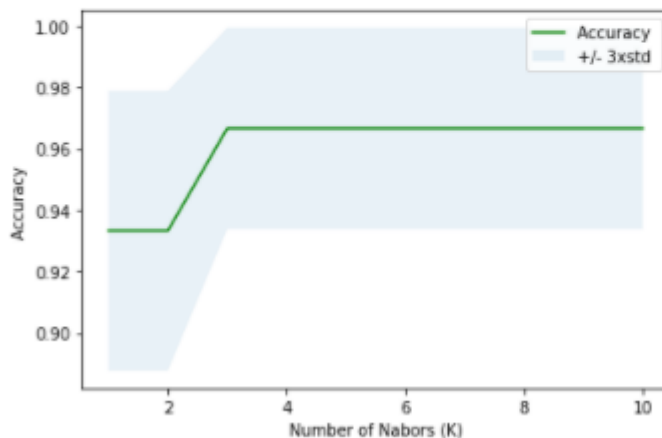
mean_acc

array([0.93333333, 0.93333333, 0.96666667, 0.96666667, 0.96666667,
       0.96666667, 0.96666667, 0.96666667, 0.96666667, 0.96666667])
```

Tujuan dengan adanya training berulang – ulang sebanyak nilai K yang diuji bertujuan untuk mencari rata – rata akumulasi akurasi skor dan nilai k terbaik.

8. Menentukan Akumulasi Akurasi Skor & K Terbaik.

```
✓ [17] plt.plot(range(1,Ks),mean_acc,'g')
0d plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc, alpha=0.10)
plt.legend(('Accuracy ', '+/- 3xstd'))
plt.ylabel('Accuracy ')
plt.xlabel('Number of Nabors (K)')
plt.tight_layout()
plt.show()
```



Dari proses training model secara terus menerus sesuai k yang ada kita bisa membuat grafik lines untuk melihat akumulasi score dari setiap k, maka dihasilkan nilai k terbaik ada pada k-3 dan skor akurasi terbaiknya ada pada nilai lebih dari 0.96.

Selain menggunakan grafik / visual, untuk mendapatkan nilai k dan skor akurasi terbaik dan akurat setelah training model bisa menggunakan code:

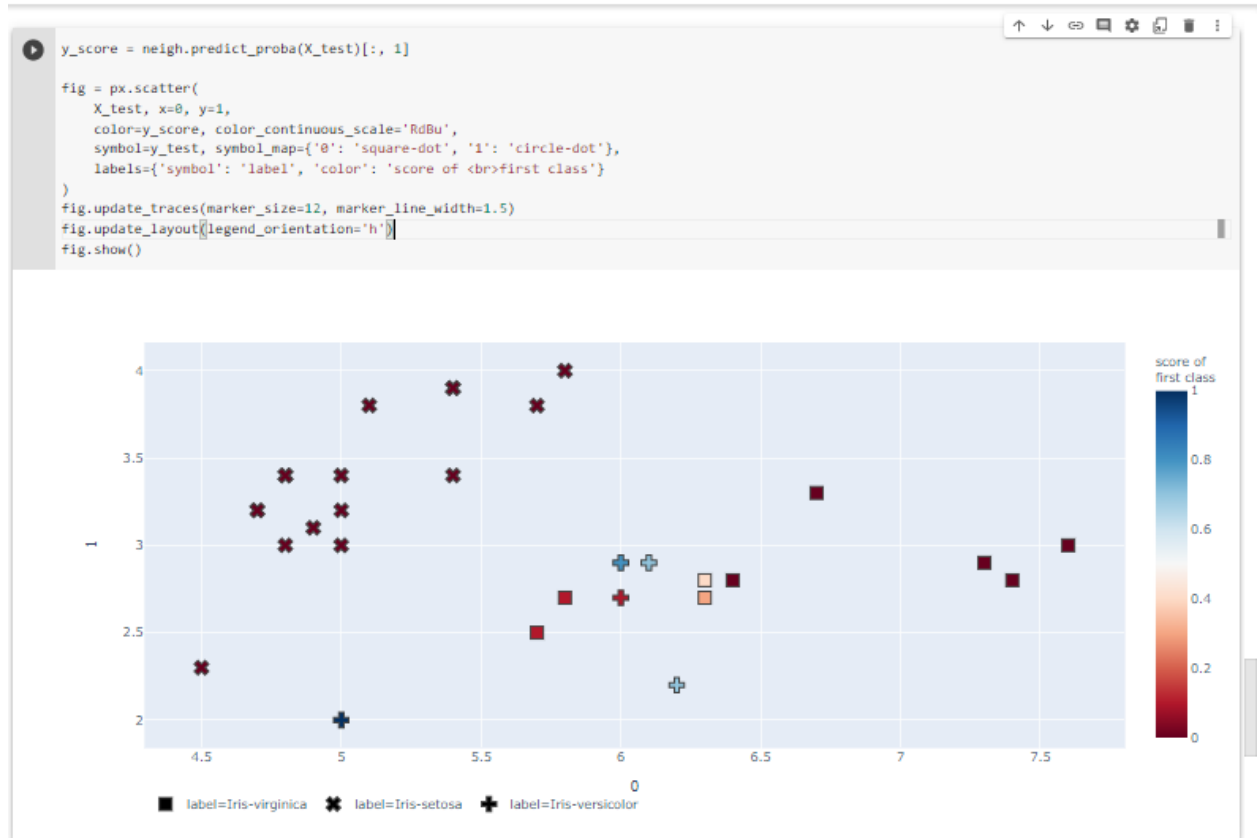
```

[18] print( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()+1)

The best accuracy was with 0.9666666666666667 with k= 3

```

9. Melihat Cluster Secara Visual



10. Melihat Cluster Secara Visual

Untuk pengujian sendiri dengan cara melakukan predik pada sample data yang sudah disiapkan dimana sample data itu sendiri merupakan sebagian data (3 rows) yang diambil dari file iris.csv


```

[42] !wget -O /content/sample_data/sample_data.csv https://raw.githubusercontent.com/irvandandung/iris-project/master/sample_data.csv
--2021-08-13 12:23:57-- https://raw.githubusercontent.com/irvandandung/iris-project/master/sample_data.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 161 [text/plain]
Saving to: '/content/sample_data/sample_data.csv'

/content/sample_data/sample_data.csv 100%[=====] 161 --.-KB/s in 0s

2021-08-13 12:23:57 (7.42 MB/s) - '/content/sample_data/sample_data.csv' saved [161/161]

[43] datauji = pd.read_csv('/content/sample_data/sample_data.csv')
datauji.head()

```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	150	5.9	3.0	5.1	1.8	Iris-virginica
2	51	7.0	3.2	4.7	1.4	Iris-versicolor

Sebelum memasukan sample data ke dalam func predict untuk dilakukan pengujian, kita harus mengubah data di atas menjadi array list terlebih dahulu dimana data yang diambil hanyalah data column Sepal Length Cm, Sepal Width Cm, Petal Length Cm, Petal Width Cm.

```

[44] x_new = datauji.iloc[:, 1:-1].values
x_new

array([[5.1, 3.5, 1.4, 0.2],
       [5.9, 3. , 5.1, 1.8],
       [7. , 3.2, 4.7, 1.4]])

```

Setelah itu lakukan pengujian data di atas dengan memasukkannya ke dalam func predict.

```

neigh = KNeighborsClassifier(n_neighbors = 3).fit(X_train,y_train)
y_predict = neigh.predict(x_new)
print('prediction for item x_new :', y_predict)
print('x_new index-0 id 1 adalah Iris-virginica:', y_predict[0] == 'Iris-setosa')
print('x_new index-1 id 150 adalah Iris-setosa:', y_predict[1] == 'Iris-virginica')
print('x_new index-1 id 51 adalah Iris-versicolor:', y_predict[2] == 'Iris-versicolor')

prediction for item x_new : ['Iris-setosa' 'Iris-virginica' 'Iris-versicolor']
x_new index-0 id 1 adalah Iris-virginica: True
x_new index-1 id 150 adalah Iris-setosa: True
x_new index-1 id 51 adalah Iris-versicolor: True

```

Ketika hasil predict sesuai expektasi dan menghasilkan nilai True yang menandakan data yang di compare itu sama maka data train dan data test yang telah di olah di atas itu telah berhasil menghasilkan keakuratan data yang cukup baik.

BAB III

KESIMPULAN

Dari pembahasan yang sudah dijelaskan di bab 3 dapat disimpulkan bahwa salah satu algoritma yang dapat digunakan untuk mengklasifikasikan ke 3 kelompok bunga iris tersebut adalah dengan menggunakan Algoritma kNN (k-Nearest Neighbor). Algoritma kNN (k-Nearest Neighbor) ini adalah algoritma klasifikasi berdasarkan tetangga terdekat dimana hasil yang didapatkan dari hasil pengujian yang telah dilakukan cukup memuaskan, karena memiliki nilai akurasi skor yang cukup tinggi (hampir mendekati 1). Dengan akurasi serta pengujian yang telah dilakukan, hasil pengklasifikasian 3 kelompok bunga iris tersebut cukup akurat menggunakan algoritma ini.

REFERENSI

- Parvin H, Alizadeh H, Bidgoli B M. *MKNN: Modified K-Nearest Neighbor*. Proceedings of the Word Congress on Engineering and Computer Science 2008 (WCECS 2008). San Francisco. 2008: 831-834.
- N. Krisandi, Helmi, and B. Prihandono, “Algoritma K-Nearest Neighbor dalam Klasifikasi Data Hasil Produksi Kelapa Sawit pada PT. Minamas Kecamatan Parindu,” *Bul. Ilm. Math. Stat. dan Ter.*, vol. 02, no. 1, pp. 33–38, 2013.
- D. Santoso, D. E. Ratnawati, and Indriati, “Perbandingan Kinerja Metode Naïve Bayes, K-Nearest Neighbor, dan Metode Gabungan K-Means dan LVQ dalam Pengkategorian Buku Komputer Berbahasa Indonesia berdasarkan Judul dan Sinopsis,” *Repos. J. Mhs. PTIIK UB*, vol. 4, no. 9, 2014.
- G. Toker and Ö. Kirmemiş, “TEXT CATEGORIZATION USING k-NEAREST NEIGHBOR CLASSIFICATION.”
- X. Yan, W. Li, W. Chen, W. Luo, C. Zhang, and Q. Wu, “Weighted K-Nearest Neighbor Classification Algorithm Based on Genetic Algorithm,” *TELKOMNIKA*, vol. 11, no. 10, pp. 6173–6178, 2013.
- D. Arifin, I. Ariesianti, and A. Z. Arifin, “Implementasi Algoritma K-Nearest Neighbour Yang Berdasarkan One Pass Clustering Untuk Kategorisasi Teks,” pp.1–7.
- Suguna N, Thanushkodi K. An Improved k-Nearest Neighbor Classification Using Genetic Algorithm. *International Journal of Computer Science Issues*. 2010. 7(4): 18-21.
- Bhatia N, Vandana. Survey of Nearest Neighbor Techniques. *International Journal of Computer Science and Information Security*. 2010. 8(2): 302-305.
- Leidiyana, H. (2013). Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Pemilikan Kendaraan Bermotor. *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic*, 65-76.
- Modul Pertemuan ke 8 Tentang Klasifikasi Menggunakan KNN