LEARNING : KNN
Observasi
oleh: Muhammad Shulhannur

PROBLEM :

Bangunlah suatu program komputer melakukan analisis, desain, dan implementasi algoritma k-nearest neighbor (kNN), yang mel akukan seleksi dan estimasi model kNN tersebut menggunakan 5-fold cross-validation ,yang menghasilkan akurasi tertinggi, jika diberikan dataset (himpunan data) Pima India Diabetes Dataset (PIDD) pada file "Diabetes.csv", yang berisi 768 objek data (baris), sehingga harus membuat lima datasets baru menggunakan skema 5-fold cross-validation. Pertama, bagi objek data ke dalam lima subsets (sub himpunan) dengan porsi yang sama, masing-masing berisi sat per lima (20%) data. Kemudian, buat lima dataset baru dengan komposisi objek-objek data pada training set (data latih) dan testing set (data uji) sebagai berikut:

1. Baris ke-1 sampai baris ke-614 sebagai training set dan sisanya sebagai testing set
2. Baris ke-1 sampai baris ke-461 ditambah baris ke-642 sampai 768 sebagai training set dan yang lain sebagai testing set
3. Baris ke-1 sampai baris ke-307 ditambah baris ke-462 sampai 768 sebagai training set dan yang lain sebagai testing set
4. Baris ke-1 sampai baris ke-154 ditambah baris ke-308 sampai 768 sebagai training set dan yang lain sebagai testing set
5. Baris ke-155 sampai sampai 768 sebagai training set dan yang lain sebagai testing set

HAL-HAL YANG DIOBSERVASI :

1. Penggunaan Bahasa Pemrograman, Tools, dan Libraries

Penulis menggunakan bahasa pemrograman Python, dengan editor - compiler online Google Colab, dan memanggil library yang terdiri atas :

```
[24] import numpy as np
     import pandas
     import io
     from sklearn.preprocessing import StandardScaler
     from sklearn.impute import SimpleImputer
     from operator import itemgetter
     from collections import Counter
     from google.colab import files
```

Justifikasi yang dapat penulis berikan mengenai preferensi tersebut adalah karena penggunaan Python tidak memerlukan compile, serta Google Colab memberikan kemudahan dalam menulis dokumentasi, markdown dan notes.

Proses yang harus dibangun (bisa berupa fungsi/prosedur):

Perhitungan ukuran jarak

```
▾ CALCULATING THE MANHATTAN DISTANCE

[29] def manhattan(x1, x2):
         return np.sum(np.abs(np.array(x1)-np.array(x2)))

▾ CALCULATING THE EUCLIDEAN DISTANCE

[30] def euclidean(x1, x2):
         return np.sqrt(np.sum((np.array(x1)-np.array(x2))**2))
```

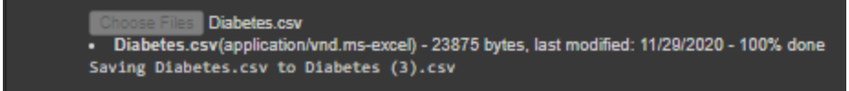
Prapemrosesan data

CONSTANTS AND GLOBAL VARIABLES

```
[25] k = 3
     AverageValidation = []
     iterationA = 20
     iterationB = 5
```

IMPORTED/READ DATA

```
[26] uploaded = files.upload()
     df = pandas.read_csv(io.BytesIO(uploaded['Diabetes.csv']))
```



DIVIDE AS ARRAY LISTS

```
[27] x = df.iloc[:,:-1]
     y = df['Outcome'].values
```

IN CASE OF MISSING VALUE

```
[28] x = x.replace(0, np.nan)

     imp_mean = SimpleImputer(missing_values=np.nan, strategy='mean')
     imp_mean.fit(x)
     x = imp_mean.transform(x)
```

Klasifikasi kNN

KNN CLASSIFICATION

```
[31] def kNN(k):
       result = []
       hyperparameter = []
       kset = []
       for i in testing_x:
         result = []
         for j in training_x:
           result.append(manhattan(i, j))
         sorting = np.argsort(result)[:k]
         for l in sorting:
           kset.append(training_y[l])
         mode = Counter(kset).most_common(1)[0][0]
         kset = []
         hyperparameter.append(mode)
       return hyperparameter
```

Pemilihan nilai k terbaik dan Perhitungan rata-rata akurasi kNN menggunakan 5-fold cross-validation

BEST VALUE OF K AND AVERAGE ACCURACY CALCULATION

```
[32] for i in range(iterationA):
       Accuracy = 0
       for j in range(iterationB):
         if j == 0:
           training_x = x[:614]
           training_y = y[:614]
           testing_x = x[614:]
           testing_y = y[614:]
         elif j == 1:
           training_x = np.concatenate((x[:461], x[614:]))
           training_y = np.concatenate((y[:461], y[614:]))
           testing_x = x[461:614]
           testing_y = y[461:614]
         elif j == 2:
           training_x = np.concatenate((x[:307], x[461:]))
           training_y = np.concatenate((y[:307], y[461:]))
           testing_x = x[307:461]
           testing_y = y[307:461]
         elif j == 3:
           training_x = np.concatenate((x[:154], x[307:]))
           training_y = np.concatenate((y[:154], y[307:]))
           testing_x = x[154:307]
           testing_y = y[154:307]
         elif j == 4:
           training_x = x[:155]
           training_y = y[:155]
           testing_x = x[155:]
           testing_y = y[155:]
         scaler = StandardScaler()
         training_x = scaler.fit_transform(training_x)
         testing_x = scaler.transform(testing_x)
         Accuracy += (np.sum(kNN(k) == testing_y) / len(testing_y))*100
       print(" For K value of =",k,", the average accuracy is = ", Accuracy/5)
       AverageValidation.append([k, Accuracy/5])
       AverageValidation = sorted(AverageValidation, key=itemgetter(1), reverse=True)
       k += 1
     print('K = ', AverageValidation[0][0], ', is the best value of K, which has the accuracy of = ', AverageValidation[0][1])
```

Output dari sistem adalah:

Nilai k terbaik hasil pembelajaran kNN dan Rata-rata akurasi kNN menggunakan 5-fold cross-validation.

```
 For K value of = 3 , the average accuracy is =  72.23966674019452
 For K value of = 4 , the average accuracy is =  72.49855817555655
 For K value of = 5 , the average accuracy is =  74.71186566474927
 For K value of = 6 , the average accuracy is =  72.92312822108427
 For K value of = 7 , the average accuracy is =  73.21634373260896
 For K value of = 8 , the average accuracy is =  73.57523443407716
 For K value of = 9 , the average accuracy is =  74.90889954281184
 For K value of = 10 , the average accuracy is =  74.61907517468404
 For K value of = 11 , the average accuracy is =  75.16588423890985
 For K value of = 12 , the average accuracy is =  75.19956858120182
 For K value of = 13 , the average accuracy is =  75.62286746722022
 For K value of = 14 , the average accuracy is =  75.39723803901907
 For K value of = 15 , the average accuracy is =  76.24404905567941
 For K value of = 16 , the average accuracy is =  76.01502432996531
 For K value of = 17 , the average accuracy is =  76.56713127685204
 For K value of = 18 , the average accuracy is =  75.69088003979088
 For K value of = 19 , the average accuracy is =  76.17794737648879
 For K value of = 20 , the average accuracy is =  75.65612947438107
 For K value of = 21 , the average accuracy is =  75.78387962036365
 For K value of = 22 , the average accuracy is =  75.2959634592875
K =  17 , is the best value of K, which has the accuracy of =  76.56713127685204
```