LAPORAN TUGAS

Nama	: Rakha Asyrofi	Mata Kuliah	: Kecerdasan Komputasional
NRP	: 05111950010038	Dosen	: Dr. Eng. Chastine Fatichah, S.kom, M.Kom.

Lakukan Coding di Python untuk menyelesaikan problem klasifikasi dengan tahapan dibawah ini:

1. Download dataset Cardiotography di UCI Machine Learning Repository dengan link URL https://archive.ics.uci.edu/ml/datasets/Cardiotocography
2. Lakukan proses normalisasi data jika datanya terdapat perbedaan skala (opsional)
3. Lakuka proses reduksi dimensi menggunakan PCA atau LDA pada dataset yang sudah ternormalisasi.
4. Lakukan proses seleksi fitur menggunakan salah satu metode pada dataset yang sudah ternormalisasi.
5. Lakukan proses klasifikasi menggunakan metode SVM pada dataset dengan fitur lengka, dengan fitur yang sudah direduksi dan dengan fitur yang sudah dilakukan seleksi.
6. Lakukan proses klasifikasi menggunakan metode ANN pada dataset yang dengan fitur lengkap dan fitur yang sudah direduksi dan dengan fitur yang sudah dilakukan seleksi.
7. Bandingkan hasil akurasi, sensitivity dari ujicoba ke5 dan ke6 dan ambil kesimpulan kombinasi metode dengan hasil yang terbaik.

Sebelumnya perlu ketahui dahulu data yang kita ambil yaitu Cardiotography (CTG). CTG merupakan proses yang digunakan untuk mencatat Fetal Hearth Rate (FHR) dan Uterine Constrains (UC) selama kehamilan. Hasil dari analisa CTG ini kita gunakan untuk mengklasifikasikan Janin menjadi salah satu beberapa pola morfologis atau keadaan janin. Klasifikasi ini secara konvensional telah dilakukan oleh dokter kandungan berdasarkan standar dan pedoman yang disetujui tetap tidak menghilangkan sifat tugas atau memungkinkan kesalahan klasifikasi yang tinggi. Teknik machine learning yang digunakan membuat klasifikasi ini dengan akurasi tinggi tetapi tidak ada perbandingan luas untuk menentukan model terbaik yang telah dilaukan. Peneliti melakukan prediksi untuk keadaan jani dan pola morfologis menggunakan 2 model sebagai dimension reduction dan 2 model untuk klasifikasi[1]. Peneliti juga mengeksplorasi korelasi antara dua set label untuk melihat bagaimana learning rate salah satunya dapat mempengaruhi prediksi yang lain. Peneliti kemduian menunjukan bahwa model yang kita gunakan memiliki kinerja lebih baik daripada para peneliti lain yang menggunakan set data UCI tersebut, sebagai perbandingan antara ANN dan SVM sehingga didaptkan nilai yang sesuai dengan yang kita harapkan [2].

Dari algoritma machine learning yang digunakan, SVM dengan RBF Linear lebih bagus dengan Akurasi sampai 96.47% sedangkan ANN kernel Log memiliki akurasi 41.78% dari sebuah hasil perbandingan antara kedua klasifikasi tersebut. Hasil ini nantinya diharapkan memprediksi pola morfologis dengan asumsi kondisi Janin yang diketahui dan memprediksi keadaan janin tersebut ketika pola morfologis diketahui memberikan akurasi yang lebih baik daripada prediksi semula. Ini disebabkan oleh korelasi yang ada antar kedua perbandingan metode tersebut.

1. Langkah pertama yang saya kerjakan yaitu membuatkan library yang dibutuhkan seperti sklearn, pandas, numpy dan itertools. Library tersebut berguna untuk memanggil fungsi yang digunakan sebagai perintah code untuk melaksanakan pekerjaan di computer. Kurang lebih tampilan yang didapatkan seperti berikut:

1. Download dataset Cardiotocography di UCI Machine Learning Repository dengan link URL https://archive.ics.uci.edu/ml/datasets/Cardiotocography

Import Library

```
import pandas as pd
import numpy as np
from sklearn.model_selection import StratifiedShuffleSplit
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
import itertools
```

2. Selanjutnya kita lakukan proses data cleaning dan preprocessing, berguna untuk menyiapkan file dataset sebagai sumber yang digunakan Namun karena didalam file excel tersebut terdapat berbagai macam sheet didalamnya yatu Description, Data, Columns.. Sehingga perlu dilakukan proses persing guna, untuk mendapatkan data yang kita inginkan. Lalu kita ambil data tersebut dan tampilkan satu sheet yang diperlukan sesuai dengan row dan column yang tersedia dalam column tersebut dan lakukan penghapusan data-data yang tidak diperlukan.

Data Cleaning & Preprocessing

```
[2] fn = r'/content/CTG.xls'
    xl = pd.ExcelFile(fn)
    xl.sheet_names

    for sh in xl.sheet_names:
      df = xl.parse(sh)
      print('Processing: [{}] ...'.format(sh))
      print(df.head())
```

```
Processing: [Description] ...
  Unnamed: 0  Unnamed: 1  ... Unnamed: 12                  Unnamed: 13
0        NaN         NaN  ...         NaN                          NaN
1        NaN         NaN  ...         NaN                          NaN
2  Worksheet         NaN  ...         NaN                          NaN
3        NaN         NaN  ...         NaN  DR is removed since p(K-W)=1
4        NaN         NaN  ...           p                          NaN

[5 rows x 14 columns]
Processing: [Data] ...
  Unnamed: 0 Unnamed: 1 Unnamed: 2  ...     22 Unnamed: 44   23
0          b          e         AC  ...  CLASS         NaN  NSP
1        240        357          0  ...      9         NaN    2
2          5        632          4  ...      6         NaN    1
3        177        779          2  ...      6         NaN    1
4        411       1192          2  ...      6         NaN    1

[5 rows x 46 columns]
Processing: [Raw Data] ...
      FileName       Date      SegFile      b  ...   FS  SUSP  CLASS  NSP
0          NaN        NaT          NaN    NaN  ...  NaN   NaN    NaN  NaN
```

```
[3] dfs = {sh:xl.parse(sh) for sh in xl.sheet_names}
    dfs.keys()
    panjang = len(dfs['Raw Data'])
    dfs['Raw Data'].head(panjang)
```

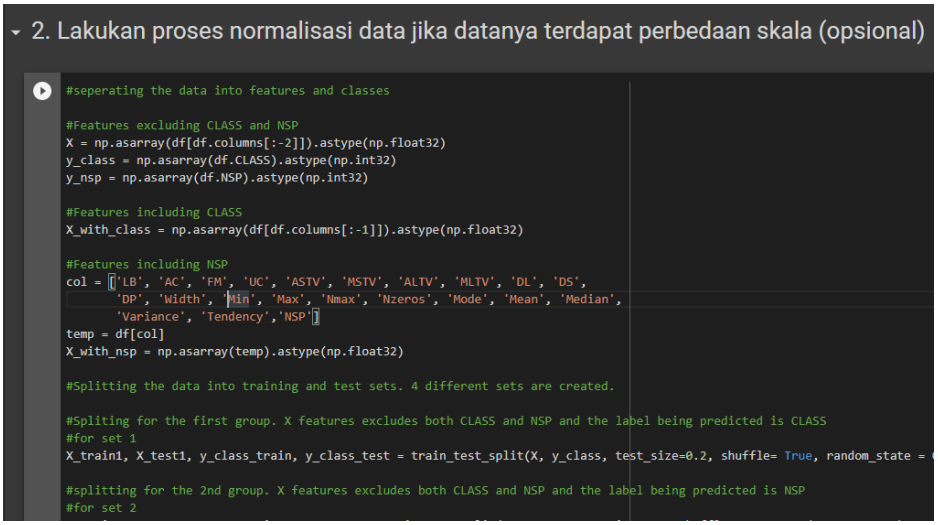
	FileName	Date	SegFile	b	e	LBE	LB	AC	FM	UC	ASTV	MSTV
0	NaN	NaT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Variab10.txt	1996-12-01	CTG0001.txt	240.0	357.0	120.0	120.0	0.0	0.0	0.0	73.0	0.5
2	Fmcs_1.txt	1996-05-03	CTG0002.txt	5.0	632.0	132.0	132.0	4.0	0.0	4.0	17.0	2.1
3	Fmcs_1.txt	1996-05-03	CTG0003.txt	177.0	779.0	133.0	133.0	2.0	0.0	5.0	16.0	2.1
4	Fmcs_1.txt	1996-05-03	CTG0004.txt	411.0	1192.0	134.0	134.0	2.0	0.0	6.0	16.0	2.4
...	...	...	...	...	...	...	...	...	...	...	...	...
2125	S8001045.dsp	1998-06-06	CTG2127.txt	1576.0	3049.0	140.0	140.0	1.0	0.0	9.0	78.0	0.4
2126	S8001045.dsp	1998-06-06	CTG2128.txt	2796.0	3415.0	142.0	142.0	1.0	1.0	5.0	74.0	0.4
2127	NaN	NaT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2128	NaN	NaT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2129	NaN	NaT	NaN	NaN	NaN	NaN	NaN	NaN	564.0	23.0	87.0	7.0

```
#Read the data into a Data Frame
df = pd.read_excel('/content/CTG.xls', sheet_name='Raw Data')

#remove the 3 lines indicating the total
df = df.dropna()

#remove irrelevant columns
df.drop(['FileName', 'Date', 'SegFile', 'b', 'e', 'LBE', 'DR', 'A', 'B', 'C', 'D', '
df = df.reset_index()
df.drop(['index'], axis =1, inplace=True)

df.to_csv('cleaned_data.csv', index=False)
df.columns
```

```
Index(['LB', 'AC', 'FM', 'UC', 'ASTV', 'MSTV', 'ALTV', 'MLTV', 'DL', 'DS',
       'DP', 'Width', 'Min', 'Max', 'Nmax', 'Nzeros', 'Mode', 'Mean', 'Median',
       'Variance', 'Tendency', 'CLASS', 'NSP'],
      dtype='object')
```

3. Selanjutnya melakukan proses normalisasi data jika datanya terdapt perbedaan skala. Dengan mengisi nilai-nilai yang hilang yang masing ditentukan masing-masing kolom untuk meprediksi nilainya. Lalu digunakan atribut sebagai variable independent dan memeperlakukan kolom sebagai variable dependet, membuat tugas kita untuk mengklasifikasikannya menjadi multi-kelas. Terdapat 2 macam yang nanti kita gunakan sebagai proses reduksi dimensi yaitu Principal Component Analyisi (PCA) dan latent Dirichlet Allocation (LDA) serta 2 macam algoritma klasifikasi yaitu Support Vector Machine (SVM) dan Artificial Neural Network (ANN) dari berbagai model yang saya sebutkan tersebut untuk menentukan bagaimana hasil baik dari algoritma tersebut[3].

   Sebelum itu perlu saya jelaskan bagaimana saya bisa memecah fitu dan kelas yang ada pada dataset tersebut. Dengan membuat sebuah Class yang kita labeli sebagai X_with_class dan NSP yang kita labeli dengan X_with_class serta memecah menjadi 4 group yang masing-masing terdiri atas X_train, X_test, y_class_train dan y_class_test.dan seterusnya. Jadi ada 2 model NSP dan 2 model class didalamnya.

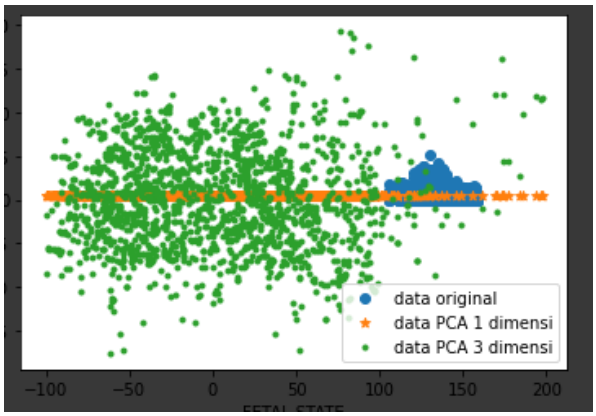


4. Selanjutnya kita lakukan proses reduksi dimensi menggunakan PCA maupun LDA, saya coba gunakan kedua metode kepada data yang telah ternomalisasi. Kita memisahnya menjadi 2 bagian yaitu PCA dan LDA. Berikut ini penjelasnnya:
   a. Principal Componen Analysisi (PCA)

      Pertama kita memerlukan library numpy dan matplotlib sebagai fitur plot gambar yang diperlukan untuk sebuah data, dengan menggunakan fungsi pca yang didalamnya nanti meliputi dari input data dan berapa banyak dimensi yang diperlukan. Lalu dengan penggunaan centering, berguna untuk mengurangi nilai mean dari row datanya didalamnya kita gunakan fungsi mean data by row, lalu lakukan perulangan sebanyak rownya, kemudian kita kurangi dengan data sebelumnya.[3]

      Kemudian kita lakukan proses eigen decomposition dari matrix kovarians yang dibuat. Dari hasil tersebut lalu diproyeksikan dengan matriks proyeksi n-eigenvector (v) serrta n-eigenvalue (lamda) yang paling besar. Hasil data input kita proyeksikan dengan cara dilakukan dengan dot product dengan matrix proyeksinya. Kemudian kita dapatkan nilai matriks kovarian, eigen value dan

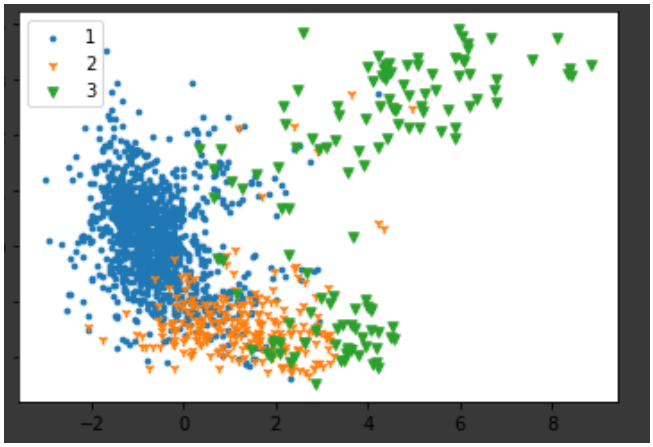
eigenvectornya. Serta mengembalikan nilai data terebut ke hasil proyeksi kembali. Lalu gunakan kita buat variable yang menunjukkan X_train sebagai data asli dan kita gunakan fungsi data_terproyeksi_dimensi menjadi berapa dimensi yang diperlukan. Lalu kita membuat sebuah plotting data sesuai dengan visualisasi yang diperlukan. Berikut ini hasil yang didapatkan dari hasil PCA



b. Latent Dirichlet Allocation (LDA)

Selain PCA, peneliti mencoba menggunakan LDA sebagai alternatif lain dalam mereduk dimensi, dengan fungsi skelearn.discriminant_analysis dan kita ambil Linear Discriminant Analysis kita gunakan misalnya kita memerlukan dimesi tujuan berupa 2. Lalu kita merepresentasikan X sebagai data training dan y sebagai label untuk pembanding. Kita lakukan proses fitting didalam komponen tersebut [4].

Lalu menstranformasikan data asli tersebut ke data LDA, sehingga nantinya kita bisa visualisasikan dengan itertools untuk mendapatkan 2 label yang didapatkan.



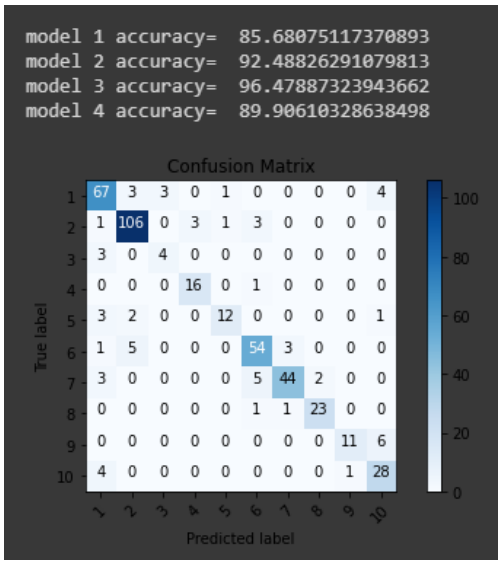
5. Lalu kita lakukan proses seleksi fitur menggunakan salah satu metode yang kita dapatkan hasil normalisasinya. Didapatkanlah, 4 model data yaitu X_train1 samapi Xtrain4 dan X_tes1 samapai X_test4
6. Setelah dilakukan proses normalisasi barulah kita menggunakan metode SVM dan ANN pada dataset untuk mendapatkan fitur lengkap yang telah direkduksi diemsinya sebelumnya untuk diseleksi. Ada 2 macam model klasifikasi yang digunakan yaitu SVM dan .

a. Support Vector Machine (SVM)

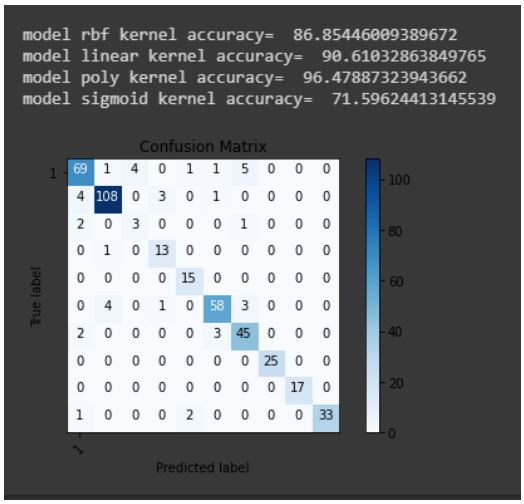
Proses untuk membentuk sebuah SVM classifier, kita membuat sebuah SVM classifier instance didalamnya dengan memanggil scaler, lalu kita laukan proses Grid untuk menari parameter yang pas untuk proses 5-fold cross validation'nya..

Lalu menseleksi dari dari estimator yang dibutuhkan. Sehingga didaltkan nilai mean test score dari sebuah data tersebut [2].

Lalu kita buat confussion matrix untuk membuat mendapatkan nilai cross validation antar 4 model kelas tersebut. Sehingga nantinya kita bisa mencetak hasil klasifikasinya. Yang kita namakan targetmya menjadi normal, suspect dan pathological. Dari beberapa iterasi yang diperlukan kita dapatkan hasil menjadi sebagai berikut. Ternyata model ke 3 memiliki akurasi yang tinggi dari sebuah klasifikasi SVM dengan nilai sekitar 96.47% kecocokan.

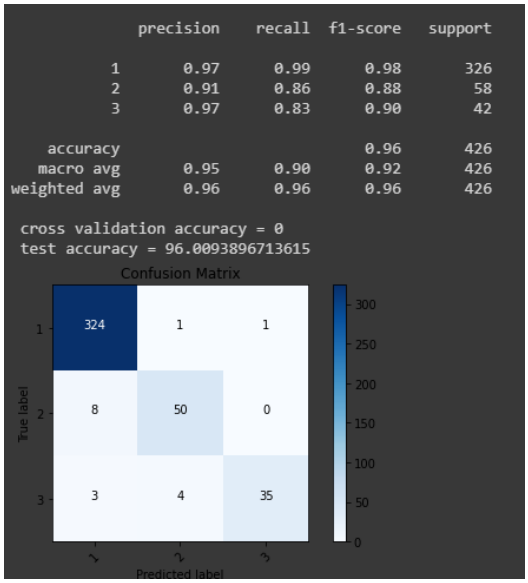


Adapun beberapa percobaan dengan SVM menggunakan library yang disediakan oleh Skelearn dengan berbagai kernel didalamnya yaitu rbf, linear, poly dan sigmoid, dengan menggunakan parameter C, penalty error dengan nilai term: 1 serta penggunaan kriteria toleransi dengan menggunakan 1e-3 misalnya maka didapatkan bahwa akurasi paling tinggi di model 4 menggunakan linear poly dengan nilai akurasinya sekitar 96.47 %.



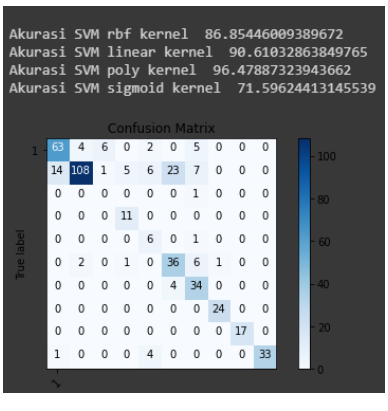
b. Artificial Neural Network (ANN)

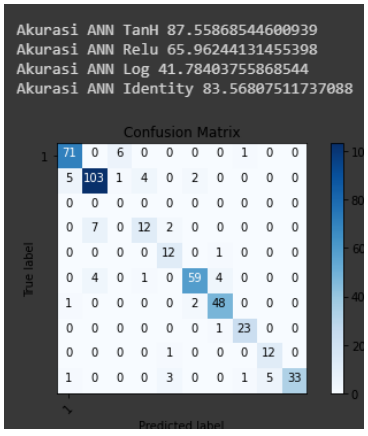
Klasifikasi yang lain digunakan selain SVM yaitu menggunakan MLP (Multilayer Perceptron). Hal pertama yang kita buat adalah membuat sebuah 4 hidden layer dengan 30 neuron didalamnya [2]. Lalu kita ambil nilai cross validation yang paling akurat. Maka didapatkan hasil yaitu pada data model 4 dengan iterasi yang diperlukan sampai 286, dengan tingkat akurasi mencapai 90.61% sedangkan model 1 dengan iterasi 341 memiliki akurasi senilai 81.22%, model 2 dengan iterasi 210 tingkat akurasi mencapai 91.54% dan model 3 iterasi 121 dengan tingkat akurasi 96.01%. Maka perbandingan antar model tersebut nilai model 3 lebih bagus daripada model yang lain.terlihat di gambar dibawah ini.



Adapun apabila kita gunakan ANN pada sklearn yang digunakan fungsi aktivasinya yaitu berupa relu, tanh, identiy. Lalu penggunaan Solver dengan adam serta alpha yang digunakan 1e-4 dipilih dari yang terbaik. Lalu dibuat learning rate secar invscalling maka didapatkan hasil pada model 4 didapatkan kernel tanh memiliki akurasi lebih tinggi dari yang lain sekitar 87.55%

7. Setelah melalui proses yang Panjang dan lama maka bisa disimpulkan dari hasil yang didapatkan bahwa dari perbandingan hasil akurasi, sensitivity dan specifity serta melalui ujicoba diantar kedua klasifikasi tersebut didapatkan kesimpulan bahwa metode yang digunakan terbaik adalah dengan SVM dengan kernel poly lebih baik dengan nilai akurasi 96.47% dan nilai ANN dengan aktivasi Log yang paling rendah sekitar 41.78%





Referensi

[1] A. Subasi and M. I. Gursoy, “EEG signal classification using PCA, ICA, LDA and support vector machines,” *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8659–8666, 2010, doi: 10.1016/j.eswa.2010.06.065.

[2] M. KavitaMahajan and M. Rajput, “A Comparative study of ANN and SVM for EEG Classification,” *Int. J. Eng.*, vol. 1, no. 6, pp. 1–6, 2012.

[3] L. J. Rozario, M. R. Haque, Z. Islam, and M. S. Uddin, “Quantitative Analysis of PCA , ICA , LDA and SVM in Face Recognition,” vol. 8, no. 9, pp. 1613–1616, 2014.

[4] J. Mazanec, M. Melišek, M. Oravec, and J. Pavlovičová, “Support vector machines, PCA and LDA in face recognition,” *J. Electr. Eng.*, vol. 59, no. 4, pp. 203–209, 2008.