CERDAS MENGUASAI GIT

CERDAS MENGUASAI GIT
Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center

Kreatif Industri Nusantara

Penulis:
Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:
M. Yusril Helmi Setyawan

Penyunting:
Syafrial Fachrie Pane
Khaera Tunnisa
Diana Asri Wijayanti

Desain sampul dan Tata letak:
Deza Martha Akbar

Penerbit:
Kreatif Industri Nusantara

Redaksi:
Jl. Ligar Nyawang No. 2
Bandung 40191
Tel. 022 2045-8529
Email : awangga@kreatif.co.id

Distributor:
Informatics Research Center
Jl. Sariasih No. 54
Bandung 40151
Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

'Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.'
Imam Syafi'i

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

DAFTAR ISI

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

Bandung, Jawa Barat
Februari, 2019

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.

ACRONYMS

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

GLOSSARY

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

SYMBOLS

A Amplitude

& Propositional logic symbol

a Filter Coefficient

$\mathcal{B}$ Number of Beats

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

INTRODUCTION

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABC\mathcal{D}\mathcal{E}\mathcal{F}\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

BAB 1

CHAPTER 1

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

BAB 2

CHAPTER 2

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

BAB 3

CHAPTER 3

3.1 1174006 - Kadek Diva Krishna Murti

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

```
@inproceedings{awangga2017colenak,
  title={Colenak: GPS tracking model for post-stroke rehabilitation
    program using AES-CBC URL encryption and QR-Code},
  author={Awangga, Rolly Maulana and Fathonah, Nuraini Siti and
    Hasanudin, Trisna Irmayadi},
  booktitle={Information Technology, Information Systems and
    Electrical Engineering (ICITISEE), 2017 2nd International
    conferences on},
  pages={255--260},
  year={2017},
  organization={IEEE}
}
```


By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

Gambar 3.1 Kecerdasan Buatan.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

2. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

3. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

3.1.1 Teori

3.1.2 Praktek

3.1.3 Penanganan Error

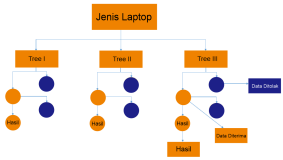
3.1.4 Bukti Tidak Plagiat

Gambar 3.2 Kecerdasan Buatan.

3.2 1174021 - Muhammad Fahmi

3.2.1 Soal Teori

1. Jelaskan apa itu random forest, sertakan gambar ilustrasi buatan sendiri.
   Random Forest merupakan algoritma yang digunakan terhadapap klasifikasi data dalam jumlah yang besar. Klasifikasi pada random forest dilakukan dengan penggabungan dicision tree dengan melakuakn training terhadap sempel data yang dimiliki. Semakin banyak dicision tree maka data yang di dapat akan semakin akurat. Untuk gambar Random Forest dapat dilihat dibawah ini :



Gambar 3.3 Random Forest.

2. Jelaskan cara membaca dataset kasus dan artikan makna setiap file dan isi field masing-masing file.

   - Pertama download dataset terlebih dahulu lalu buka dengan menggunakan software spyder guna melihat isi dari dataset tersebut.
   - Data tersebut memiliki extensi file bernama .txt dan didalamnya terdapat class dari field.
   - Misalnya saja pada data jenis burung memiliki file index dan angka, dimana index berisi angka yang memiliki makna berupa jenis burung.
   - bahkan nama burung sedangkan field memiliki isi nilai berupa 0 dan 1 yang dimana sifatnya boolean atau Ya dan Tidak.
   - Hal ini dikarenakan komputer hanya dapat membaca bilangan biner maka dari itu field yang di isikan berupa angka.
   - Artinya angka 0 berarti tidak dan angka 1 berarti Ya.

3. Jelaskan apa itu cross validation.
   Cross Validation merupakan sebuah teknik validasi model yang berfungsi untuk melakukan penilaian bagaimana hasil analisis statistik akan digeneralisasi ke data yang lebih independen. Cross validation digunakan dengan tujuan prediksi, dan bila kita ingin memperkirakan seberapa akurat model model prediksi yang dilakukan dalam sebuah praktek. Tujuan dari cross validation yaitu untuk mendefinisikan dataset guna menguju dalam fase pelatihan untuk membatasi masalah seperti overfitting dan underfitting serta mendapatkan wawasan tentang bagaimana model akan digeneralisasikan ke set data independen.

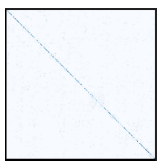
4. Jelaskan apa arti score 44 % pada random forest, 27% pada decission tree dan 29 % dari SVM.
   Dimana Score 44 % diperoleh dari hasil pengelohan dataset jenis burung. Dimana akan dilakukan proses pembagian data testing dan data training lalu diproses dan menghasilkan score sebanyak 44 % dimana menjelaskan bahwa score tersebut digunakan sebagai pembanding dalam tingkat keakuratannya. Pada dicision tree akan memperoleh data lebih kecil yaitu sebanyak 27 % hal ini dikarenakan data yang diolah menggunakan dicision tree dibagi menjadi beberapa tree dan lalu disimpulkan untuk mendapatkan data yang akurat. Pada SVM

akan memperoleh score sebanyak 29 % hal ini dikarenakan data yang dimiliki masih bernilai netral sehingga tingkat keakuratannya masih belum jelas.

5. Jelaskan bagaimana cara membaca confusion matriks dan contohnya memakai gambar atau ilustrasi sendiri.
   Untuk membaca confusion matriks dapat menggunakan source code sebagai berikut :

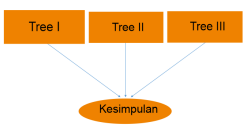
```
fig = plt.figure() #hasil plot sebagai figure
fig.clf() #figure di ambil dari clf
ax = fig.gca(projection='3d') #ax sebagai projection 3d
x = rf_params[:,0] #x sebagai index 0
y = rf_params[:,1] #y sebagai index 1
```

   Dimana numpy akan mengurus semua data yang berhubungan dengan matrix. Pada source code tersebut digunakan dalam melakukan read pada dataset burung dengan menggunakan metode confusion matrix. Dalam confusion matrix memiliki 4 istilah yaitu True Positive yang merupakan data posotif yang terditeksi benar, True Negatif yang merupakan data negatif akan tetapi terditeksi benar, False Positif merupakan data negatif namun terditeksi sebagai data positif, False Negatif merupakan data posotif namun terditeksi sebagai data negatif. Adapun contoh hasil read dataset menggunakan confusion matrix dapat dilihat dibawah ini :



Gambar 3.4 Confusion Matriks.

6. Jelaskan apa itu voting pada random forest disertai dengan ilustrasi gambar sendiri
   Voting pada random forest ialah sebuah proses pemilihan dari tree yang dimana akan dimunculkan hasilnya dan disimpulkan menjadi informasi yang pasti. Untuk lebih jelasnya saya akan memberikan sebuah contoh bagaimana voting bekerja :



Gambar 3.5 Decision Tree.

Dimana ditunjukkan pada gambar diatas terdapat 3 tree. Dalam tree tersebut akan dilakukan proses voting. Saya akan memberikan contoh kasus, dimana akan diadakan voting untuk menentukan sebuah laptop. Dalam tree akan diberikan sejumlah data misalnya saja data tersebut berupa gambar, yang dimana data tersebut akan dipilih dengan cara voting. Hasil voting akhir dari setiap tree menunjukkan laptop asus, yang berarti kesimpulan dari data yang telah diberikan menyatakan gambar tersebut adalah laptop asus. Bagaimana apabila terjadi perbedaan data misalnya saja pada tree 1 dan 2 menunyatakan laptop asus sedangkan pada tree 3 menyatakan laptop HP, maka kesimpulan yang di ambil adalah laptop asus dikarenakan hasil voting terbanyak adalah laptop asus.

3.2.2 Praktek Program

1. Buat aplikasi sederhana menggunakan pandas dan jelaskan arti setiap baris kode yang dibuat(harus beda dengan teman sekelas).

```
# In[44]: Soal1

import pandas as fahmi #melakukan import pada library pandas
    sebagai fahmi

laptop = {"Nama Laptop" : ['Asus','HP','Lenovo','Samsung']} #
    membuat varibel yang bernama laptop, dan mengisi dataframe
    nama2 laptop
```

Kode di atas digunakan untuk mengimpor atau mengirim library pandas sebagai fahmi, Hasilnya adalah sebagai berikut :

```
                Nama Laptop
0     Fahmi Punya Laptop Asus
1       Fahmi Punya Laptop HP
2   Fahmi Punya Laptop Lenovo
3  Fahmi Punya Laptop Samsung
```

Gambar 3.6 Hasil Soal 1.

2. buat aplikasi sederhana menggunakan numpy dan jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas)

```
import numpy as fahmi #melakukan import numpy sebagai fahmi

matrix_x = fahmi.eye(10) #membuat matrix dengan numpy dengan
    menggunakan fungsi eye
matrix_x #deklrasikan matrix_x yang telah dibuat

print (matrix_x) #print matrix_x yang telah dibuat dengan 10x10
```

Fungsi eye disini berguna untuk memanggil matrix identitas dengan jumlah colom dan baris sesuai yang ditentukan. Disini saya telah menentukan 10 colom

dan baris maka dari itu hasilnya akan memunculkan matrix identitas 10x10, Hasilnya adalah sebagai berikut :

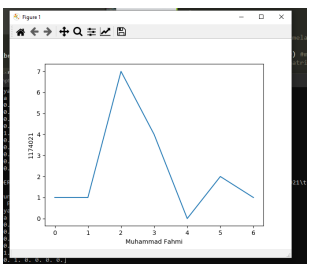
```
[[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]]
```

Gambar 3.7 Hasil Soal 2.

3. buat aplikasi sederhana menggunakan matplotlib dan jelaskan arti dari setiap baris kode(harus beda dengan teman sekelas)

```
import matplotlib.pyplot as fahmi #import matploblib sebagai
    fahmi

fahmi.plot([1,1,7,4,0,2,1]) #memberikan nilai plot atau grafik
    pada fahmi
fahmi.xlabel('Muhammad Fahmi') #memberikan label pada x
fahmi.ylabel('1174021') #memberikan label pada y
fahmi.show() #print hasil plot berbentuk grafik
```

Jalankan source code diatas dengan spyder atau anaconda sehingga hasilnya akan seperti berikut :



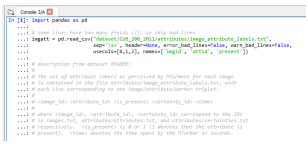
Gambar 3.8 Hasil Soal 3.

4. jalankan program klasifikasi Random Forest pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Source Code pertama pada random forest berfungsi untuk membaca dataset yang memiliki format text file dengan mendefinisikan variabel yang bernama imgatt. Variabel tersebut berisi value untuk membaca data.

```
# In[1]: RANDOM FOREST

import pandas as pd #import library pandas sebagai as

imgatt = pd.read_csv("data/CUB_200_2011/attributes/
    image_attribute_labels.txt",
                     sep='\s+', header=None, error_bad_lines=
    False, warn_bad_lines=False,
                     usecols=[0,1,2], names=['imgid', 'attid',
     'present']) #library imgatt sebagai membaca file csv dari
     dataset, dengan ketentuan yang ada.
```

Hasilnya adalah seperti ini :

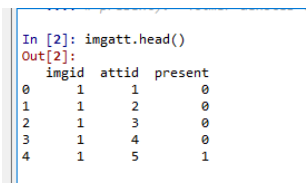


Gambar 3.9 Hasil Soal 4 - 1

- Pada source code berikutnya akan mengembalikan baris teratas dari DataFrame variabel imgatt.

```
# In[2]:

imgatt.head() #menampilkan data yang di baca tadi tetapi hanya
     data paling atas
```

Hasilnya adalah seperti ini :



```
In [2]: imgatt.head()
Out[2]:
   imgid  attid  present
0      1      1        0
1      1      2        0
2      1      3        0
3      1      4        0
4      1      5        1
```

Gambar 3.10 Hasil Soal 4 - 2

- Pada output berikutnya akan menampilkan jumlah kolom dan baris dari DataFrame imgatt.

```
# In[3]:

imgatt.shape #menampilkan jumlah data, kolom
```

Hasilnya adalah seperti ini :

```
In [3]: imgatt.shape
Out[3]: (3677856, 3)
```

Gambar 3.11 Hasil Soal 4 - 3

- Variabel imgatt2 telah menggunakan function yang bernama pivot guna mengubah kolom jadi baris dan sebaliknya dari DataFrame sebelumnya.

```
# In[4]:

imgatt2 = imgatt.pivot(index='imgid', columns='attid', values=
    'present') #membuat variabel baru dari fungsi imgatt,
    dengan mengganti index dan kolom (kebalikan)
```

Hasilnya adalah seperti ini :

```
In [4]: imgatt2 = imgatt.pivot(index='imgid', columns='attid', values='present')
```

Gambar 3.12 Hasil Soal 4 - 4

- Variabel imgatt2 head berfungsi untuk mengembalikan value teratas pada DataFrame imgatt2.

```
# In[5]:

imgatt2.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

Hasilnya adalah seperti ini :

```
In [5]: imgatt2.head()
Out[5]:
attid  1    2    3    4    5    6    7    ...  306  307  308  309  310  311  312
imgid                                      ...
1        0    0    0    0    1    0    0  ...    0    0    1    0    0    0    0
2        0    0    0    0    0    0    0  ...    0    0    0    0    0    0    0
3        0    0    0    0    1    0    0  ...    0    0    1    0    0    1    0
4        0    0    0    0    1    0    0  ...    1    0    0    1    0    0    0
5        0    0    0    0    1    0    0  ...    0    0    0    0    0    0    0

[5 rows x 312 columns]
```

Gambar 3.13 Hasil Soal 4 - 5

- Menghasilkan jumlah kolom dan baris pada DataFrame imgatt2.

```
# In[6]:

imgatt2.shape #menampilkan jumlah data, kolom
```

Hasilnya adalah seperti ini :

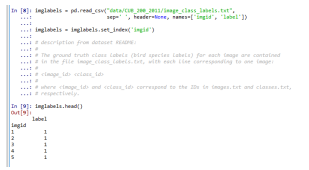
```
In [6]: imgatt2.shape
Out[6]: (11788, 312)
```

Gambar 3.14 Hasil Soal 4 - 6

- Menunjukkan dalam melakukan pivot yang mana imgid menjadi sebuah index yang unik.

```
# In[7]:

imglabels = pd.read_csv("data/CUB_200_2011/image_class_labels.
    txt",
                        sep=' ', header=None, names=['imgid',
    'label']) #baca data csv dengan ketentuan yang ada

imglabels = imglabels.set_index('imgid') #variabel imglabels
    sebagai set index (imgid)
```

Hasilnya adalah seperti ini :

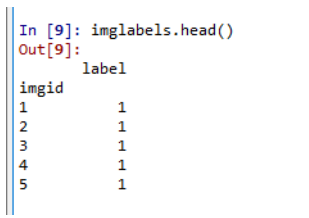


Gambar 3.15 Hasil Soal 4 - 7

- Akan melakukan load jawabannya yang berisi apakah burung tersebut termasuk spesies yang mana. Kolom tersebut yaitu imgid dan label.

```
# In[8]:

imglabels.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

Hasilnya adalah seperti ini :



Gambar 3.16 Hasil Soal 4 - 8

- Menunjukkan bahwa jumlah baris sebanyak 11788 dan kolom 1 yang dimana kolom tersebut adalah jenis spesies pada burung.

```
# In[9]:

imglabels.shape #menampilkan jumlah data, kolom
```

Hasilnya adalah seperti ini :

```
In [10]: imglabels.shape
Out[10]: (11788, 1)
```

Gambar 3.17 Hasil Soal 4 - 9

- Melakukan join antara imgatt2 dengan imglabels dikarenakan memiliki isi yang sama sehingga akan mendapatkan sebuah data ciri-ciri dan data jawaban sehingga bisa dikategorikan sebagai supervised learning.

```
# In[10]:

df = imgatt2.join(imglabels) #varibel df sebagai fungsi join
    dari data imgatt2 ke variabel imglabels
df = df.sample(frac=1) #variabel df sebagai sample dengan
    ketentuan frac=1
```

Hasilnya adalah seperti ini :

```
In [11]: df = imgatt2.join(imglabels)
    ...: df = df.sample(frac=1)
```

Gambar 3.18 Hasil Soal 4 - 10

- Melakukan drop pada label yang ada didepan dan akan menggunakan label yang baru di joinkan.

```
# In[11]:

df_att = df.iloc[:, :312] #membuat kolom dengan ketentuan 312
df_label = df.iloc[:, 312:] #membuat kolom dengan ketentuan
    312
```

Hasilnya adalah seperti ini :

```
In [12]: df_att = df.iloc[:, :312]
    ...: df_label = df.iloc[:, 312:]
```

Gambar 3.19 Hasil Soal 4 - 11

- Mengecek isi 5 data teratas pada df att.

```
# In[12]:

df_att.head() #menampilkan data yang di baca tadi tetapi hanya
    data paling atas
```

Hasilnya adalah seperti ini :

```
In [13]: df_att.head()
Out[13]:
         1    2    3    4    5    6    7   ...  306  307  308  309  310  311  312
                                            ...
imgid                                       ...
2449     0    0    0    0    0    0    1   ...    0    0    0    0    0    0    1
988      0    0    0    0    0    0    0   ...    0    0    0    0    0    0    0
7472     0    0    0    0    0    0    0   ...    0    0    0    0    0    0    1
7884     0    0    0    0    0    1    0   ...    0    0    0    0    1    0    0
9834     0    0    0    0    0    0    1   ...    0    0    1    0    0    0    1

[5 rows x 312 columns]
```

Gambar 3.20 Hasil Soal 4 - 12

- Mengecek isi data teratas dari df label.

```
# In[13]:

df_label.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

Hasilnya adalah seperti ini :

```
In [14]: df_label.head()
Out[14]:
       label
imgid
2449      43
988       18
7472     128
7084     121
9834     168
```

Gambar 3.21 Hasil Soal 4 - 13

- Membagi 8000 row pertama menjadi data training dan sisanya adalah data testing.

```
# In[14]:

df_train_att = df_att[:8000] #akan membagi 8000 row pertama
    menjadi data training dan sisanya adalah data testing
df_train_label = df_label[:8000] #akan membagi 8000 row
    pertama menjadi data training dan sisanya adalah data
    testing
df_test_att = df_att[8000:] #kebalikan dari akan membagi 8000
    row pertama menjadi data training dan sisanya adalah data
    testing
df_test_label = df_label[8000:] #kebalikan dari akan membagi
    8000 row pertama menjadi data training dan sisanya adalah
    data testing

df_train_label = df_train_label['label'] #menampilkan data
    training
df_test_label = df_test_label['label'] #menampilkan data
    testing
```

Hasilnya adalah seperti ini :

```
In [15]: df_train_att = df_att[:8000]
    ...: df_train_label = df_label[:8000]
    ...: df_test_att = df_att[8000:]
    ...: df_test_label = df_label[8000:]
    ...:
    ...: df_train_label = df_train_label['label']
    ...: df_test_label = df_test_label['label']
```

Gambar 3.22 Hasil Soal 4 - 14

- Pemanggilan class RandomForestClassifier. Dimana artinya menunjukkan banyak kolom pada setiap tree adalah 50.

```
# In[15]:

from sklearn.ensemble import RandomForestClassifier #import
    randomforestclassifier
clf = RandomForestClassifier(max_features=50, random_state=0,
    n_estimators=100) #clf sebagai variabel untuk klafisikasi
    random forest
```

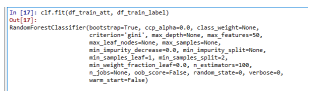
Hasilnya adalah seperti ini :

Gambar 3.23 Hasil Soal 4 - 15

- Menunjukkan hasil prediksi dari Random Forest.

```
# In[16]:

clf.fit(df_train_att, df_train_label) #variabel clf untuk fit
    yaitu training
```

Hasilnya adalah seperti ini :

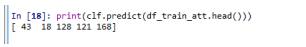


Gambar 3.24 Hasil Soal 4 - 16

- Menampilkan besaran akurasi dari prediksi pada Random Forest yang merupakan score perolehan klarifikasi.

```
# In[17]:

print(clf.predict(df_train_att.head())) #print clf yang di
    prediksi dari training tetapi hanya menampilkan data
    paling atas
```

Hasilnya adalah seperti ini :



Gambar 3.25 Hasil Soal 4 - 17

- Menampilkan besaran akurasi dari prediksi pada Random Forest yang merupakan score perolehan klarifikasi.

```
# In[18]:

clf.score(df_test_att, df_test_label) #print clf sebagai
    testing yang sudah di training tadi
```

Hasilnya adalah seperti ini :

```
In [19]: clf.score(df_test_att, df_test_label)
Out[19]: 0.4390179514255544
```

Gambar 3.26 Hasil Soal 4 - 18

5. Jalankan program confusion matrix pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Melakukan import confusion matrix pada library sklearn matriks.

```
# In[19]: Confusion Matrix

from sklearn.metrics import confusion_matrix #import Confusion
    Matrix
pred_labels = clf.predict(df_test_att) #sebagai data testing
cm = confusion_matrix(df_test_label, pred_labels) #cm sebagai
    variabel data label
```

Hasilnya adalah seperti ini :

```
In [20]: from sklearn.metrics import confusion_matrix
    ...: pred_labels = clf.predict(df_test_att)
    ...: cm = confusion_matrix(df_test_label, pred_labels)
```

Gambar 3.27 Hasil Soal 5 - 1

- Menampilkan isi cm dalam bentuk matrix yang berupa array.

```
# In[20]:

cm #menampilkan data label berbentuk array
```

Hasilnya adalah seperti ini :

```
In [21]: cm
Out[21]:
array([[ 8,  1,  1, ...,  0,  1,  0],
       [ 0, 11,  0, ...,  0,  0,  0],
       [ 0,  1,  6, ...,  0,  0,  0],
       ...,
       [ 0,  0,  0, ...,  1,  0,  0],
       [ 0,  0,  0, ...,  0,  3,  0],
       [ 0,  0,  0, ...,  0,  0, 16]], dtype=int64)
```

Gambar 3.28 Hasil Soal 5 - 2

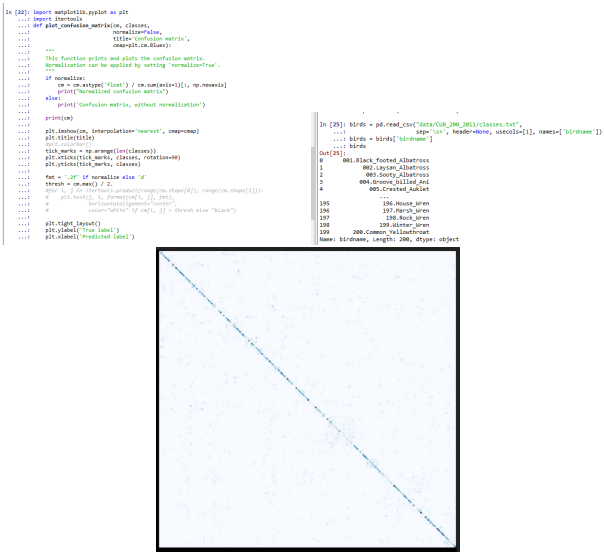
- Membuat dan menampilkan hasil plot.

```
# In[21]:

import matplotlib.pyplot as plt #import library matplotlib
    sebagai plt
import itertools #import library itertools
def plot_confusion_matrix(cm, classes,
```

```
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues): #membuat fungsi
    dengan ketentuan data yang ada pada cm

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis
    ]
        print("Normalized confusion matrix") #jika normalisasi
     sebagai ketentuan yang ada maka print normalized
    confusion matrix
    else:
        print('Confusion matrix, without normalization') #jika
     tidak memenuhi kondisi if maka pring else

    print(cm) #print data cm

    plt.imshow(cm, interpolation='nearest', cmap=cmap) #plt
    sebagai fungsi untuk membuat plot
    plt.title(title) #membuat title pada plot
    #plt.colorbar()
    tick_marks = np.arange(len(classes)) #membuat marks pada
    plot
    plt.xticks(tick_marks, classes, rotation=90) #membuat
    ticks pada x
    plt.yticks(tick_marks, classes) #membuat ticks pada y

    fmt = '.2f' if normalize else 'd' #fmt sebagai normalisasi
    thresh = cm.max() / 2.  #variabel thresh menambil data max
     pada cm kemudian dibagi 2

    plt.tight_layout() #mengatur layout pada plot
    plt.ylabel('True label') #memberi nama label pada sumbu y
    plt.xlabel('Predicted label') #memberi nama label pada
    sumbu x


# In[22]:

birds = pd.read_csv("data/CUB_200_2011/classes.txt",
                    sep='\s+', header=None, usecols=[1], names
    =['birdname']) #membaca csv dengan ketentuan nama birdname
birds = birds['birdname'] #nama birds dengan ketentuan
    birdname
birds #menampilkan data birds


# In[23]:

import numpy as np #import library numpy sebagai np
np.set_printoptions(precision=2) #np sebagai variabel yang
    membuat set precision=2
plt.figure(figsize=(60,60), dpi=300) #plot sebagai figure
    dengan ketentuan sizw 60,60 dan dpi 300
plot_confusion_matrix(cm, classes=birds, normalize=True) #data
     cm dan clas birds dibuat sebagai plot
```

```
plt.show() #menampilkan hasil plot yang berbentuk grafik
```

Hasilnya adalah seperti dibawah ini :



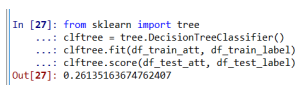
Gambar 3.29 Menampilkan hasil plot

6. Jalankan program klasifikasi SVM dan Decission Tree pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan klarifikasi dengan decission tree menggunakan dataset yang sama dan akan memunculkan akurasi prediksi.

```
# In[24]: Mencoba dengan metode Decission Tree dan SVM

from sklearn import tree #import library tree
clftree = tree.DecisionTreeClassifier() #clftree sebagai
    variabel untuk decision tree
clftree.fit(df_train_att, df_train_label) #sebagai data
    training
clftree.score(df_test_att, df_test_label) #sebagai data
    testing
```

Hasilnya adalah seperti ini :



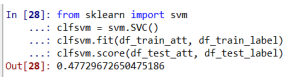
```
In [27]: from sklearn import tree
    ...: clftree = tree.DecisionTreeClassifier()
    ...: clftree.fit(df_train_att, df_train_label)
    ...: clftree.score(df_test_att, df_test_label)
Out[27]: 0.26135163674762407
```

Gambar 3.30 Hasil Soal 6 - 1

- Menunjukkan klarifikasi dengan SVM menggunakan dataset yang sama dan akan memunculkan akurasi prediksi.

```
# In[25]:

from sklearn import svm #import library svm
clfsvm = svm.SVC() #clfsvm sebagai variabel untuk mengatur
    fungsi SVC
clfsvm.fit(df_train_att, df_train_label) #sebagai data
    training
clfsvm.score(df_test_att, df_test_label) #sebagai data testing
```

Hasilnya adalah seperti ini :



Gambar 3.31 Hasil Soal 6 - 2

7. Jalankan program cross validaiton pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan hasil cross validation pada Random Forest.

```
# In[26]: Pengecekan Cross Validation

from sklearn.model_selection import cross_val_score #import
    cross_val_score
scores = cross_val_score(clf, df_train_att, df_train_label, cv
    =5) #variabel scores sebagai variabel prediksi dari data
    training
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.
    std() * 2)) #print data scores dengan ketentuan akurasi
```

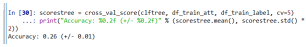
Hasilnya adalah seperti ini :

Gambar 3.32 Hasil Soal 7 - 1

- Menunjukkan hasil cross validation pada Desiccion Tree.

```
# In[27]:

scorestree = cross_val_score(clftree, df_train_att,
    df_train_label, cv=5) #sebagai prediksi menggunakan scores
     dan metode tree
print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(),
    scorestree.std() * 2)) #menampilkan dengan ketentuan yang
    ada
```

Hasilnya adalah seperti ini :

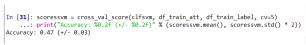


Gambar 3.33 Hasil Soal 7 - 2

- Menunjukkan hasil cross validation pada SVM.

```
# In[28]:

scoressvm = cross_val_score(clfsvm, df_train_att,
    df_train_label, cv=5) #sebagai data training
print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(),
    scoressvm.std() * 2)) #sebagai data testing dan output
    akurasi
```

Hasilnya adalah seperti ini :



Gambar 3.34 Hasil Soal 7 - 3

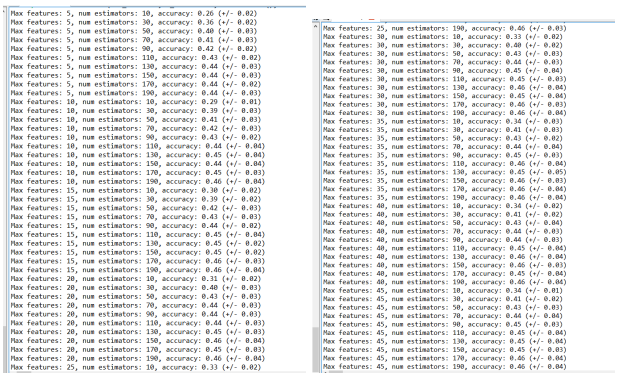
8. Jalankan program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan informasi-informasi tree yang dibuat.

```
# In[29]: Pengamatan komponen informasi

max_features_opts = range(5, 50, 5) #max_features_opts sebagai
     variabel untuk membuat range 5,50,5
n_estimators_opts = range(10, 200, 20) #n_estimators_opts
    sebagai variabel untuk membuat range 10,200,20
rf_params = np.empty((len(max_features_opts)*len(
    n_estimators_opts),4), float) #rf_params sebagai variabel
    untuk menjumlahkan yang sudah di tentukan sebelumnya
i = 0
for max_features in max_features_opts: #pengulangan
    for n_estimators in n_estimators_opts: #pengulangan
        clf = RandomForestClassifier(max_features=max_features
    , n_estimators=n_estimators) #menampilkan variabel csf
        scores = cross_val_score(clf, df_train_att,
    df_train_label, cv=5) #scores sebagai variabel training
        rf_params[i,0] = max_features #index 0
        rf_params[i,1] = n_estimators #index 1
        rf_params[i,2] = scores.mean() #index 2
        rf_params[i,3] = scores.std() * 2 #index 3
        i += 1 #dengan ketentuan i += 1
        print("Max features: %d, num estimators: %d, accuracy:
     %0.2f (+/- %0.2f)" %       (max_features, n_estimators,
    scores.mean(), scores.std() * 2))
```

```
17          #print hasil pengulangan yang sudah ditentukan
```

Hasilnya adalah seperti ini :

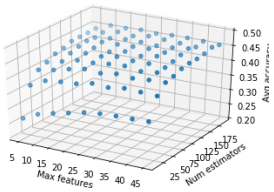


Gambar 3.35 Hasil Soal 8 - 1

- Menunjukkan hasil dari plotting komponen informasi sehingga dapat kita baca sebagai grafik 3D.

```
1  # In[30]:
2
3  import matplotlib.pyplot as plt #import library matplotlib
       sebagai plt
4  from mpl_toolkits.mplot3d import Axes3D #import axes3D untuk
       menampilkan plot 3 dimensi
5  from matplotlib import cm #memanggil data cm yang sudah
       tersedia
6  fig = plt.figure() #hasil plot sebagai figure
7  fig.clf() #figure di ambil dari clf
8  ax = fig.gca(projection='3d') #ax sebagai projection 3d
9  x = rf_params[:,0] #x sebagai index 0
10 y = rf_params[:,1] #y sebagai index 1
11 z = rf_params[:,2] #z sebagai index 2
12 ax.scatter(x, y, z) #membuat plot scatter x y z
13 ax.set_zlim(0.2, 0.5) #set zlim dengan ketentuan yang ada
14 ax.set_xlabel('Max features') #memberi nama label x
15 ax.set_ylabel('Num estimators') #memberi nama label y
16 ax.set_zlabel('Avg accuracy') #memberi nama label z
17 plt.show() #print hasil plot yang sudah dibuat.
```

Hasilnya adalah seperti ini :



Gambar 3.36 Hasil Soal 8 - 2

3.2.3 Penanganan Error

1. ScreenShoot Error

Gambar 3.37 FileNotFoundError

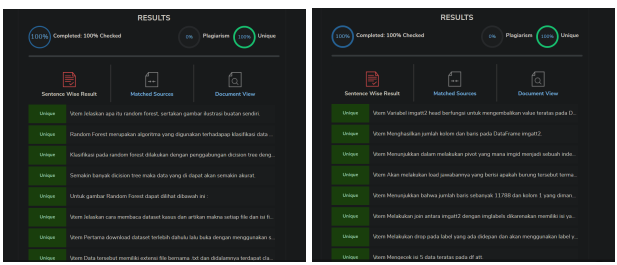
2. Tuliskan Kode Error dan Jenis Error

   - FileNotFoundError

3. Cara Penangan Error

   - FileNotFoundError
     Error terdapat pada kesalahan baca file csv, yang tidak terbaca. Dikarenakan letak file yang dibaca tidak para direktori yang sama. Seharusnya letakkan file di direktori yang sama.

3.2.4 Bukti Tidak Plagiat



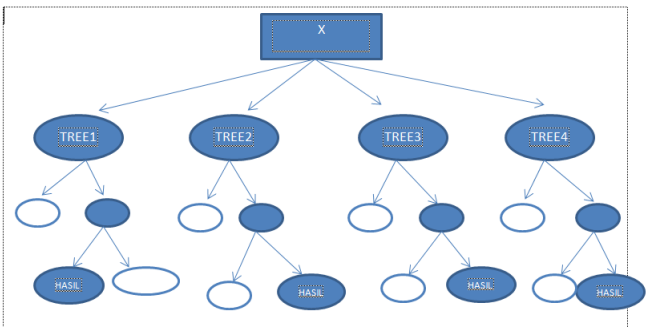
Gambar 3.38 Bukti Tidak Melakukan Plagiat Chapter 3

3.3 Muhammad Tomy 1174031

3.3.1 Teori

1. Jelaskan apa itu random forest, sertakan gambar ilustrasi buatan sendiri.

   Random Forest adalah konstruk data yang diterapkan pada machine learning yang mengembangkan sejumlah besar pohon keputusan acak yang menganalisis sekumpulan variabel. Jenis algoritma ini membantu meningkatkan cara teknologi menganalisis data yang kompleks. Juga merupakan algoritma machine learning yang fleksibel, mudah digunakan, bahkan tanpa penyetelan hyperparameter, dengan hasil yang baik. Ini juga merupakan salah satu algoritma yang paling banyak digunakan, karena kesederhanaan dan faktanya dapat digunakan untuk tugas klasifikasi dan regresi.
   Dibawah ini merupakan salah satu ilustrasi penggunaan Random Forest.



Gambar 3.39 Random Forest

2. Jelaskan cara membaca dataset khusus dan artikan makna setiap file dan isi field masing masing file. Dataset adalah kumpulan data. Paling umum satu data set sesuai dengan isi tabel database tunggal, atau matriks data statistik tunggal, di mana setiap kolom tabel mewakili variabel tertentu, dan setiap baris sesuai dengan anggota tertentu dari dataset yang dipertanyakan.

   - Gunakan librari Pandas pada python untuk dapat membaca dataset dengan format text file.
   - Setelah itu, buat variabel baru ”dataset” yang berisikan perintah untuk membaca file csv.
   - Memanggil Librari Panda untuk membaca dataset
   - Membuat variabel ”Dataset” yang berisikan pdreadcsv untuk membaca dataset. Pada contoh ini menggunakan txt tapi tetap bisa membaca datasetnya, mengapa? Karena pada saat dijalankan librari panda secara otomatis akan mengubah data dalam bentuk text file ke format csv.

3. Jelaskan apa itu Cross Validation. Cross Validation adalah prosedur resampling yang digunakan untuk mengevaluasi model machine learning pada sampel data yang terbatas. Prosedur ini memiliki parameter tunggal yang disebut k yang mengacu pada jumlah grup tempat sampel data yang akan dibagi. Karena itu, prosedur ini sering disebut k-fold cross-validation. Proses penentuan apakah hasil numerik yang mengukur hubungan yang dihipotesiskan antar variabel, dapat diterima sebagai deskripsi data, dikenal sebagai Validationi. Umumnya, estimasi kesalahan untuk model dibuat setelah training, lebih dikenal sebagai evaluasi residu. Dalam proses ini, estimasi numerik dari perbedaan respons yang diprediksi dan yang asli dilakukan, juga disebut kesalahan training. Namun, ini hanya memberi kita gambaran tentang seberapa baik model kita pada data yang digunakan untuk melatihnya. Sekarang mungkin bahwa model tersebut kurang cocok atau overfitting data. Jadi, masalah dengan teknik evaluasi ini adalah bahwa itu tidak memberikan indikasi seberapa baik pelajar akan menggeneralisasi ke set data independen / tidak terlihat. Model ini dikenal sebagai Cross Validation.

4. Jelaskan apa arti score 44 % pada random forest, 27 % pada decision tree dan 29 % dari SVM. Itu merupakan presentase keakurasian prediksi yang dilakukan pada saat testing menggunakan label pada dataset yang digunakan. Score merupakan mendefinisikan aturan evaluasi model. Maka pada saat dijalankan akan muncuk persentase tersebut yang menunjukan keakurasian atau keberhasilan dari prediksi yang dilakukan. Jika menggunakan Random Forest maka hasilnya 40% , jika menggunakan Decission Tree hasil prediksinya yaitu 27% dan pada SVM 29% .

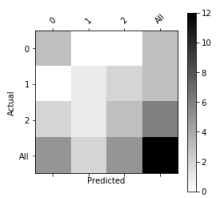
5. Jelaskan bagaimana cara membaca confusion matriks dan contohnya memakai gambar atau ilustrasi sendiri. Perthitungan Confusion Matriks dapat dilakukan sebagai berikut. Disini saya menggunakan data yang dibuat sendiri untuk menampilkan data aktual dan prediksi.

   - Import librari Pandas, Matplotlib, dan Numpy.
   - Buat variabel y actu yang berisikan data aktual.
   - Buat variabel y pred berisikan data yang akan dijadikan sebagai prediksi.
   - Buat variabel df confusion yang berisikan crosstab untuk membangun tabel tabulasi silang yang dapat menunjukkan frekuensi kemunculan kelompok data tertentu.
   - Pada variabel df confusion definisikan lagi nama baris yaitu Actual dan kolomnya Predicted
   - Kemudian definisikan suatu fungsi yang diberi nama plot confusion matrix yang berisikan pendefinisian confusion matrix dan juga akan di plotting.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 17 11:54:37 2020
4
```

```
@author: ACER
"""

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
y_actu = pd.Series([2, 0, 2, 2, 0, 1, 1, 2, 2, 0, 1, 2], name='Actual')
y_pred = pd.Series([0, 0, 2, 1, 0, 2, 1, 0, 2, 0, 2, 2], name='Predicted')
df_confusion = pd.crosstab(y_actu, y_pred)
df_confusion = pd.crosstab(y_actu, y_pred, rownames=['Actual'], colnames=['Predicted'], margins=True)
def plot_confusion_matrix(df_confusion, title='Confusion matrix', cmap=plt.cm.gray_r):
    plt.matshow(df_confusion, cmap=cmap) # imshow
    #plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(df_confusion.columns))
    plt.xticks(tick_marks, df_confusion.columns, rotation=45)
    plt.yticks(tick_marks, df_confusion.index)
    #plt.tight_layout()
    plt.ylabel(df_confusion.index.name)
    plt.xlabel(df_confusion.columns.name)
plot_confusion_matrix(df_confusion)
plt.show()
```

```
In [17]: runfile('E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan/1.py',
wdir='E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan')
```



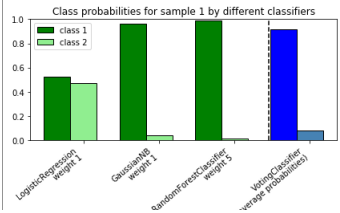
Gambar 3.40 Confusion Matriks

6. Jelaskan apa itu voting pada random forest disertai dengan ilustrasi gambar sendiri.

   Voting yaitu suara untuk setiap target yang diprediksi pada saat melakukan Random Forest. Pertimbangkan target prediksi dengan voting tertinggi sebagai prediksi akhir dari algoritma random forest.

   - Untuk menggunakan Voting pada Random Forest dapat dilihat code berikut. Disini saya mengilustrasikan voting untuk berbagai macam algoritma terutama Random Forest.

```
In [19]: runfile('E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan/src/2.py',
wdir='E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan/src')
```



Gambar 3.41 Voting

3.3.2 Praktikum

1. pandas

   pada baris ke satu yaitu perintah mengimport library padas pada python atau anaconda kemudian di inisialisasikan menjadi karakter. selanjutnya ada sebuah array yang berisi a b c d. selanjutnya penggunaan array tipe series dan yang terakhir perintah print untuk menampilkan data pada karakter.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 17 12:06:12 2020

@author: ACER
"""

import pandas as pd
data = np.array(['a','b','c','d'])
karakter = pd.Series(data)
print(karakter)
```

```
In [20]: runfile('E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan/src/3.py',
wdir='E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan/src')
0    a
1    b
2    c
3    d
dtype: object
```

Gambar 3.42 hasil

2. numpy

   Arti tiap baris codingan pada aplikasi sederhana numpy adalah sebagai berikut : pada baris ke satu yaitu mengimport numpy yang di inisialisasi menjadi np kemudian pada baris selanjutnya berisikan arange yang berarti membuat data yang berisi 12 dan ada reshape yang berfungsi merubah bentuk dari satu baris menjadi 2 baris data. Lalu yang terkhir ada perintah untuk print yaitu menampilkan data dari dika.

```
# -*- coding: utf-8 -*-
```

```
"""
Created on Tue Mar 17 12:12:39 2020

@author: ACER
"""

import numpy as np
tomy=np.arange(12).reshape(6,2)
print(tomy)
```

```
In [21]: runfile('E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan/src/4.py',
wdir='E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan/src')
[[ 0  1]
 [ 2  3]
 [ 4  5]
 [ 6  7]
 [ 8  9]
 [10 11]]
```

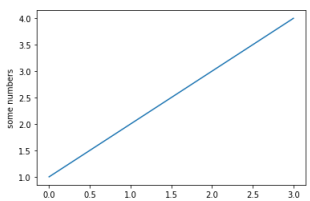
Gambar 3.43 hasil

3. matplotlib

   Arti tiap baris aplikasi sederhana matplotlib pada baris ke satu yaitu memasukan library matplotlib.pyplot yang di definisikan menjadi plt kemudian plt.plot untuk menentukan grafik yang akan dibuat. lalu membuat variabel y dengan nama some number yang terakhir untuk menampilkan data pada sebuah grafik.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 17 12:13:16 2020

@author: ACER
"""

import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```

```
In [22]: runfile('E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan/src/5.py',
wdir='E:/Kuliah/Semester 6/Kecerdasan Buatan/Tugas/git bahan/src')
```



Gambar 3.44 hasil

4. Random Forest

Arti tiap baris hasil codingan random forest pada baris pertama random forest di import dari sklearn dengan ketentuan yaitu Nilai default untuk parameter yang mengontrol ukuran pohon (mis. Max_depth, min_samples_leaf, dll.) Mengarah ke pohon yang tumbuh besar dan tidak di-unsuned yang berpotensi sangat besar pada beberapa set data. Untuk mengurangi konsumsi memori, kompleksitas dan ukuran pohon harus dikontrol dengan menetapkan nilai parameter tersebut.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 17 12:15:21 2020

@author: ACER
"""

from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
X, y = make_classification(n_samples=1000, n_features=4,
                           n_informative=2, n_redundant=0,
                           random_state=0, shuffle=False)
clf = RandomForestClassifier(max_depth=2, random_state=0)
clf.fit(X, y)
print(clf.feature_importances_)
print(clf.predict([[0, 0, 0, 0]]))
```

```
In [27]: X, y = make_classification(n_samples=1000, n_features=4,
    ...:                           n_informative=2, n_redundant=0,
    ...:                           random_state=0, shuffle=False)
    ...: clf = RandomForestClassifier(max_depth=2, random_state=0)
    ...: clf.fit(X, y)
Out[27]:
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=2, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False)

In [28]: print(clf.feature_importances_)
[0.14205973 0.76664038 0.0282433  0.06305659]

In [29]: print(clf.predict([[0, 0, 0, 0]]))
[1]
```

Gambar 3.45 hasil

5. Confusion Matrix

arti codingan pada hasil tiap codingan confusion matrix pada baris pertama codingan tersebut mendeskripsikan atau mengimport confusion matrix dari sklearn kemudian dibuat variabel y_true untuk nilai target ground truth (benar). y_pred untuk Taksiran target seperti yang dikembalikan oleh classifier. lalu menampilkan kedua variabel.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 17 12:17:15 2020

@author: ACER
```

```
"""

from sklearn.metrics import confusion_matrix
y_true = [2, 0, 2, 2, 0, 1]
y_pred = [0, 0, 2, 2, 0, 2]
confusion_matrix(y_true, y_pred)
```

```
In [35]: from sklearn.metrics import confusion_matrix
    ...: y_true = [2, 0, 2, 2, 0, 1]
    ...: y_pred = [0, 0, 2, 2, 0, 2]
    ...: confusion_matrix(y_true, y_pred)
Out[35]:
array([[2, 0, 0],
       [0, 0, 1],
       [1, 0, 2]], dtype=int64)
```

Gambar 3.46 hasil

6. SVM dan Decision Tree

   Seperti pengklasifikasi lainnya, DecisionTreeClassifier mengambil input dua array: array X, jarang atau padat, dengan ukuran n_samples, n_features memegang sampel pelatihan, dan array Y dari nilai integer, ukuran n_samples,Atau, probabilitas setiap kelas dapat diprediksi. Seperti pengklasifikasi lainnya, SVC, NuSVC dan LinearSVC mengambil input dua array: array X ukuran n_samples, n_features memegang sampel pelatihan, dan array y label kelas (string atau bilangan bulat), ukuran n_samples:

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 17 12:17:49 2020

@author: ACER
"""

from sklearn import tree
X = [[0, 0], [1, 1]]
Y = [0, 1]
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, Y)
clf.predict([[2., 2.]])

from sklearn import svm
X = [[0, 0], [1, 1]]
y = [0, 1]
clf = svm.SVC()
clf.fit(X, y)
clf.predict([[2., 2.]])
```

7. Cross Validation

   digunakan untuk memeriksa akurasi dari ketepatan hasil pengolahan data tersebut maka akan didapat nilai rata-rata 60 persen dari hasil pengolahan data tersebut untuk lebih jelasnya dapat di lihat pada gambar pada codingan tersebut pada

```
In [40]: from sklearn import svm
    ...: X = [[0, 0], [1, 1]]
    ...: y = [0, 1]
    ...: clf = svm.SVC()
    ...: clf.fit(X, y)
    ...: clf.predict([[2., 2.]])
Out[40]: array([1])

In [41]: from sklearn import tree
    ...: X = [[0, 0], [1, 1]]
    ...: Y = [0, 1]
    ...: clf = tree.DecisionTreeClassifier()
    ...: clf = clf.fit(X, Y)
    ...: clf.predict([[2., 2.]])
Out[41]: array([1])
```

Gambar 3.47 hasil

baris ke satu melakukan import library dari sklern kemudian pada baris selanjutnya mengisi nilai skor dengan nilai pada variabel lontong setelah hal tersebut dilakukan kemudian data tersebut di eksekusi. berikut merupakan hasil dari code tersebut dapat dilihat pada gambar.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 17 12:18:20 2020

@author: ACER
"""

from sklearn.model_selection import cross_val_score
scores = cross_val_score(lontong, pecel_att, pecel_pass, cv=5)
# show average score and +/- two standard deviations away
#(covering 95% of scores)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std
    () * 2))
```

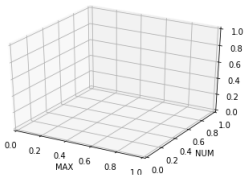
```
In [61]: runfile('D:/SEMESTER 6/wert/9.py', wdir='D:/SEMESTER 6/wert')
Accuracy: 0.59 (+/- 0.05)
```

Gambar 3.48 hasil

8. program pengamatan arti dari hasil program pengamatan. perogram pengamatan dapat mengamati dari 3 aspek diatas yaitu svm, random, dan decision tree. Yang memiliki variabel X Y Z dan di tampilkan dalam bentuk grafik.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 17 12:19:01 2020

@author: ACER
"""

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
fig = plt.figure()
```

```
fig.clf()
ax = fig.gca(projection='3d')
plt.xlabel('MAX')
plt.ylabel('NUM')
plt.zlabel('AVG')
ax.scatter(x, y, z)
ax.set_zlim(0.2, 0.5)
ax.set_xlabel('Max features')
ax.set_ylabel('Num estimator')
ax.set_zlabel('Avg accuracy')
plt.show()
```



Gambar 3.49 hasil

3.3.3 Penanganan Error

Screenshot error

1. Untuk gambar screenshot error

```
  File "<ipython-input-60-d14a3944647a>", line 1, in <module>
    runfile('D:/SEMESTER 6/wert/1/5.py', wdir='D:/SEMESTER 6/wert/1')

  File "C:\Users\User\Anaconda3\lib\site-packages\spyder_kernels\customize
\spydercustomize.py", line 827, in runfile
    execfile(filename, namespace)

  File "C:\Users\User\Anaconda3\lib\site-packages\spyder_kernels\customize
\spydercustomize.py", line 110, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "D:/SEMESTER 6/wert/1/5.py", line 10
    plt.ylabel(some numbers)
                    ^
SyntaxError: invalid syntax
```

Gambar 3.50 hasil

Code Errornya

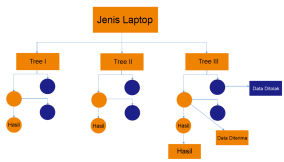
```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel(some numbers)
plt.show()
```

pada kode ylabel memiliki nama atau isi some numbers tetapi pada tipe data tertentu harus di awali dan diakhiri dengan tanda petik 2 atau 1. Pada kodingnya hanya kurang tanda petik 1.

3.4 1174005 - Oniwaldus Bere Mali

3.4.1 Soal Teori

1. Jelaskan apa itu random forest, sertakan gambar ilustrasi buatan sendiri.
   Random Forest merupakan algoritma yang digunakan terhadapap klasifikasi data dalam jumlah yang besar. Klasifikasi pada random forest dilakukan dengan penggabungan dicision tree dengan melakuakn training terhadap sempel data yang dimiliki. Semakin banyak dicision tree maka data yang di dapat akan semakin akurat. Untuk gambar Random Forest dapat dilihat dibawah ini :



Gambar 3.51 Random Forest.

2. Jelaskan cara membaca dataset kasus dan artikan makna setiap file dan isi field masing-masing file.

   - Pertama download dataset terlebih dahulu lalu buka dengan menggunakan software spyder guna melihat isi dari dataset tersebut.
   - Data tersebut memiliki extensi file bernama .txt dan didalamnya terdapat class dari field.
   - Misalnya saja pada data jenis burung memiliki file index dan angka, dimana index berisi angka yang memiliki makna berupa jenis burung.
   - bahkan nama burung sedangkan field memiliki isi nilai berupa 0 dan 1 yang dimana sifatnya boolean atau Ya dan Tidak.
   - Hal ini dikarenakan komputer hanya dapat membaca bilangan biner maka dari itu field yang di isikan berupa angka.
   - Artinya angka 0 berarti tidak dan angka 1 berarti Ya.

3. Jelaskan apa itu cross validation.
   Cross Validation merupakan sebuah teknik validasi model yang berfungsi untuk melakukan penilaian bagaimana hasil analisis statistik akan digeneralisasi ke

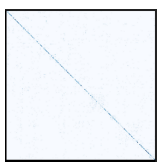
data yang lebih independen. Cross validation digunakan dengan tujuan prediksi, dan bila kita ingin memperkirakan seberapa akurat model model prediksi yang dilakukan dalam sebuah praktek. Tujuan dari cross validation yaitu untuk mendefinisikan dataset guna menguju dalam fase pelatihan untuk membatasi masalah seperti overfitting dan underfitting serta mendapatkan wawasan tentang bagaimana model akan digeneralisasikan ke set data independen.

4. Jelaskan apa arti score 44 % pada random forest, 27% pada decission tree dan 29 % dari SVM.
   Dimana Score 44 % diperoleh dari hasil pengelohan dataset jenis burung. Dimana akan dilakukan proses pembagian data testing dan data training lalu diproses dan menghasilkan score sebanyak 44 % dimana menjelaskan bahwa score tersebut digunakan sebagai pembanding dalam tingkat keakuratannya. Pada dicision tree akan memperoleh data lebih kecil yaitu sebanyak 27 % hal ini dikarenakan data yang diolah menggunakan dicision tree dibagi menjadi beberapa tree dan lalu disimpulkan untuk mendapatkan data yang akurat. Pada SVM akan memperoleh score sebanyak 29 % hal ini dikarenakan data yang dimiliki masih bernilai netral sehingga tingkat keakuratannya masih belum jelas.

5. Jelaskan bagaimana cara membaca confusion matriks dan contohnya memakai gambar atau ilustrasi sendiri.
   Untuk membaca confusion matriks dapat menggunakan source code sebagai berikut :

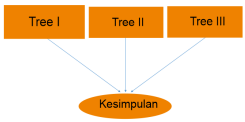
```
from matplotlib import cm #memanggil data cm yang sudah tersedia
fig = plt.figure() #hasil plot sebagai figure
fig.clf() #figure di ambil dari clf
ax = fig.gca(projection='3d') #ax sebagai projection 3d
x = rf_params[:,0] #x sebagai index 0
```

   Dimana numpy akan mengurus semua data yang berhubungan dengan matrix. Pada source code tersebut digunakan dalam melakukan read pada dataset burung dengan menggunakan metode confusion matrix. Dalam confusion matrix memiliki 4 istilah yaitu True Positive yang merupakan data posotif yang terditeksi benar, True Negatif yang merupakan data negatif akan tetapi terditeksi benar, False Positif merupakan data negatif namun terditeksi sebagai data positif, False Negatif merupakan data posotif namun terditeksi sebagai data negatif. Adapun contoh hasil read dataset menggunakan confusion matrix dapat dilihat dibawah ini :



Gambar 3.52 Confusion Matriks.

6. Jelaskan apa itu voting pada random forest disertai dengan ilustrasi gambar sendiri
   Voting pada random forest ialah sebuah proses pemilihan dari tree yang dimana akan dimunculkan hasilnya dan disimpulkan menjadi informasi yang pasti. Untuk lebih jelasnya saya akan memberikan sebuah contoh bagaimana voting bekerja :



Gambar 3.53 Decision Tree.

Dimana ditunjukkan pada gambar diatas terdapat 3 tree. Dalam tree tersebut akan dilakukan proses voting. Saya akan memberikan contoh kasus, dimana akan diadakan voting untuk menentukan sebuah laptop. Dalam tree akan diberikan sejumlah data misalnya saja data tersebut berupa gambar, yang dimana data tersebut akan dipilih dengan cara voting. Hasil voting akhir dari setiap tree menunjukkan laptop asus, yang berarti kesimpulan dari data yang telah diberikan menyatakan gambar tersebut adalah laptop asus. Bagaimana apabila terjadi perbedaan data misalnya saja pada tree 1 dan 2 menunyatakan laptop asus sedangkan pada tree 3 menyatakan laptop HP, maka kesimpulan yang di ambil adalah laptop asus dikarenakan hasil voting terbanyak adalah laptop asus.

3.4.2 Praktek Program

1. Buat aplikasi sederhana menggunakan pandas dan jelaskan arti setiap baris kode yang dibuat(harus beda dengan teman sekelas).

```
# In[44]: Soal1

import pandas as oni #melakukan import pada library pandas
    sebagai oni

laptop = {"Nama Laptop" : ['Asus','HP','Lenovo','Samsung']} #
    membuat varibel yang bernama laptop, dan mengisi dataframe
    nama2 laptop
```

Kode di atas digunakan untuk mengimpor atau mengirim library pandas sebagai fahmi, Hasilnya adalah sebagai berikut :

```
                Nama Laptop
0     Fahmi Punya Laptop Asus
1       Fahmi Punya Laptop HP
2   Fahmi Punya Laptop Lenovo
3  Fahmi Punya Laptop Samsung
```

Gambar 3.54 Hasil Soal 1.

2. buat aplikasi sederhana menggunakan numpy dan jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas)

```
import numpy as oni
matrix_x = oni.eye(10) #membuat matrix dengan numpy dengan
    menggunakan fungsi eye
matrix_x #deklrasikan matrix_x yang telah dibuat

print (matrix_x) #print matrix_x yang telah dibuat dengan 10x10
```

Fungsi eye disini berguna untuk memanggil matrix identitas dengan jumlah colom dan baris sesuai yang ditentukan. Disini saya telah menentukan 10 colom dan baris maka dari itu hasilnya akan memunculkan matrix identitas 10x10, Hasilnya adalah sebagai berikut :

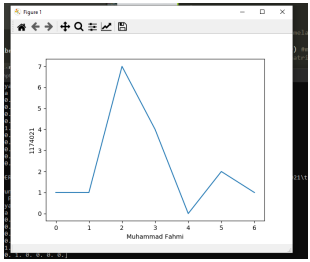
```
[[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]]
```

Gambar 3.55 Hasil Soal 2.

3. buat aplikasi sederhana menggunakan matplotlib dan jelaskan arti dari setiap baris kode(harus beda dengan teman sekelas)

```

oni.plot([1,1,7,4,0,0,5]) #memberikan nilai plot atau grafik pada
    fahmi
oni.xlabel('Oniwaldus Bere Mali') #memberikan label pada x
oni.ylabel('1174021') #memberikan label pada y
oni.show() #print hasil plot berbentuk grafik
```

Jalankan source code diatas dengan spyder atau anaconda sehingga hasilnya akan seperti berikut :



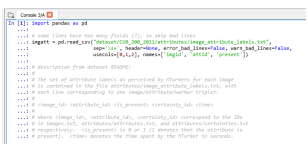
Gambar 3.56 Hasil Soal 3.

4. jalankan program klasifikasi Random Forest pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

   - Source Code pertama pada random forest berfungsi untuk membaca dataset yang memiliki format text file dengan mendefinisikan variabel yang bernama imgatt. Variabel tersebut berisi value untuk membaca data.

```
# In[1]: RANDOM FOREST

import pandas as pd #import library pandas sebagai as

imgatt = pd.read_csv("D:/KULIAH/SEMESTER 6/Kecerdasan Buatan/Chapter 3/src/1174005/dataset/CUB_200_2011/attributes/image_attribute_labels.txt",
                     sep='\s+', header=None, error_bad_lines=False, warn_bad_lines=False,
                     usecols=[0,1,2], names=['imgid', 'attid', 'present']) #library imgatt sebagai membaca file csv dari dataset, dengan ketentuan yang ada.
```

   Hasilnya adalah seperti ini :



Gambar 3.57 Hasil Soal 4 - 1

   - Pada source code berikutnya akan mengembalikan baris teratas dari DataFrame variabel imgatt.

```
# In[2]:

imgatt.head() #menampilkan data yang di baca tadi tetapi hanya data paling atas
```

Hasilnya adalah seperti ini :

```
In [2]: imgatt.head()
Out[2]:
   imgid  attid  present
0      1      1        0
1      1      2        0
2      1      3        0
3      1      4        0
4      1      5        1
```

Gambar 3.58 Hasil Soal 4 - 2

- Pada output berikutnya akan menampilkan jumlah kolom dan baris dari DataFrame imgatt.

```
# In[3]:

imgatt.shape #menampilkan jumlah data, kolom
```

Hasilnya adalah seperti ini :

```
In [3]: imgatt.shape
Out[3]: (3677856, 3)
```

Gambar 3.59 Hasil Soal 4 - 3

- Variabel imgatt2 telah menggunakan function yang bernama pivot guna mengubah kolom jadi baris dan sebaliknya dari DataFrame sebelumnya.

```
# In[4]:

imgatt2 = imgatt.pivot(index='imgid', columns='attid', values=
    'present') #membuat variabel baru dari fungsi imgatt,
    dengan mengganti index dan kolom (kebalikan)
```

Hasilnya adalah seperti ini :

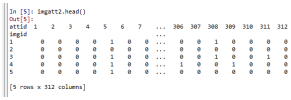
```
In [4]: imgatt2 = imgatt.pivot(index='imgid', columns='attid', values='present')
```

Gambar 3.60 Hasil Soal 4 - 4

- Variabel imgatt2 head berfungsi untuk mengembalikan value teratas pada DataFrame imgatt2.

```
# In[5]:

imgatt2.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

Hasilnya adalah seperti ini :

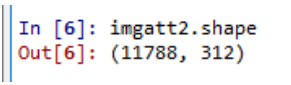


Gambar 3.61 Hasil Soal 4 - 5

- Menghasilkan jumlah kolom dan baris pada DataFrame imgatt2.

```
# In[6]:

imgatt2.shape #menampilkan jumlah data, kolom
```

Hasilnya adalah seperti ini :

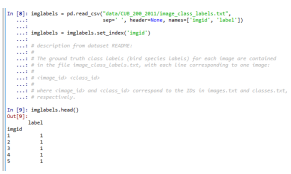


Gambar 3.62 Hasil Soal 4 - 6

- Menunjukkan dalam melakukan pivot yang mana imgid menjadi sebuah index yang unik.

```
# In[7]:

imglabels = pd.read_csv("D:/KULIAH/SEMESTER 6/Kecerdasan
    Buatan/Chapter 3/src/1174005/dataset/CUB_200_2011/
    image_class_labels.txt",
                        sep=' ', header=None, names=['imgid',
    'label']) #baca data csv dengan ketentuan yang ada

imglabels = imglabels.set_index('imgid') #variabel imglabels
    sebagai set index (imgid)
```

Hasilnya adalah seperti ini :

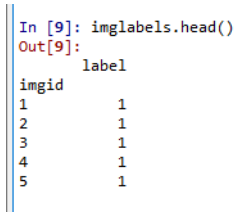


Gambar 3.63 Hasil Soal 4 - 7

- Akan melakukan load jawabannya yang berisi apakah burung tersebut termasuk spesies yang mana. Kolom tersebut yaitu imgid dan label.

```
# In[8]:

imglabels.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

Hasilnya adalah seperti ini :

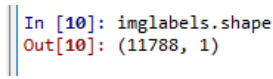


Gambar 3.64 Hasil Soal 4 - 8

- Menunjukkan bahwa jumlah baris sebanyak 11788 dan kolom 1 yang dimana kolom tersebut adalah jenis spesies pada burung.

```
# In[9]:

imglabels.shape #menampilkan jumlah data, kolom
```

Hasilnya adalah seperti ini :

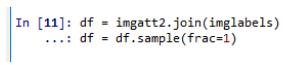


Gambar 3.65 Hasil Soal 4 - 9

- Melakukan join antara imgatt2 dengan imglabels dikarenakan memiliki isi yang sama sehingga akan mendapatkan sebuah data ciri-ciri dan data jawaban sehingga bisa dikategorikan sebagai supervised learning.

```
# In[10]:

df = imgatt2.join(imglabels) #varibel df sebagai fungsi join
    dari data imgatt2 ke variabel imglabels
df = df.sample(frac=1) #variabel df sebagai sample dengan
    ketentuan frac=1
```

Hasilnya adalah seperti ini :



Gambar 3.66 Hasil Soal 4 - 10

- Melakukan drop pada label yang ada didepan dan akan menggunakan label yang baru di joinkan.

```
# In[11]:

df_att = df.iloc[:, :312] #membuat kolom dengan ketentuan 312
df_label = df.iloc[:, 312:] #membuat kolom dengan ketentuan
    312
```

Hasilnya adalah seperti ini :

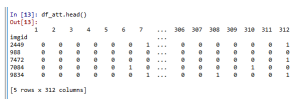
```
In [12]: df_att = df.iloc[:, :312]
    ...: df_label = df.iloc[:, 312:]
```

Gambar 3.67 Hasil Soal 4 - 11

- Mengecek isi 5 data teratas pada df att.

```
# In[12]:

df_att.head() #menampilkan data yang di baca tadi tetapi hanya
    data paling atas
```

Hasilnya adalah seperti ini :



Gambar 3.68 Hasil Soal 4 - 12

- Mengecek isi data teratas dari df label.

```
# In[13]:

df_label.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

Hasilnya adalah seperti ini :

```
In [14]: df_label.head()
Out[14]:
        label
imgid
2449       43
988        18
7472      128
7084      121
9834      168
```

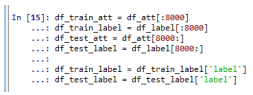
Gambar 3.69 Hasil Soal 4 - 13

- Membagi 8000 row pertama menjadi data training dan sisanya adalah data testing.

```
# In[14]:

df_train_att = df_att[:8000] #akan membagi 8000 row pertama
    menjadi data training dan sisanya adalah data testing
df_train_label = df_label[:8000] #akan membagi 8000 row
    pertama menjadi data training dan sisanya adalah data
    testing
```

```
df_test_att = df_att[8000:] #kebalikan dari akan membagi 8000 row pertama menjadi data training dan sisanya adalah data testing
df_test_label = df_label[8000:] #kebalikan dari akan membagi 8000 row pertama menjadi data training dan sisanya adalah data testing

df_train_label = df_train_label['label'] #menampilkan data training
df_test_label = df_test_label['label'] #menampilkan data testing
```

Hasilnya adalah seperti ini :



Gambar 3.70 Hasil Soal 4 - 14

- Pemanggilan class RandomForestClassifier. Dimana artinya menunjukkan banyak kolom pada setiap tree adalah 50.

```
# In[15]:

from sklearn.ensemble import RandomForestClassifier #import randomforestclassifier
clf = RandomForestClassifier(max_features=50, random_state=0, n_estimators=100) #clf sebagai variabel untuk klafisikasi random forest
```

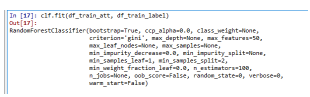
Hasilnya adalah seperti ini :

Gambar 3.71 Hasil Soal 4 - 15

- Menunjukkan hasil prediksi dari Random Forest.

```
# In[16]:

clf.fit(df_train_att, df_train_label) #variabel clf untuk fit yaitu training
```

Hasilnya adalah seperti ini :



Gambar 3.72 Hasil Soal 4 - 16

- Menampilkan besaran akurasi dari prediksi pada Random Forest yang merupakan score perolehan klarifikasi.

```
# In[17]:

print(clf.predict(df_train_att.head())) #print clf yang di
    prediksi dari training tetapi hanya menampilkan data
    paling atas
```

Hasilnya adalah seperti ini :

```
In [18]: print(clf.predict(df_train_att.head()))
[ 43  18 128 121 168]
```

Gambar 3.73 Hasil Soal 4 - 17

- Menampilkan besaran akurasi dari prediksi pada Random Forest yang merupakan score perolehan klarifikasi.

```
# In[18]:

clf.score(df_test_att, df_test_label) #print clf sebagai
    testing yang sudah di training tadi
```

Hasilnya adalah seperti ini :

```
In [19]: clf.score(df_test_att, df_test_label)
Out[19]: 0.4390179514255544
```

Gambar 3.74 Hasil Soal 4 - 18

5. Jalankan program confusion matrix pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Melakukan import confusion matrix pada library sklearn matriks.

```
# In[19]: Confusion Matrix

from sklearn.metrics import confusion_matrix #import Confusion
    Matrix
pred_labels = clf.predict(df_test_att) #sebagai data testing
cm = confusion_matrix(df_test_label, pred_labels) #cm sebagai
    variabel data label
```

Hasilnya adalah seperti ini :

```
In [20]: from sklearn.metrics import confusion_matrix
    ...: pred_labels = clf.predict(df_test_att)
    ...: cm = confusion_matrix(df_test_label, pred_labels)
```

Gambar 3.75 Hasil Soal 5 - 1

- Menampilkan isi cm dalam bentuk matrix yang berupa array.

```
# In[20]:

cm #menampilkan data label berbentuk array
```

Hasilnya adalah seperti ini :

```
In [21]: cm
Out[21]:
array([[ 8,  1,  1, ...,  0,  1,  0],
       [ 0, 11,  0, ...,  0,  0,  0],
       [ 0,  1,  6, ...,  0,  0,  0],
       ...,
       [ 0,  0,  0, ...,  1,  0,  0],
       [ 0,  0,  0, ...,  0,  3,  0],
       [ 0,  0,  0, ...,  0,  0, 16]], dtype=int64)
```

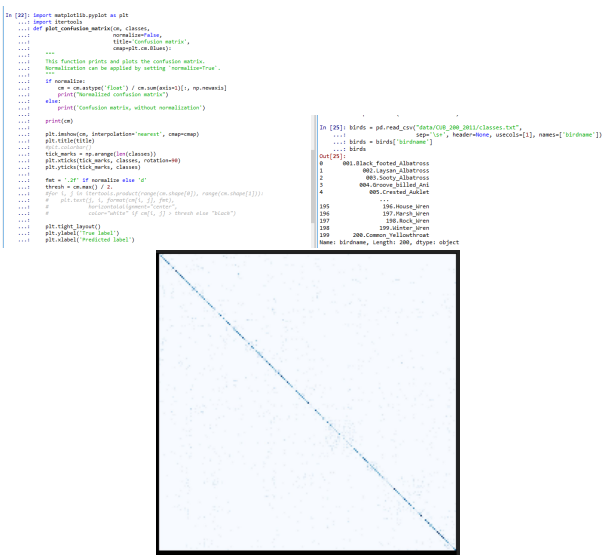
Gambar 3.76 Hasil Soal 5 - 2

- Membuat dan menampilkan hasil plot.

```

# In[21]:

import matplotlib.pyplot as plt #import library matplotlib
    sebagai plt
import itertools #import library itertools
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues): #membuat fungsi
    dengan ketentuan data yang ada pada cm

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis
    ]
        print("Normalized confusion matrix") #jika normalisasi
     sebagai ketentuan yang ada maka print normalized
    confusion matrix
    else:
        print('Confusion matrix, without normalization') #jika
     tidak memenuhi kondisi if maka pring else

    print(cm) #print data cm

    plt.imshow(cm, interpolation='nearest', cmap=cmap) #plt
    sebagai fungsi untuk membuat plot
    plt.title(title) #membuat title pada plot
    #plt.colorbar()
    tick_marks = np.arange(len(classes)) #membuat marks pada
    plot
    plt.xticks(tick_marks, classes, rotation=90) #membuat
    ticks pada x
    plt.yticks(tick_marks, classes) #membuat ticks pada y

    fmt = '.2f' if normalize else 'd' #fmt sebagai normalisasi
    thresh = cm.max() / 2.  #variabel thresh menambil data max
     pada cm kemudian dibagi 2

    plt.tight_layout() #mengatur layout pada plot
    plt.ylabel('True label') #memberi nama label pada sumbu y
```

```
    plt.xlabel('Predicted label') #memberi nama label pada
    sumbu x


# In[22]:

birds = pd.read_csv("data/CUB_200_2011/classes.txt",
                    sep='\s+', header=None, usecols=[1], names
    =['birdname']) #membaca csv dengan ketentuan nama birdname
birds = birds['birdname'] #nama birds dengan ketentuan
    birdname
birds #menampilkan data birds


# In[23]:

import numpy as np #import library numpy sebagai np
np.set_printoptions(precision=2) #np sebagai variabel yang
    membuat set precision=2
plt.figure(figsize=(60,60), dpi=300) #plot sebagai figure
    dengan ketentuan sizw 60,60 dan dpi 300
plot_confusion_matrix(cm, classes=birds, normalize=True) #data
     cm dan clas birds dibuat sebagai plot
```

Hasilnya adalah seperti dibawah ini :



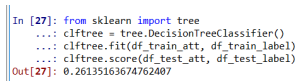
Gambar 3.77 Menampilkan hasil plot

6. Jalankan program klasifikasi SVM dan Decission Tree pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan klarifikasi dengan decission tree menggunakan dataset yang sama dan akan memunculkan akurasi prediksi.

```
# In[24]: Mencoba dengan metode Decission Tree dan SVM

from sklearn import tree #import library tree
clftree = tree.DecisionTreeClassifier() #clftree sebagai
    variabel untuk decision tree
clftree.fit(df_train_att, df_train_label) #sebagai data
    training
```

Hasilnya adalah seperti ini :

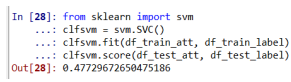


Gambar 3.78 Hasil Soal 6 - 1

- Menunjukkan klarifikasi dengan SVM menggunakan dataset yang sama dan akan memunculkan akurasi prediksi.

```
# In[25]:

from sklearn import svm #import library svm
clfsvm = svm.SVC() #clfsvm sebagai variabel untuk mengatur
    fungsi SVC
clfsvm.fit(df_train_att, df_train_label) #sebagai data
    training
```

Hasilnya adalah seperti ini :



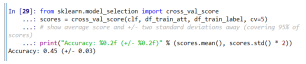
Gambar 3.79 Hasil Soal 6 - 2

7. Jalankan program cross validaiton pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan hasil cross validation pada Random Forest.

```
# In[26]: Pengecekan Cross Validation

from sklearn.model_selection import cross_val_score #import
    cross_val_score
scores = cross_val_score(clf, df_train_att, df_train_label, cv
    =5) #variabel scores sebagai variabel prediksi dari data
    training
```

Hasilnya adalah seperti ini :

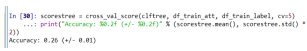


Gambar 3.80 Hasil Soal 7 - 1

- Menunjukkan hasil cross validation pada Desiccion Tree.

```
1
2 # In[27]:
3
4 scorestree = cross_val_score(clftree, df_train_att,
      df_train_label, cv=5) #sebagai prediksi menggunakan scores
       dan metode tree
```

Hasilnya adalah seperti ini :

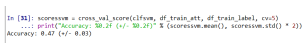


Gambar 3.81 Hasil Soal 7 - 2

- Menunjukkan hasil cross validation pada SVM.

```
1
2 # In[28]:
3
4 scoressvm = cross_val_score(clfsvm, df_train_att,
      df_train_label, cv=5) #sebagai data training
```

Hasilnya adalah seperti ini :



Gambar 3.82 Hasil Soal 7 - 3

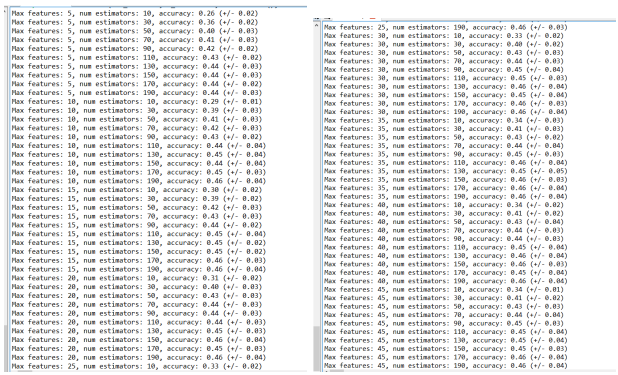
8. Jalankan program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan informasi-informasi tree yang dibuat.

```
1
2 # In[29]: Pengamatan komponen informasi
3
4 max_features_opts = range(5, 50, 5) #max_features_opts sebagai
       variabel untuk membuat range 5,50,5
5 n_estimators_opts = range(10, 200, 20) #n_estimators_opts
      sebagai variabel untuk membuat range 10,200,20
```

```
rf_params = np.empty((len(max_features_opts)*len(
    n_estimators_opts),4), float) #rf_params sebagai variabel
    untuk menjumlahkan yang sudah di tentukan sebelumnya
i = 0
for max_features in max_features_opts: #pengulangan
    for n_estimators in n_estimators_opts: #pengulangan
        clf = RandomForestClassifier(max_features=max_features
    , n_estimators=n_estimators) #menampilkan variabel csf
        scores = cross_val_score(clf, df_train_att,
    df_train_label, cv=5) #scores sebagai variabel training
        rf_params[i,0] = max_features #index 0
        rf_params[i,1] = n_estimators #index 1
        rf_params[i,2] = scores.mean() #index 2
        rf_params[i,3] = scores.std() * 2 #index 3
        i += 1 #dengan ketentuan i += 1
        print("Max features: %d, num estimators: %d, accuracy:
     %0.2f (+/- %0.2f)" %        (max_features, n_estimators,
    scores.mean(), scores.std() * 2))
```

Hasilnya adalah seperti ini :



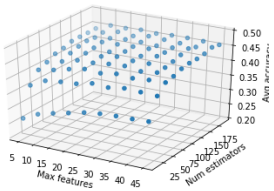
Gambar 3.83 Hasil Soal 8 - 1

- Menunjukkan hasil dari plotting komponen informasi sehingga dapat kita baca sebagai grafik 3D.

```

# In[30]:

import matplotlib.pyplot as plt #import library matplotlib
    sebagai plt
from mpl_toolkits.mplot3d import Axes3D #import axes3D untuk
    menampilkan plot 3 dimensi
from matplotlib import cm #memanggil data cm yang sudah
    tersedia
fig = plt.figure() #hasil plot sebagai figure
fig.clf() #figure di ambil dari clf
ax = fig.gca(projection='3d') #ax sebagai projection 3d
x = rf_params[:,0] #x sebagai index 0
y = rf_params[:,1] #y sebagai index 1
```

```
z = rf_params[:,2] #z sebagai index 2
ax.scatter(x, y, z) #membuat plot scatter x y z
ax.set_zlim(0.2, 0.5) #set zlim dengan ketentuan yang ada
ax.set_xlabel('Max features') #memberi nama label x
ax.set_ylabel('Num estimators') #memberi nama label y
ax.set_zlabel('Avg accuracy') #memberi nama label z
```

Hasilnya adalah seperti ini :



Gambar 3.84 Hasil Soal 8 - 2

3.4.3 Penanganan Error

1. ScreenShoot Error

Gambar 3.85 FileNotFoundError

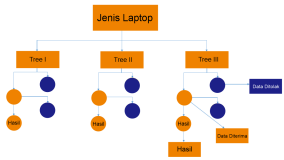
2. Tuliskan Kode Error dan Jenis Error
   - FileNotFoundError
3. Cara Penangan Error
   - FileNotFoundError
     Error terdapat pada kesalahan baca file csv, yang tidak terbaca. Dikarenakan letak file yang dibaca tidak para direktori yang sama. Seharusnya letakkan file di direktori yang sama.

3.5 1174027 - Harun Ar - Rasyid

3.5.1 Teori

1. Jelaskan apa itu random forest, sertakan gambar ilustrasi buatan sendiri.
   Random Forest merupakan algoritma yang digunakan terhadapap klasifikasi data dalam jumlah yang besar. Klasifikasi pada random forest dilakukan dengan penggabungan dicision tree dengan melakuakn training terhadap sempel

data yang dimiliki. Semakin banyak dicision tree maka data yang di dapat akan semakin akurat. Untuk gambar Random Forest dapat dilihat dibawah ini :



Gambar 3.86 Random Forest.

2. Jelaskan cara membaca dataset kasus dan artikan makna setiap file dan isi field masing-masing file.

   - Pertama download dataset terlebih dahulu lalu buka dengan menggunakan software spyder guna melihat isi dari dataset tersebut.
   - Data tersebut memiliki extensi file bernama .txt dan didalamnya terdapat class dari field.
   - Misalnya saja pada data jenis burung memiliki file index dan angka, dimana index berisi angka yang memiliki makna berupa jenis burung.
   - bahkan nama burung sedangkan field memiliki isi nilai berupa 0 dan 1 yang dimana sifatnya boolean atau Ya dan Tidak.
   - Hal ini dikarenakan komputer hanya dapat membaca bilangan biner maka dari itu field yang di isikan berupa angka.
   - Artinya angka 0 berarti tidak dan angka 1 berarti Ya.

3. Jelaskan apa itu cross validation.
   Cross Validation merupakan sebuah teknik validasi model yang berfungsi untuk melakukan penilaian bagaimana hasil analisis statistik akan digeneralisasi ke data yang lebih independen. Cross validation digunakan dengan tujuan prediksi, dan bila kita ingin memperkirakan seberapa akurat model model prediksi yang dilakukan dalam sebuah praktek. Tujuan dari cross validation yaitu untuk mendefinisikan dataset guna menguju dalam fase pelatihan untuk membatasi masalah seperti overfitting dan underfitting serta mendapatkan wawasan tentang bagaimana model akan digeneralisasikan ke set data independen.

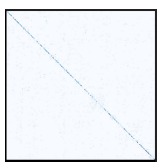
4. Jelaskan apa arti score 44 % pada random forest, 27% pada decission tree dan 29 % dari SVM.
   Dimana Score 44 % diperoleh dari hasil pengelohan dataset jenis burung. Dimana akan dilakukan proses pembagian data testing dan data training lalu diproses dan menghasilkan score sebanyak 44 % dimana menjelaskan bahwa score tersebut digunakan sebagai pembanding dalam tingkat keakuratannya. Pada dicision

tree akan memperoleh data lebih kecil yaitu sebanyak 27 % hal ini dikarenakan data yang diolah menggunakan dicision tree dibagi menjadi beberapa tree dan lalu disimpulkan untuk mendapatkan data yang akurat. Pada SVM akan memperoleh score sebanyak 29 % hal ini dikarenakan data yang dimiliki masih bernilai netral sehingga tingkat keakuratannya masih belum jelas.

5. Jelaskan bagaimana cara membaca confusion matriks dan contohnya memakai gambar atau ilustrasi sendiri.
   Untuk membaca confusion matriks dapat menggunakan source code sebagai berikut :

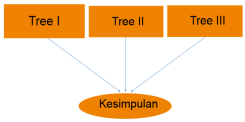
```
fig = plt.figure() #hasil plot sebagai figure
fig.clf() #figure di ambil dari clf
ax = fig.gca(projection='3d') #ax sebagai projection 3d
x = rf_params[:,0] #x sebagai index 0
y = rf_params[:,1] #y sebagai index 1
```

   Dimana numpy akan mengurus semua data yang berhubungan dengan matrix. Pada source code tersebut digunakan dalam melakukan read pada dataset burung dengan menggunakan metode confusion matrix. Dalam confusion matrix memiliki 4 istilah yaitu True Positive yang merupakan data posotif yang terditeksi benar, True Negatif yang merupakan data negatif akan tetapi terditeksi benar, False Positif merupakan data negatif namun terditeksi sebagai data positif, False Negatif merupakan data posotif namun terditeksi sebagai data negatif. Adapun contoh hasil read dataset menggunakan confusion matrix dapat dilihat dibawah ini :



Gambar 3.87 Confusion Matriks.

6. Jelaskan apa itu voting pada random forest disertai dengan ilustrasi gambar sendiri
   Voting pada random forest ialah sebuah proses pemilihan dari tree yang dimana akan dimunculkan hasilnya dan disimpulkan menjadi informasi yang pasti. Untuk lebih jelasnya saya akan memberikan sebuah contoh bagaimana voting bekerja :



Gambar 3.88 Decision Tree.

Dimana ditunjukkan pada gambar diatas terdapat 3 tree. Dalam tree tersebut akan dilakukan proses voting. Saya akan memberikan contoh kasus, dimana akan diadakan voting untuk menentukan sebuah laptop. Dalam tree akan diberikan sejumlah data misalnya saja data tersebut berupa gambar, yang dimana data tersebut akan dipilih dengan cara voting. Hasil voting akhir dari setiap tree menunjukkan laptop asus, yang berarti kesimpulan dari data yang telah diberikan menyatakan gambar tersebut adalah laptop asus. Bagaimana apabila terjadi perbedaan data misalnya saja pada tree 1 dan 2 menunyatakan laptop asus sedangkan pada tree 3 menyatakan laptop HP, maka kesimpulan yang di ambil adalah laptop asus dikarenakan hasil voting terbanyak adalah laptop asus.

3.5.2 Praktek

1. Buat aplikasi sederhana menggunakan pandas dan jelaskan arti setiap baris kode yang dibuat(harus beda dengan teman sekelas).

```
import pandas as pd #digunakan untuk import library pandas
harun = pd.read_csv('dataku.csv',sep=';') #digunakan untuk
    memanggil file csv dan dipisahkan dengan ;
len(harun) #untuk mengetahui jumlah data
print(harun.nama) #digunakan untuk mengetahui kolom nama dari
    varivel harun
```

Kode di atas digunakan untuk mengimpor atau mengirim library pandas sebagai pd, Hasilnya adalah sebagai berikut :

```
0          Harun Ar - Rasyid
1    Evietania charis sujadi
2             Nico Sembiring
3                  Magdalena
4               Luigi Intan
5               Nadya Rahma
6         Habib Abdul Rasyid
7        Etika Khusnul Laeli
8           Tomy Nur Maulidi
```

Gambar 3.89 Hasil Soal 1.

2. buat aplikasi sederhana menggunakan numpy dan jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas)

```
import numpy as np #digunakan untuk import library numpy
totumur = np.sum(harun.umur) # menggunakan fungsi sum bawaan dari
    numpy
print(totumur) #menampilkan isi variabel umur
```

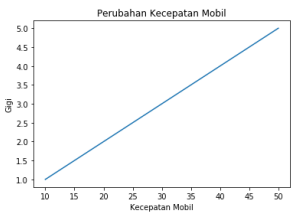
Fungsi sum disini berguna untuk menjumlahkan data. Hasilnya adalah sebagai berikut :

Gambar 3.90 Hasil Soal 2.

3. buat aplikasi sederhana menggunakan matplotlib dan jelaskan arti dari setiap baris kode(harus beda dengan teman sekelas)

```
import matplotlib.pyplot as mt #import matploblib
mt.plot([10,20,30,40,50],[1,2,3,4,5]) #digunakan untuk membuat
    fungsi graph
mt.xlabel("Kecepatan Mobil") #membuat label untuk garis x
mt.ylabel("Gigi") # membuat label untuk y
mt.title("Perubahan Kecepatan Mobil") #membuat judul graph
mt.show() # menampilkan graph
```

Jalankan source code diatas dengan spyder atau anaconda sehingga hasilnya akan seperti berikut :



Gambar 3.91 Hasil Soal 3.

4. jalankan program klasifikasi Random Forest pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Source Code pertama pada random forest berfungsi untuk membaca dataset yang memiliki format text file dengan mendefinisikan variabel yang bernama imgatt. Variabel tersebut berisi value untuk membaca data.

```
import pandas as pd #import library pandas sebagai as

imgatt = pd.read_csv("data/CUB_200_2011/attributes/
    image_attribute_labels.txt",
                     sep='\s+', header=None, error_bad_lines=
    False, warn_bad_lines=False,
                     usecols=[0,1,2], names=['imgid', 'attid',
     'present']) #library imgatt sebagai membaca file csv dari
     dataset, dengan ketentuan yang ada.
```

- Pada source code berikutnya akan mengembalikan baris teratas dari DataFrame variabel imgatt.

```
imgatt.head() #menampilkan data yang di baca tadi tetapi hanya
     data paling atas
```

- Pada output berikutnya akan menampilkan jumlah kolom dan baris dari DataFrame imgatt.

```
imgatt.shape #menampilkan jumlah data, kolom
```

- Variabel imgatt2 telah menggunakan function yang bernama pivot guna mengubah kolom jadi baris dan sebaliknya dari DataFrame sebelumnya.

```
imgatt2 = imgatt.pivot(index='imgid', columns='attid', values=
    'present') #membuat variabel baru dari fungsi imgatt,
    dengan mengganti index dan kolom (kebalikan)
```

- Variabel imgatt2 head berfungsi untuk mengembalikan value teratas pada DataFrame imgatt2.

```
imgatt2.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

- Menghasilkan jumlah kolom dan baris pada DataFrame imgatt2.

```
imgatt2.shape #menampilkan jumlah data, kolom
```

- Menunjukkan dalam melakukan pivot yang mana imgid menjadi sebuah index yang unik.

```
imglabels = pd.read_csv("data/CUB_200_2011/image_class_labels.
    txt",
                        sep=' ', header=None, names=['imgid',
    'label']) #baca data csv dengan ketentuan yang ada

imglabels = imglabels.set_index('imgid') #variabel imglabels
    sebagai set index (imgid)
```

- Akan melakukan load jawabannya yang berisi apakah burung tersebut termasuk spesies yang mana. Kolom tersebut yaitu imgid dan label.

```
imglabels.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

- Menunjukkan bahwa jumlah baris sebanyak 11788 dan kolom 1 yang dimana kolom tersebut adalah jenis spesies pada burung.

```
imglabels.shape #menampilkan jumlah data, kolom
```

- Melakukan join antara imgatt2 dengan imglabels dikarenakan memiliki isi yang sama sehingga akan mendapatkan sebuah data ciri-ciri dan data jawaban sehingga bisa dikategorikan sebagai supervised learning.

```
df = imgatt2.join(imglabels) #varibel df sebagai fungsi join
    dari data imgatt2 ke variabel imglabels
df = df.sample(frac=1) #variabel df sebagai sample dengan
    ketentuan frac=1
```

- Melakukan drop pada label yang ada didepan dan akan menggunakan label yang baru di joinkan.

```
df_att = df.iloc[:, :312] #membuat kolom dengan ketentuan 312
df_label = df.iloc[:, 312:] #membuat kolom dengan ketentuan
    312
```

- Mengecek isi 5 data teratas pada df att.

```
df_att.head() #menampilkan data yang di baca tadi tetapi hanya
     data paling atas
```

- Mengecek isi data teratas dari df label.

```
df_label.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

- Membagi 8000 row pertama menjadi data training dan sisanya adalah data testing.

```
df_train_att = df_att[:8000] #akan membagi 8000 row pertama
    menjadi data training dan sisanya adalah data testing
df_train_label = df_label[:8000] #akan membagi 8000 row
    pertama menjadi data training dan sisanya adalah data
    testing
df_test_att = df_att[8000:] #kebalikan dari akan membagi 8000
    row pertama menjadi data training dan sisanya adalah data
    testing
df_test_label = df_label[8000:] #kebalikan dari akan membagi
    8000 row pertama menjadi data training dan sisanya adalah
    data testing

df_train_label = df_train_label['label'] #menampilkan data
    training
df_test_label = df_test_label['label'] #menampilkan data
    testing
```

- Pemanggilan class RandomForestClassifier. Dimana artinya menunjukkan banyak kolom pada setiap tree adalah 50.

```
from sklearn.ensemble import RandomForestClassifier #import
    randomforestclassifier
clf = RandomForestClassifier(max_features=50, random_state=0,
    n_estimators=100) #clf sebagai variabel untuk klafisikasi
    random forest
```

- Menunjukkan hasil prediksi dari Random Forest.

```
clf.fit(df_train_att, df_train_label) #variabel clf untuk fit
    yaitu training
```

- Menampilkan besaran akurasi dari prediksi pada Random Forest yang merupakan score perolehan klarifikasi.

```
print(clf.predict(df_train_att.head())) #print clf yang di
    prediksi dari training tetapi hanya menampilkan data
    paling atas
```

5. Jalankan program confusion matrix pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Melakukan import confusion matrix pada library sklearn matriks.

```
from sklearn.metrics import confusion_matrix #import Confusion
    Matrix
pred_labels = clf.predict(df_test_att) #sebagai data testing
cm = confusion_matrix(df_test_label, pred_labels) #cm sebagai
    variabel data label
```

- Menampilkan isi cm dalam bentuk matrix yang berupa array.

```
cm #menampilkan data label berbentuk array
```

- Membuat dan menampilkan hasil plot.

```
import matplotlib.pyplot as plt #import library matplotlib
    sebagai plt
import itertools #import library itertools
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues): #membuat fungsi
    dengan ketentuan data yang ada pada cm

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis
    ]
        print("Normalized confusion matrix") #jika normalisasi
     sebagai ketentuan yang ada maka print normalized
    confusion matrix
    else:
        print('Confusion matrix, without normalization') #jika
     tidak memenuhi kondisi if maka pring else

```

```
    print(cm) #print data cm

    plt.imshow(cm, interpolation='nearest', cmap=cmap) #plt
    sebagai fungsi untuk membuat plot
    plt.title(title) #membuat title pada plot
    #plt.colorbar()
    tick_marks = np.arange(len(classes)) #membuat marks pada
    plot
    plt.xticks(tick_marks, classes, rotation=90) #membuat
    ticks pada x
    plt.yticks(tick_marks, classes) #membuat ticks pada y

    fmt = '.2f' if normalize else 'd' #fmt sebagai normalisasi
    thresh = cm.max() / 2.  #variabel thresh menambil data max
     pada cm kemudian dibagi 2

    plt.tight_layout() #mengatur layout pada plot
    plt.ylabel('True label') #memberi nama label pada sumbu y
    plt.xlabel('Predicted label') #memberi nama label pada
    sumbu x


# In[22]:

birds = pd.read_csv("data/CUB_200_2011/classes.txt",
                    sep='\s+', header=None, usecols=[1], names
    =['birdname']) #membaca csv dengan ketentuan nama birdname
birds = birds['birdname'] #nama birds dengan ketentuan
    birdname
birds #menampilkan data birds


# In[23]:

import numpy as np #import library numpy sebagai np
np.set_printoptions(precision=2) #np sebagai variabel yang
    membuat set precision=2
plt.figure(figsize=(60,60), dpi=300) #plot sebagai figure
    dengan ketentuan sizw 60,60 dan dpi 300
plot_confusion_matrix(cm, classes=birds, normalize=True) #data
     cm dan clas birds dibuat sebagai plot
```

6. Jalankan program klasifikasi SVM dan Decission Tree pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

   - Menunjukkan klarifikasi dengan decission tree menggunakan dataset yang sama dan akan memunculkan akurasi prediksi.

```
from sklearn import tree #import library tree
clftree = tree.DecisionTreeClassifier() #clftree sebagai
    variabel untuk decision tree
clftree.fit(df_train_att, df_train_label) #sebagai data
    training
```

```
clftree.score(df_test_att, df_test_label) #sebagai data
    testing
```

- Menunjukkan klarifikasi dengan SVM menggunakan dataset yang sama dan akan memunculkan akurasi prediksi.

```
from sklearn import svm #import library svm
clfsvm = svm.SVC() #clfsvm sebagai variabel untuk mengatur
    fungsi SVC
clfsvm.fit(df_train_att, df_train_label) #sebagai data
    training
clfsvm.score(df_test_att, df_test_label) #sebagai data testing
```

7. Jalankan program cross validaiton pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan hasil cross validation pada Random Forest.

```
from sklearn.model_selection import cross_val_score #import
    cross_val_score
scores = cross_val_score(clf, df_train_att, df_train_label, cv
    =5) #variabel scores sebagai variabel prediksi dari data
    training
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.
    std() * 2)) #print data scores dengan ketentuan akurasi
```

- Menunjukkan hasil cross validation pada Desiccion Tree.

```
scorestree = cross_val_score(clftree, df_train_att,
    df_train_label, cv=5) #sebagai prediksi menggunakan scores
     dan metode tree
print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(),
    scorestree.std() * 2)) #menampilkan dengan ketentuan yang
    ada
```

- Menunjukkan hasil cross validation pada SVM.

```
scoressvm = cross_val_score(clfsvm, df_train_att,
    df_train_label, cv=5) #sebagai data training
print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(),
    scoressvm.std() * 2)) #sebagai data testing dan output
    akurasi
```

8. Jalankan program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan informasi-informasi tree yang dibuat.

```
max_features_opts = range(5, 50, 5) #max_features_opts sebagai
     variabel untuk membuat range 5,50,5
n_estimators_opts = range(10, 200, 20) #n_estimators_opts
    sebagai variabel untuk membuat range 10,200,20
rf_params = np.empty((len(max_features_opts)*len(
    n_estimators_opts),4), float) #rf_params sebagai variabel
    untuk menjumlahkan yang sudah di tentukan sebelumnya
i = 0
for max_features in max_features_opts: #pengulangan
    for n_estimators in n_estimators_opts: #pengulangan
        clf = RandomForestClassifier(max_features=max_features
    , n_estimators=n_estimators) #menampilkan variabel csf
        scores = cross_val_score(clf, df_train_att,
    df_train_label, cv=5) #scores sebagai variabel training
        rf_params[i,0] = max_features #index 0
        rf_params[i,1] = n_estimators #index 1
        rf_params[i,2] = scores.mean() #index 2
        rf_params[i,3] = scores.std() * 2 #index 3
        i += 1 #dengan ketentuan i += 1
        print("Max features: %d, num estimators: %d, accuracy:
     %0.2f (+/- %0.2f)" %       (max_features, n_estimators,
    scores.mean(), scores.std() * 2))
```

- Menunjukkan hasil dari plotting komponen informasi sehingga dapat kita baca sebagai grafik 3D.

```
import matplotlib.pyplot as plt #import library matplotlib
    sebagai plt
from mpl_toolkits.mplot3d import Axes3D #import axes3D untuk
    menampilkan plot 3 dimensi
from matplotlib import cm #memanggil data cm yang sudah
    tersedia
fig = plt.figure() #hasil plot sebagai figure
fig.clf() #figure di ambil dari clf
ax = fig.gca(projection='3d') #ax sebagai projection 3d
x = rf_params[:,0] #x sebagai index 0
y = rf_params[:,1] #y sebagai index 1
z = rf_params[:,2] #z sebagai index 2
ax.scatter(x, y, z) #membuat plot scatter x y z
ax.set_zlim(0.2, 0.5) #set zlim dengan ketentuan yang ada
ax.set_xlabel('Max features') #memberi nama label x
ax.set_ylabel('Num estimators') #memberi nama label y
ax.set_zlabel('Avg accuracy') #memberi nama label z
plt.show() #print hasil plot yang sudah dibuat.
```

3.5.3 Penanganan Error

1. ScreenShoot Error

Gambar 3.92 FileNotFoundError

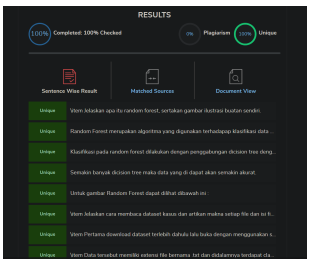
2. Tuliskan Kode Error dan Jenis Error

- FileNotFoundError

3. Cara Penangan Error

   - FileNotFoundError
     Error terdapat pada kesalahan baca file csv, yang tidak terbaca. Dikarenakan letak file yang dibaca tidak para direktori yang sama. Seharusnya letakkan file di direktori yang sama.

3.5.4 Bukti Tidak Plagiat



Gambar 3.93 FileNotFoundError

3.6 1174008 - Arjun Yuda Firwanda

3.6.1 Soal Teori

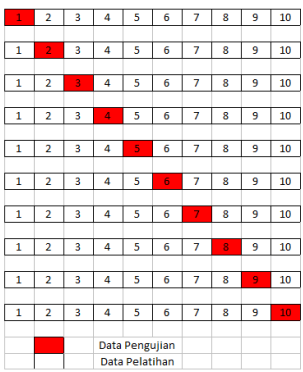
1. Jelaskan apa itu random forest, sertakan gambar ilustrasi buatan sendiri.
   Random Forest merupakan suatu metode penyelesaian permasalahan. Metode Random Forest sebagai sebutan metode Decision Tree atau metode yang menyerupai pohon pengambilan keputusan sebuah diagram air yang memiliki sebuah root node yang digunakan untuk mengumpulkan suatu data.

Gambar 3.94 Random Forest.

2. Jelaskan cara membaca dataset kasus dan artikan makna setiap file dan isi field masing-masing file.

   - Pertama download dataset terlebih dahulu lalu buka dengan menggunakan software spyder guna melihat isi dari dataset tersebut.
   - Data tersebut memiliki extensi file bernama .txt dan didalamnya terdapat class dari field.
   - Misalnya saja pada data jenis burung memiliki file index dan angka, dimana index berisi angka yang memiliki makna berupa jenis burung.
   - bahkan nama burung sedangkan field memiliki isi nilai berupa 0 dan 1 yang dimana sifatnya boolean atau Ya dan Tidak.
   - Hal ini dikarenakan komputer hanya dapat membaca bilangan biner maka dari itu field yang di isikan berupa angka.
   - Artinya angka 0 berarti tidak dan angka 1 berarti Ya.

3. Jelaskan apa itu cross validation.
   Cross Validation suatu metode yang menerapkan statistik yang biasanya dapat digunakan untuk menilai kinerja model serta algoritma, Pada data tersebut dapat dipisahkan menjadi dua bagian subset, seperti data proses pembelajaran dan data validasi atau evaluasi. Selain itu juga Cross Validation ini dapat digunakan dikarenakan mampu mengurangi waktu komputasi dengan tetap serta dapat menjaga keakuratan estimasi.

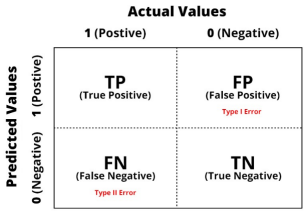


Gambar 3.95 Random Forest.

4. Jelaskan apa arti score 44 % pada random forest, 27% pada decission tree dan 29 % dari SVM.
   Dimana Score 44 % diperoleh dari hasil pengelohan dataset jenis burung. Dimana akan dilakukan proses pembagian data testing dan data training lalu diproses

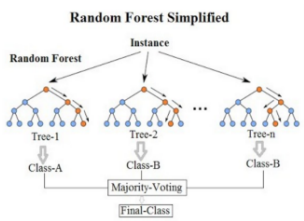
dan menghasilkan score sebanyak 44 % dimana menjelaskan bahwa score tersebut digunakan sebagai pembanding dalam tingkat keakuratannya. Pada dicision tree akan memperoleh data lebih kecil yaitu sebanyak 27 % hal ini dikarenakan data yang diolah menggunakan dicision tree dibagi menjadi beberapa tree dan lalu disimpulkan untuk mendapatkan data yang akurat. Pada SVM akan memperoleh score sebanyak 29 % hal ini dikarenakan data yang dimiliki masih bernilai netral sehingga tingkat keakuratannya masih belum jelas.

5. Jelaskan bagaimana cara membaca confusion matriks dan contohnya memakai gambar atau ilustrasi sendiri.
   Confusion Matrix atau error matrix yang dapat memberikan informasi perbandingan hasil klasifikasi sebenarnya. Confusion Matrix berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya telah diketahui.



Gambar 3.96 Random Forest.

6. Jelaskan apa itu voting pada random forest disertai dengan ilustrasi gambar sendiri
   Voting pada random forest ialah sebuah proses pemilihan dari tree yang dimana akan dimunculkan hasilnya dan disimpulkan menjadi informasi yang pasti. Penentuan klasifikasi menguunakan metode random forest diambil berdasarkan hasil voting dari decision tree yang terbentuk. Setelah decision tree atau pohon terbentuk, maka akan dilakukan voting pada setiap kelas dari data sampel. Kemudian, selanjutnya mengkombinasikan vote dari setiap kelas kemudian diambil vote yang paling banyak. Dengan menggunakan metode penyelesaian random forest pada klasifikasi data maka, akan menghasilkan vote yang paling baik.



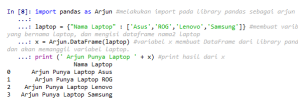
Gambar 3.97 Random Forest.

3.6.2 Praktek Program

1. Buat aplikasi sederhana menggunakan pandas dan jelaskan arti setiap baris kode yang dibuat(harus beda dengan teman sekelas).

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 17 10:34:35 2020

@author: Arjun
```

Kode di atas digunakan untuk mengimpor atau mengirim library pandas sebagai Arjun, Hasilnya adalah sebagai berikut :

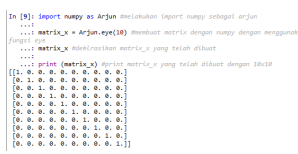


Gambar 3.98 Hasil Soal 1.

2. buat aplikasi sederhana menggunakan numpy dan jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas)

```

laptop = {"Nama Laptop" : ['Asus','ROG','Lenovo','Samsung']} #
    membuat varibel yang bernama laptop, dan mengisi dataframe
    nama2 laptop
x = Arjun.DataFrame(laptop) #variabel x membuat DataFrame dari
    library pandas dan akan memanggil variabel laptop.
print (' Arjun Punya Laptop ' + x) #print hasil dari x

# In[44]: Soal2
```

Fungsi eye disini berguna untuk memanggil matrix identitas dengan jumlah colom dan baris sesuai yang ditentukan. Disini saya telah menentukan 10 colom dan baris maka dari itu hasilnya akan memunculkan matrix identitas 10x10, Hasilnya adalah sebagai berikut :

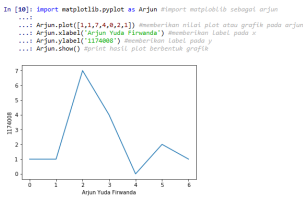


Gambar 3.99 Hasil Soal 2.

3. buat aplikasi sederhana menggunakan matplotlib dan jelaskan arti dari setiap baris kode(harus beda dengan teman sekelas)

```
matrix_x #deklrasikan matrix_x yang telah dibuat

print (matrix_x) #print matrix_x yang telah dibuat dengan 10x10


# In[44]: Soal3
```

Jalankan source code diatas dengan spyder atau anaconda sehingga hasilnya akan seperti berikut :



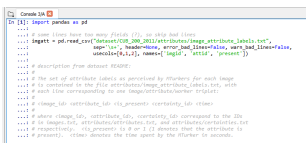
Gambar 3.100 Hasil Soal 3.

4. jalankan program klasifikasi Random Forest pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

   - Source Code pertama pada random forest berfungsi untuk membaca dataset yang memiliki format text file dengan mendefinisikan variabel yang bernama imgatt. Variabel tersebut berisi value untuk membaca data.

```
# In[1]: RANDOM FOREST

import pandas as pd #import library pandas sebagai as

imgatt = pd.read_csv("data/CUB_200_2011/attributes/
    image_attribute_labels.txt",
                     sep='\s+', header=None, error_bad_lines=
    False, warn_bad_lines=False,
                     usecols=[0,1,2], names=['imgid', 'attid',
     'present']) #library imgatt sebagai membaca file csv dari
     dataset, dengan ketentuan yang ada.
```

Hasilnya adalah seperti ini :



Gambar 3.101 Hasil Soal 4 - 1

- Pada source code berikutnya akan mengembalikan baris teratas dari DataFrame variabel imgatt.

```
# In[2]:

imgatt.head() #menampilkan data yang di baca tadi tetapi hanya
    data paling atas
```

Hasilnya adalah seperti ini :

```
In [2]: imgatt.head()
Out[2]:
   imgid  attid  present
0      1      1        0
1      1      2        0
2      1      3        0
3      1      4        0
4      1      5        1
```

Gambar 3.102 Hasil Soal 4 - 2

- Pada output berikutnya akan menampilkan jumlah kolom dan baris dari DataFrame imgatt.

```
# In[3]:

imgatt.shape #menampilkan jumlah data, kolom
```

Hasilnya adalah seperti ini :

```
In [3]: imgatt.shape
Out[3]: (3677856, 3)
```

Gambar 3.103 Hasil Soal 4 - 3

- Variabel imgatt2 telah menggunakan function yang bernama pivot guna mengubah kolom jadi baris dan sebaliknya dari DataFrame sebelumnya.

```
# In[4]:

imgatt2 = imgatt.pivot(index='imgid', columns='attid', values=
    'present') #membuat variabel baru dari fungsi imgatt,
    dengan mengganti index dan kolom (kebalikan)
```

Hasilnya adalah seperti ini :

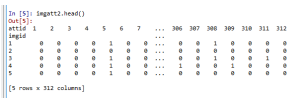
```
In [4]: imgatt2 = imgatt.pivot(index='imgid', columns='attid', values='present')
```

Gambar 3.104 Hasil Soal 4 - 4

- Variabel imgatt2 head berfungsi untuk mengembalikan value teratas pada DataFrame imgatt2.

```
# In[5]:

imgatt2.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

Hasilnya adalah seperti ini :

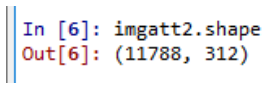


Gambar 3.105 Hasil Soal 4 - 5

- Menghasilkan jumlah kolom dan baris pada DataFrame imgatt2.

```
# In[6]:

imgatt2.shape #menampilkan jumlah data, kolom
```

Hasilnya adalah seperti ini :

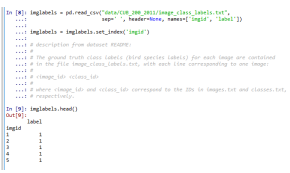


Gambar 3.106 Hasil Soal 4 - 6

- Menunjukkan dalam melakukan pivot yang mana imgid menjadi sebuah index yang unik.

```
# In[7]:

imglabels = pd.read_csv("data/CUB_200_2011/image_class_labels.
    txt",
                        sep=' ', header=None, names=['imgid',
    'label']) #baca data csv dengan ketentuan yang ada

imglabels = imglabels.set_index('imgid') #variabel imglabels
    sebagai set index (imgid)
```

Hasilnya adalah seperti ini :



Gambar 3.107 Hasil Soal 4 - 7

- Akan melakukan load jawabannya yang berisi apakah burung tersebut termasuk spesies yang mana. Kolom tersebut yaitu imgid dan label.

```
# In[8]:

imglabels.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

Hasilnya adalah seperti ini :

```
In [9]: imglabels.head()
Out[9]:
       label
imgid
1          1
2          1
3          1
4          1
5          1
```

Gambar 3.108 Hasil Soal 4 - 8

- Menunjukkan bahwa jumlah baris sebanyak 11788 dan kolom 1 yang dimana kolom tersebut adalah jenis spesies pada burung.

```
# In[9]:

imglabels.shape #menampilkan jumlah data, kolom
```

Hasilnya adalah seperti ini :

```
In [10]: imglabels.shape
Out[10]: (11788, 1)
```

Gambar 3.109 Hasil Soal 4 - 9

- Melakukan join antara imgatt2 dengan imglabels dikarenakan memiliki isi yang sama sehingga akan mendapatkan sebuah data ciri-ciri dan data jawaban sehingga bisa dikategorikan sebagai supervised learning.

```
# In[10]:

df = imgatt2.join(imglabels) #varibel df sebagai fungsi join
    dari data imgatt2 ke variabel imglabels
df = df.sample(frac=1) #variabel df sebagai sample dengan
    ketentuan frac=1
```

Hasilnya adalah seperti ini :

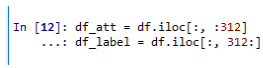
```
In [11]: df = imgatt2.join(imglabels)
    ...: df = df.sample(frac=1)
```

Gambar 3.110 Hasil Soal 4 - 10

- Melakukan drop pada label yang ada didepan dan akan menggunakan label yang baru di joinkan.

```
# In[11]:

df_att = df.iloc[:, :312] #membuat kolom dengan ketentuan 312
df_label = df.iloc[:, 312:] #membuat kolom dengan ketentuan
    312
```

Hasilnya adalah seperti ini :

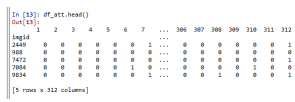


Gambar 3.111 Hasil Soal 4 - 11

- Mengecek isi 5 data teratas pada df att.

```
# In[12]:

df_att.head() #menampilkan data yang di baca tadi tetapi hanya
    data paling atas
```

Hasilnya adalah seperti ini :

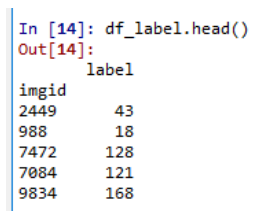


Gambar 3.112 Hasil Soal 4 - 12

- Mengecek isi data teratas dari df label.

```
# In[13]:

df_label.head() #menampilkan data yang di baca tadi tetapi
    hanya data paling atas
```

Hasilnya adalah seperti ini :



Gambar 3.113 Hasil Soal 4 - 13

- Membagi 8000 row pertama menjadi data training dan sisanya adalah data testing.

```
# In[14]:

df_train_att = df_att[:8000] #akan membagi 8000 row pertama menjadi data training dan sisanya adalah data testing
df_train_label = df_label[:8000] #akan membagi 8000 row pertama menjadi data training dan sisanya adalah data testing
df_test_att = df_att[8000:] #kebalikan dari akan membagi 8000 row pertama menjadi data training dan sisanya adalah data testing
df_test_label = df_label[8000:] #kebalikan dari akan membagi 8000 row pertama menjadi data training dan sisanya adalah data testing

df_train_label = df_train_label['label'] #menampilkan data training
df_test_label = df_test_label['label'] #menampilkan data testing
```

Hasilnya adalah seperti ini :

```
In [15]: df_train_att = df_att[:8000]
    ...: df_train_label = df_label[:8000]
    ...: df_test_att = df_att[8000:]
    ...: df_test_label = df_label[8000:]
    ...:
    ...: df_train_label = df_train_label['label']
    ...: df_test_label = df_test_label['label']
```

Gambar 3.114 Hasil Soal 4 - 14

- Pemanggilan class RandomForestClassifier. Dimana artinya menunjukkan banyak kolom pada setiap tree adalah 50.

```
# In[15]:

from sklearn.ensemble import RandomForestClassifier #import randomforestclassifier
clf = RandomForestClassifier(max_features=50, random_state=0, n_estimators=100) #clf sebagai variabel untuk klafisikasi random forest
```

Hasilnya adalah seperti ini :

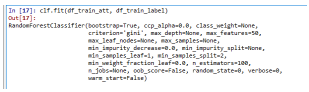
```
In [16]: from sklearn.ensemble import RandomForestClassifier
    ...: clf = RandomForestClassifier(max_features=50, random_state=0, n_estimators=100)
```

Gambar 3.115 Hasil Soal 4 - 15

- Menunjukkan hasil prediksi dari Random Forest.

```
# In[16]:

clf.fit(df_train_att, df_train_label) #variabel clf untuk fit yaitu training
```

Hasilnya adalah seperti ini :

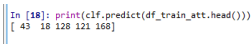


Gambar 3.116 Hasil Soal 4 - 16

- Menampilkan besaran akurasi dari prediksi pada Random Forest yang merupakan score perolehan klarifikasi.

```
# In[17]:

print(clf.predict(df_train_att.head())) #print clf yang di
    prediksi dari training tetapi hanya menampilkan data
    paling atas
```

Hasilnya adalah seperti ini :

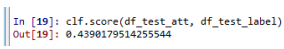


Gambar 3.117 Hasil Soal 4 - 17

- Menampilkan besaran akurasi dari prediksi pada Random Forest yang merupakan score perolehan klarifikasi.

```
# In[18]:

clf.score(df_test_att, df_test_label) #print clf sebagai
    testing yang sudah di training tadi
```

Hasilnya adalah seperti ini :



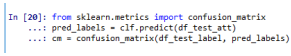
Gambar 3.118 Hasil Soal 4 - 18

5. Jalankan program confusion matrix pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Melakukan import confusion matrix pada library sklearn matriks.

```
# In[19]: Confusion Matrix

from sklearn.metrics import confusion_matrix #import Confusion
    Matrix
pred_labels = clf.predict(df_test_att) #sebagai data testing
cm = confusion_matrix(df_test_label, pred_labels) #cm sebagai
    variabel data label
```

Hasilnya adalah seperti ini :

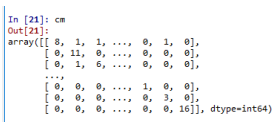


Gambar 3.119 Hasil Soal 5 - 1

- Menampilkan isi cm dalam bentuk matrix yang berupa array.

```
# In[20]:

cm #menampilkan data label berbentuk array
```

Hasilnya adalah seperti ini :



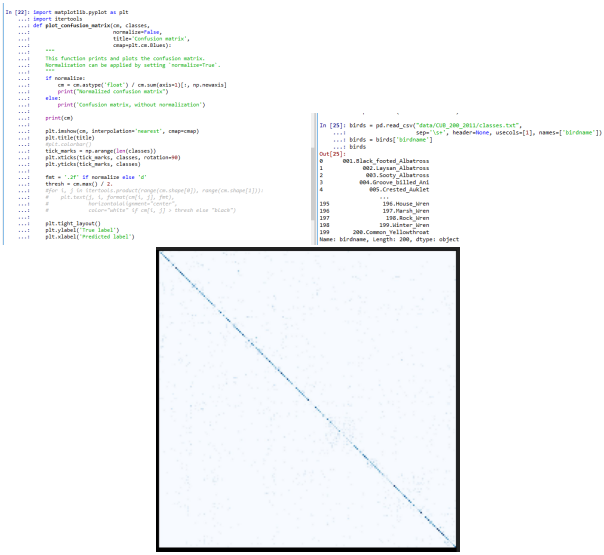
Gambar 3.120 Hasil Soal 5 - 2

- Membuat dan menampilkan hasil plot.

```
# In[21]:

import matplotlib.pyplot as plt #import library matplotlib
    sebagai plt
import itertools #import library itertools
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues): #membuat fungsi
    dengan ketentuan data yang ada pada cm

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis
    ]
        print("Normalized confusion matrix") #jika normalisasi
     sebagai ketentuan yang ada maka print normalized
    confusion matrix
    else:
        print('Confusion matrix, without normalization') #jika
     tidak memenuhi kondisi if maka pring else

    print(cm) #print data cm

    plt.imshow(cm, interpolation='nearest', cmap=cmap) #plt
    sebagai fungsi untuk membuat plot
    plt.title(title) #membuat title pada plot
    #plt.colorbar()
    tick_marks = np.arange(len(classes)) #membuat marks pada
    plot
```

```
22      plt.xticks(tick_marks, classes, rotation=90) #membuat
        ticks pada x
23      plt.yticks(tick_marks, classes) #membuat ticks pada y
24
25      fmt = '.2f' if normalize else 'd' #fmt sebagai normalisasi
26      thresh = cm.max() / 2.  #variabel thresh menambil data max
         pada cm kemudian dibagi 2
27
28      plt.tight_layout() #mengatur layout pada plot
29      plt.ylabel('True label') #memberi nama label pada sumbu y
30      plt.xlabel('Predicted label') #memberi nama label pada
        sumbu x
31
32
33  # In[22]:
34
35  birds = pd.read_csv("data/CUB_200_2011/classes.txt",
36                      sep='\s+', header=None, usecols=[1], names
        =['birdname']) #membaca csv dengan ketentuan nama birdname
37  birds = birds['birdname'] #nama birds dengan ketentuan
        birdname
38  birds #menampilkan data birds
39
40
41  # In[23]:
42
43  import numpy as np #import library numpy sebagai np
44  np.set_printoptions(precision=2) #np sebagai variabel yang
        membuat set precision=2
45  plt.figure(figsize=(60,60), dpi=300) #plot sebagai figure
        dengan ketentuan sizw 60,60 dan dpi 300
46  plot_confusion_matrix(cm, classes=birds, normalize=True) #data
         cm dan clas birds dibuat sebagai plot
47  plt.show() #menampilkan hasil plot yang berbentuk grafik
```

Hasilnya adalah seperti dibawah ini :



Gambar 3.121 Menampilkan hasil plot

6. Jalankan program klasifikasi SVM dan Decission Tree pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

   - Menunjukkan klarifikasi dengan decission tree menggunakan dataset yang sama dan akan memunculkan akurasi prediksi.

```
# In[24]: Mencoba dengan metode Decission Tree dan SVM

from sklearn import tree #import library tree
clftree = tree.DecisionTreeClassifier() #clftree sebagai
    variabel untuk decision tree
clftree.fit(df_train_att, df_train_label) #sebagai data
    training
clftree.score(df_test_att, df_test_label) #sebagai data
    testing
```

   Hasilnya adalah seperti ini :

```
In [27]: from sklearn import tree
    ...: clftree = tree.DecisionTreeClassifier()
    ...: clftree.fit(df_train_att, df_train_label)
    ...: clftree.score(df_test_att, df_test_label)
Out[27]: 0.26135163674762407
```

Gambar 3.122 Hasil Soal 6 - 1

   - Menunjukkan klarifikasi dengan SVM menggunakan dataset yang sama dan akan memunculkan akurasi prediksi.

```
# In[25]:

from sklearn import svm #import library svm
clfsvm = svm.SVC() #clfsvm sebagai variabel untuk mengatur
    fungsi SVC
clfsvm.fit(df_train_att, df_train_label) #sebagai data
    training
clfsvm.score(df_test_att, df_test_label) #sebagai data testing
```

Hasilnya adalah seperti ini :

```
In [28]: from sklearn import svm
    ...: clfsvm = svm.SVC()
    ...: clfsvm.fit(df_train_att, df_train_label)
    ...: clfsvm.score(df_test_att, df_test_label)
Out[28]: 0.47729672650475186
```

Gambar 3.123 Hasil Soal 6 - 2

7. Jalankan program cross validaiton pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan hasil cross validation pada Random Forest.

```
# In[26]: Pengecekan Cross Validation

from sklearn.model_selection import cross_val_score #import
    cross_val_score
scores = cross_val_score(clf, df_train_att, df_train_label, cv
    =5) #variabel scores sebagai variabel prediksi dari data
    training
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.
    std() * 2)) #print data scores dengan ketentuan akurasi
```

Hasilnya adalah seperti ini :

```
In [29]: from sklearn.model_selection import cross_val_score
    ...: scores = cross_val_score(clf, df_train_att, df_train_label, cv=5)
    ...: # show average score and +/- two standard deviations away (covering 95% of
scores)
    ...: print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
Accuracy: 0.45 (+/- 0.03)
```

Gambar 3.124 Hasil Soal 7 - 1

- Menunjukkan hasil cross validation pada Desiccion Tree.

```
# In[27]:

scorestree = cross_val_score(clftree, df_train_att,
    df_train_label, cv=5) #sebagai prediksi menggunakan scores
     dan metode tree
print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(),
    scorestree.std() * 2)) #menampilkan dengan ketentuan yang
    ada
```

Hasilnya adalah seperti ini :

```
In [30]: scorestree = cross_val_score(clftree, df_train_att, df_train_label, cv=5)
   ...: print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(), scorestree.std() *
2))
Accuracy: 0.26 (+/- 0.01)
```

Gambar 3.125 Hasil Soal 7 - 2

- Menunjukkan hasil cross validation pada SVM.

```
# In[28]:

scoressvm = cross_val_score(clfsvm, df_train_att,
    df_train_label, cv=5) #sebagai data training
print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(),
    scoressvm.std() * 2)) #sebagai data testing dan output
    akurasi
```

Hasilnya adalah seperti ini :

```
In [31]: scoressvm = cross_val_score(clfsvm, df_train_att, df_train_label, cv=5)
   ...: print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(), scoressvm.std() * 2))
Accuracy: 0.47 (+/- 0.03)
```

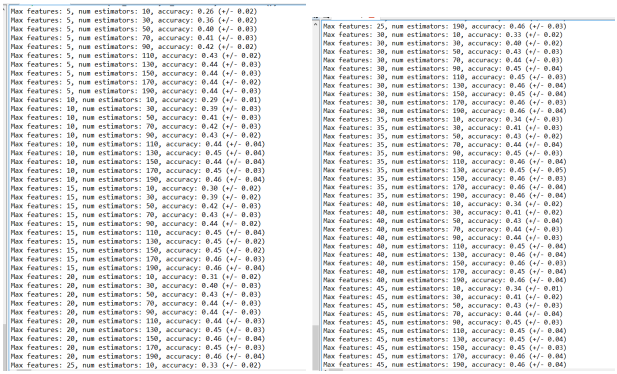
Gambar 3.126 Hasil Soal 7 - 3

8. Jalankan program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

- Menunjukkan informasi-informasi tree yang dibuat.

```
# In[29]: Pengamatan komponen informasi

max_features_opts = range(5, 50, 5) #max_features_opts sebagai
     variabel untuk membuat range 5,50,5
n_estimators_opts = range(10, 200, 20) #n_estimators_opts
    sebagai variabel untuk membuat range 10,200,20
rf_params = np.empty((len(max_features_opts)*len(
    n_estimators_opts),4), float) #rf_params sebagai variabel
    untuk menjumlahkan yang sudah di tentukan sebelumnya
i = 0
for max_features in max_features_opts: #pengulangan
    for n_estimators in n_estimators_opts: #pengulangan
        clf = RandomForestClassifier(max_features=max_features
    , n_estimators=n_estimators) #menampilkan variabel csf
        scores = cross_val_score(clf, df_train_att,
    df_train_label, cv=5) #scores sebagai variabel training
        rf_params[i,0] = max_features #index 0
        rf_params[i,1] = n_estimators #index 1
        rf_params[i,2] = scores.mean() #index 2
        rf_params[i,3] = scores.std() * 2 #index 3
        i += 1 #dengan ketentuan i += 1
        print("Max features: %d, num estimators: %d, accuracy:
     %0.2f (+/- %0.2f)" %       (max_features, n_estimators,
    scores.mean(), scores.std() * 2))
        #print hasil pengulangan yang sudah ditentukan
```

Hasilnya adalah seperti ini :

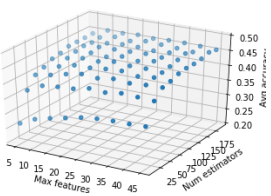


Gambar 3.127 Hasil Soal 8 - 1

- Menunjukkan hasil dari plotting komponen informasi sehingga dapat kita baca sebagai grafik 3D.

```
# In[30]:

import matplotlib.pyplot as plt #import library matplotlib
    sebagai plt
from mpl_toolkits.mplot3d import Axes3D #import axes3D untuk
    menampilkan plot 3 dimensi
from matplotlib import cm #memanggil data cm yang sudah
    tersedia
fig = plt.figure() #hasil plot sebagai figure
fig.clf() #figure di ambil dari clf
ax = fig.gca(projection='3d') #ax sebagai projection 3d
x = rf_params[:,0] #x sebagai index 0
y = rf_params[:,1] #y sebagai index 1
z = rf_params[:,2] #z sebagai index 2
ax.scatter(x, y, z) #membuat plot scatter x y z
ax.set_zlim(0.2, 0.5) #set zlim dengan ketentuan yang ada
ax.set_xlabel('Max features') #memberi nama label x
ax.set_ylabel('Num estimators') #memberi nama label y
ax.set_zlabel('Avg accuracy') #memberi nama label z
plt.show() #print hasil plot yang sudah dibuat.
```

Hasilnya adalah seperti ini:



Gambar 3.128 Hasil Soal 8 - 2

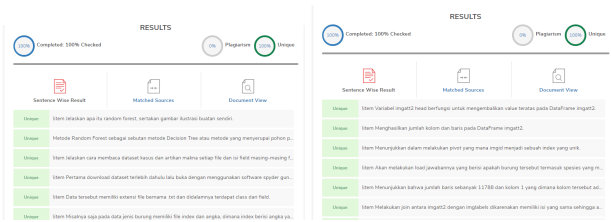
3.6.3 Penanganan Error

1. ScreenShoot Error

Gambar 3.129 FileNotFoundError

2. Tuliskan Kode Error dan Jenis Error
   - FileNotFoundError
3. Cara Penangan Error
   - FileNotFoundError
     Error terdapat pada kesalahan baca file csv, yang tidak terbaca. Dikarenakan letak file yang dibaca tidak para direktori yang sama. Seharusnya letakkan file di direktori yang sama.

3.6.4 Bukti Tidak Plagiat



Gambar 3.130 Bukti Tidak Melakukan Plagiat Chapter 3

BAB 4

CHAPTER 4

4.1 1174006 - Kadek Diva Krishna Murti

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

```
@inproceedings{awangga2017colenak,
  title={Colenak: GPS tracking model for post-stroke rehabilitation
    program using AES-CBC URL encryption and QR-Code},
  author={Awangga, Rolly Maulana and Fathonah, Nuraini Siti and
    Hasanudin, Trisna Irmayadi},
  booktitle={Information Technology, Information Systems and
    Electrical Engineering (ICITISEE), 2017 2nd International
    conferences on},
  pages={255--260},
  year={2017},
  organization={IEEE}
}
```


By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

Gambar 4.1 Kecerdasan Buatan.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

4.1.1 Teori

4.1.2 Praktek

4.1.3 Penanganan Error

4.1.4 Bukti Tidak Plagiat

Gambar 4.2 Kecerdasan Buatan.

BAB 5

CHAPTER 5

5.1 1174006 - Kadek Diva Krishna Murti

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

```
@inproceedings{awangga2017colenak,
  title={Colenak: GPS tracking model for post-stroke rehabilitation
    program using AES-CBC URL encryption and QR-Code},
  author={Awangga, Rolly Maulana and Fathonah, Nuraini Siti and
    Hasanudin, Trisna Irmayadi},
  booktitle={Information Technology, Information Systems and
    Electrical Engineering (ICITISEE), 2017 2nd International
    conferences on},
  pages={255--260},
  year={2017},
  organization={IEEE}
}
```


By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

Gambar 5.1 Kecerdasan Buatan.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

5.1.1 Teori

5.1.2 Praktek

5.1.3 Penanganan Error

5.1.4 Bukti Tidak Plagiat

Gambar 5.2 Kecerdasan Buatan.

BAB 6

CHAPTER 6

6.1 1174006 - Kadek Diva Krishna Murti

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

```
@inproceedings{awangga2017colenak,
  title={Colenak: GPS tracking model for post-stroke rehabilitation
    program using AES-CBC URL encryption and QR-Code},
  author={Awangga, Rolly Maulana and Fathonah, Nuraini Siti and
    Hasanudin, Trisna Irmayadi},
  booktitle={Information Technology, Information Systems and
    Electrical Engineering (ICITISEE), 2017 2nd International
    conferences on},
  pages={255--260},
  year={2017},
  organization={IEEE}
}
```


By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

Gambar 6.1 Kecerdasan Buatan.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

6.1.1 Teori

6.1.2 Praktek

6.1.3 Penanganan Error

6.1.4 Bukti Tidak Plagiat

Gambar 6.2 Kecerdasan Buatan.

BAB 7

CHAPTER 7

7.1 1174006 - Kadek Diva Krishna Murti

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

```
@inproceedings{awangga2017colenak,
  title={Colenak: GPS tracking model for post-stroke rehabilitation
    program using AES-CBC URL encryption and QR-Code},
  author={Awangga, Rolly Maulana and Fathonah, Nuraini Siti and
    Hasanudin, Trisna Irmayadi},
  booktitle={Information Technology, Information Systems and
    Electrical Engineering (ICITISEE), 2017 2nd International
    conferences on},
  pages={255--260},
  year={2017},
  organization={IEEE}
}
```


By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

Gambar 7.1 Kecerdasan Buatan.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

7.1.1 Teori

7.1.2 Praktek

7.1.3 Penanganan Error

7.1.4 Bukti Tidak Plagiat

Gambar 7.2 Kecerdasan Buatan.

DAFTAR PUSTAKA

[1] R. Awangga, “Sampeu: Servicing web map tile service over web map service to increase computation performance,” in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

Index