

Tugas

Kecerdasan Buatan



Helmi Salsabila

1194018

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

Politeknik Pos Indonesia

Bandung 2022

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

Contents

1	Mengenal Kecerdasan Buatan dan Scikit-Learn	1
1.1	Teori	1
1.2	Instalasi	3
1.3	Penanganan Error	5
2	Membangun Model Prediksi	6
2.1	Teori	6
2.2	scikit-learn	7
2.3	Penanganan Error	15
3	Prediksi dengan Random Forest	16
3.1	Teori	16
3.1.1	Random Forest	17
3.1.2	Confusion Matrix	19
3.1.3	Mencoba dengan metode Decission Tree dan SVM	20
3.1.4	Pengecekan Cross Validation	21
3.1.5	Pengamatan komponen informasi	21
3.2	Soal Teori	22
3.3	Praktek Program	22
3.4	Penanganan Error	30
3.5	Presentasi Tugas	31
4	Klasifikasi Teks	32
4.1	Teori	32
4.1.1	Vektorisasi data	32
4.1.2	Klasifikasi dengan Random Forest	34
4.1.3	Confusion Matrix	34
4.1.4	Pengecekan Cross Validation	34
4.2	Soal Teori	35

4.3	Praktek Program	35
4.4	Penanganan Error	40
4.5	Presentasi Tugas	41

List of Figures

1.1	Instalasi Library Scikit-Learn dan Matplotlib	3
1.2	Loading an Example Dataset	4
1.3	Learning and Predicting	4
1.4	Model persistence	4
1.5	Conventions	5
1.6	No module named 'sklearn'	5
2.1	Load dataset	8
2.2	Generate binary label	8
2.3	Generate binary label	9
2.4	Shuffle row	10
2.5	Number of passing	11
2.6	Fit a decision tree	11
2.7	Visualize tree	11
2.8	Save tree dot to png	12
2.9	scores	12
2.10	scores II	12
2.11	Show average score	13
2.12	DecisionTreeClassifier	13
2.13	DecisionTreeClassifier depth	14
2.14	Subplots	14
3.1	Aplikasi Pandas - Chapter3	23
3.2	Aplikasi Numpy - Chapter3	23
3.3	Aplikasi Matplotlib - Chapter3	23
3.4	Klasifikasi Random Fores - Chapter3	27
3.5	Confusion Matrix - Chapter3	28
3.6	Klasifikasi SVM - Chapter3	29
3.7	Cross Validaiton - Chapter3	29

3.8	Komponen Informasi - Chapter3	30
4.1	Aplikasi Pandas - Chapter4	36
4.2	Dataframe dipecah menjadi dua - Chapter4	36
4.3	Vektorisasi dan Klasifikasi $1194018 = \text{mod } 2$ - Chapter4	37
4.4	Vektorisasi dan Klasifikasi - Chapter4	39
4.5	Vektorisasi dan Klasifikasi Decision Tree - Chapter4	39
4.6	Confusion Matrix - Chapter4	39
4.7	Cross Validaiton - Chapter4	40
4.8	Komponen Informasi - Chapter4	40

Chapter 1

Mengenai Kecerdasan Buatan dan Scikit-Learn

Buku umum teori lengkap yang digunakan memiliki judul *Artificial intelligence: a modern approach*[?]. Untuk pratikum sebelum UTS menggunakan buku *Python Artificial Intelligence Projects for Beginners*[?]. Buku pelengkap penunjang penggunaan python menggunakan buku *Python code for Artificial Intelligence: Foundations of Computational Agents*[?]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti definisi kecerdasan buatan, sejarah kecerdasan buatan, perkembangan dan penggunaan di perusahaan
2. Memahami cara instalasi dan pemakaian sci-kit learn
3. Memahami cara penggunaan variabel explorer di spyder

Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten riset.

1.1 Teori

Praktek teori penunjang yang dikerjakan :

1. Buat Resume Definisi, Sejarah dan perkembangan Kecerdasan Buatan, dengan bahasa yang mudah dipahami dan dimengerti. Buatan sendiri bebas plagiat[hari ke 1](10)

2. Buat Resume mengenai definisi supervised learning, klasifikasi, regresi dan unsupervised learning. Data set, training set dan testing set.[hari ke 1](10)

Penjelasan Teori

Berikut merupakan penjelasan resume dari Teori :

- (a) Definisi AI (Kecerdasan Buatan) : Bagaimana sebuah komputer memiliki kemampuan yang sama dengan manusia yang dapat mengambil keputusan sendiri dari berbagai macam kasus yang di hadapinya. Contoh: komputer dapat berkomunikasi baik dengan kata, suara, gambar dan lain sebagainya.. Oleh karena itu AI (Kecerdasan Buatan) dapat di sebut sebagai robot/digitalisasi yang dikendalikan oleh sistem komputer untuk dapat menyelesaikan suatu tugasnya sesuai instruksi sistem.
- (b) Sejarah AI (Kecerdasan Buatan) : Tahun 1940 - 1950 mulai terbentuknya komputer modern Para ilmuwan mulai diskusi mengenai bidang sybernetics, matematika, algoritma dan teori jaringan pada tahun 1956, pada tahun yang sama McCarthy mendirikan Konferensi Dartmouth di Hanover, New Hampshire yang menemukan beberapa teori kompleks mengenai jaringan saraf dan pemikiran kreatif pada komputer. dengan demikian Kecerdasan Buatan launching.
- (c) Perkembangan AI (Kecerdasan Buatan) : 17 tahun berlalu tepatnya pada tahun 1973 Konferensi tersebut mendanai sebuah penelitian di MIT (universitas di Edinburgh,Stanford dan Carnegie Mellon) yang mana komputer pemrograman mulai membuktikan masalah aljabar , teorema geometris yang menggunakan pemahaman sintaks dan tata bahasa inggris.
- (d) Scikit-Learn Supervised Learning : Merupakan pengumpulan data yang berlabel serta menyediakan algoritma untuk mendukung penilaian di masa yang akan datang. contoh: Mobil self-driving, chatbots, pengenalan wajah, robot, sistem pakar.
- (e) Scikit-Learn Unsupervised Learning : Merupakan pengumpulan data yang tidak berlabel salah satunya yakni untuk menguji AI sebagai mana mencari tahu cara memilah bebek dan ayam atau juga menambahkan kategori yang berbeda.
- (f) Scikit-Learn Regresi : Metode analisis statistik untuk melihat pengaruh terhadap 2 atau lebih variabel. contoh: berat atau gaji.

- (g) Scikit-Learn Klasifikasi : Proses pengelompokan benda yang sama dan benda yang beda. Contoh: mengidentifikasi orang tersebut apakah pria atau wanita, orang itu udah mandi atau tidak mandi.
- (h) Scikit-Learn Dataset : Konsepnya sama dengan yang ada pada database hanya saja dataset ini berisi koleksi data table dan data relation.
- (i) Scikit-Learn Training Set : Berguna untuk algoritma klasifikasi sebagai contoh neural network, bayesian, decision tree bertujuan untuk membangun sebuah model classifier.
- (j) Scikit-Learn Testing Set : Bertujuan untuk mengukur classifier ketika berhasil melakukan klasifikasi bersifat true.

1.2 Instalasi

Membuka <https://scikit-learn.org/stable/tutorial/basic/tutorial.html>. Dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat. Dan wajib skrinsut dari komputer sendiri.

Youtube : <https://youtu.be/srzfw6J4ZaM>

1. Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer[hari ke 1](10)

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

(SesiBelajar) C:\Users\Helmi Salsabila>conda --version
conda 4.10.3

(SesiBelajar) C:\Users\Helmi Salsabila>python --version
Python 3.9.7

(SesiBelajar) C:\Users\Helmi Salsabila>pip install scikit-learn
Collecting scikit-learn
  Downloading scikit-learn-1.0.2-cp39-cp39-win-amd64.whl (7.2 MB)
    Requirement already satisfied: scipy<1.1.0 in c:\users\helmi salsabila\.conda\envs\sesibelajar\lib\site-packages (from scikit-learn) (1.0.0)
    Requirement already satisfied: numpy>=1.14.6 in c:\users\helmi salsabila\.conda\envs\sesibelajar\lib\site-packages (from scikit-learn) (1.22.2)
    Collecting joblib>=0.11
      Downloading joblib-1.1.0-py2.py3-none-any.whl (300 kB)
    Installing collected packages: threadpoolctl, joblib, scikit-learn
    Successfully installed joblib-1.1.0 scikit-learn-1.0.2 threadpoolctl-3.1.0

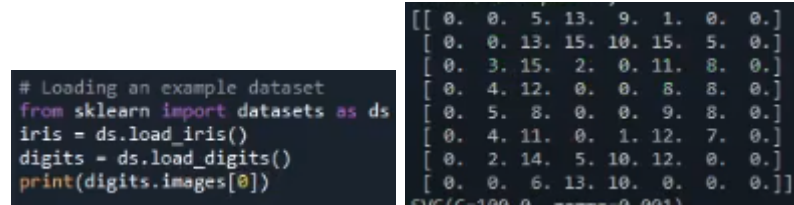
(SesiBelajar) C:\Users\Helmi Salsabila>

C:\Windows\system32\cmd.exe - pip install matplotlib
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

(SesiBelajar) C:\Users\Helmi Salsabila>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.5.1-cp39-cp39-win-amd64.whl (7.2 MB)
    Requirement already satisfied: python-dateutil>=2.7 in c:\users\helmi salsabila\.conda\envs\sesibelajar\lib\site-packages (from matplotlib) (2.8.2)
    Requirement already satisfied: numpy>=1.17 in c:\users\helmi salsabila\.conda\envs\sesibelajar\lib\site-packages (from matplotlib) (1.22.2)
    Collecting fonttools>=4.22.0
      Downloading fonttools-4.38.0-py3-none-any.whl (890 kB)
    Collecting kiwisolver>=1.0.1
      Downloading kiwisolver-1.4.0-cp39-cp39-win-amd64.whl (51 kB)
    Collecting pillow>=6.2.0
      Downloading pillow-9.5.1-cp39-cp39-win-amd64.whl (3.2 MB)
    Requirement already satisfied: packaging>=20.0 in c:\users\helmi salsabila\.conda\envs\sesibelajar\lib\site-packages (from matplotlib) (21.3)
    Requirement already satisfied: pyparsing>=2.2.1 in c:\users\helmi salsabila\.conda\envs\sesibelajar\lib\site-packages (from matplotlib) (3.0.4)
    Collecting cython>=0.10
      Downloading cython-0.11.0-py2.py3-none-any.whl (6.4 kB)
    Requirement already satisfied: six>=1.5 in c:\users\helmi salsabila\.conda\envs\sesibelajar\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
    Installing collected packages: pillow, kiwisolver, fonttools, cython, matplotlib
  
```

Figure 1.1: Instalasi Library Scikit-Learn dan Matplotlib

2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 1](10)

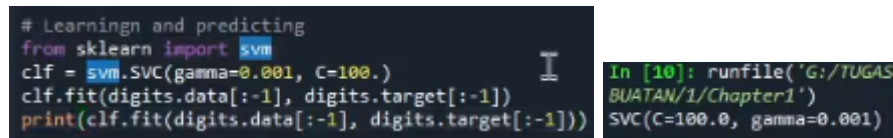


```
# Loading an example dataset
from sklearn import datasets as ds
iris = ds.load_iris()
digits = ds.load_digits()
print(digits.images[0])
```

```
[[ 0.  0.  5. 13.  9.  1.  0.  0.]
 [ 0.  0. 13. 15. 10. 15.  5.  0.]
 [ 0.  3. 15.  2.  0. 11.  8.  0.]
 [ 0.  4. 12.  0.  0.  8.  8.  0.]
 [ 0.  5.  8.  0.  0.  9.  8.  0.]
 [ 0.  4. 11.  0.  1. 12.  7.  0.]
 [ 0.  2. 14.  5. 10. 12.  0.  0.]
 [ 0.  0.  6. 13. 10.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 SVC(C=100.0, gamma=0.001)
```

Figure 1.2: Loading an Example Dataset

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)

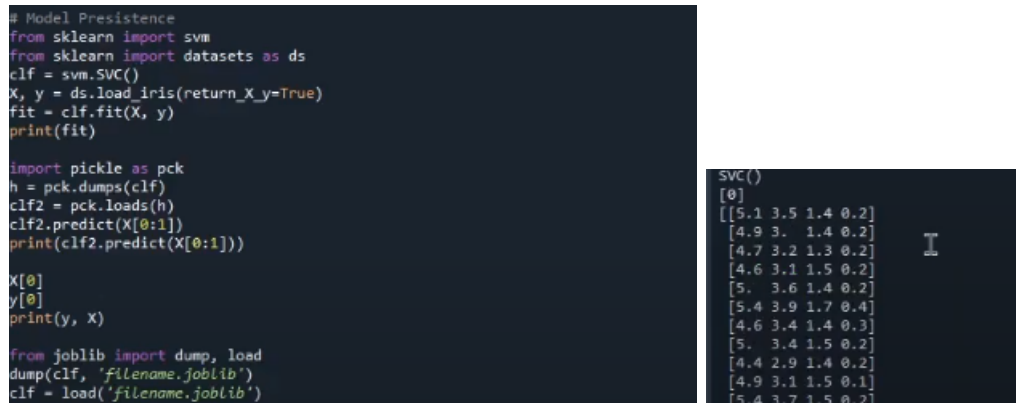


```
# Learningn and predicting
from sklearn import svm
clf = svm.SVC(gamma=0.001, C=100.)
clf.fit(digits.data[:-1], digits.target[:-1])
print(clf.fit(digits.data[:-1], digits.target[:-1]))
```

```
In [10]: runfile('G:/TUGAS
BUATAN/1/Chapter1')
SVC(C=100.0, gamma=0.001)
```

Figure 1.3: Learning and Predicting

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)



```
# Model Persistence
from sklearn import svm
from sklearn import datasets as ds
clf = svm.SVC()
X, y = ds.load_iris(return_X_y=True)
fit = clf.fit(X, y)
print(fit)

import pickle as pck
h = pck.dumps(clf)
clf2 = pck.loads(h)
clf2.predict(X[0:1])
print(clf2.predict(X[0:1]))

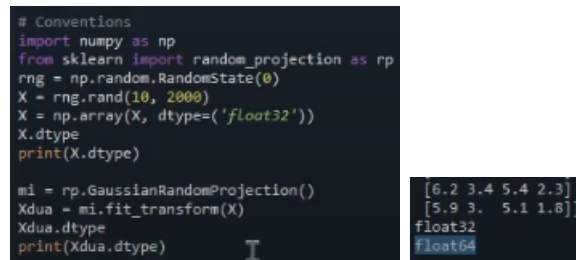
X[0]
y[0]
print(y, x)

from joblib import dump, load
dump(clf, 'filename.joblib')
clf = load('filename.joblib')
```

```
SVC()
[0]
[[5.1 3.5 1.4 0.2]
 [4.9 3. 1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5. 3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5. 3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]
 [5.4 3.7 1.5 0.2]]
```

Figure 1.4: Model persistence

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)



```
# Conventions
import numpy as np
from sklearn import random_projection as rp
rng = np.random.RandomState(0)
X = rng.rand(10, 2000)
X = np.array(X, dtype='float32')
X.dtype
print(X.dtype)

mi = rp.GaussianRandomProjection()
Xdua = mi.fit_transform(X)
Xdua.dtype
print(Xdua.dtype)
```

Output:

```
float32
float64
```

Output array:

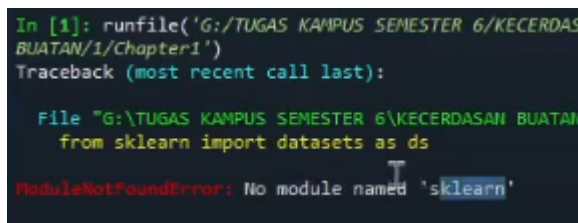
```
[[6.2 3.4 5.4 2.3]
 [5.9 3. 5.1 1.8]]
```

Figure 1.5: Conventions

1.3 Penanganan Error

Dari percobaan yang dilakukan di atas, apabila mendapatkan error maka:

1. skripsi error[hari ke 2](10)



```
In [1]: runfile('G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN
          BUATAN/1/Chapter1')
Traceback (most recent call last):

  File "G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN
    from sklearn import datasets as ds
ModuleNotFoundError: No module named 'sklearn'
```

Figure 1.6: No module named 'sklearn'

2. Tuliskan kode eror dan jenis errornya [hari ke 2](10)
 Belum menginstall library dari scikit-learn sehingga terjadi error '*No module named sklearn*'
3. Solusi pemecahan masalah error tersebut[hari ke 2](10)
 Instal terlebih dahulu library scikit-learnnya dengan menggunakan terminal yang ada pada anacondanya, kemudian ketikkan '*pip install scikit-learn*' tunggu hingga prosesnya selesai dan library sudah bisa di gunakan pada proyek anda.

Chapter 2

Membangun Model Prediksi

Untuk pratikum saati ini menggunakan buku *Python Artificial Intelligence Projects for Beginners*[?]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Dataset ada di <https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners> . Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti implementasi klasifikasi
2. Memahami data set, training dan testing data
3. Memahami Decission tree.
4. Memahami information gain dan entropi.

Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten riset. Kode program menggunakan input listing ditaruh di folder src ekstensi .py dan dipanggil ke latex dengan input listings. Tulisan dan kode tidak boleh plagiat, menggunakan bahasa indonesia yang sesuai dengan gaya bahasa buku teks.

2.1 Teori

Praktek teori penunjang yang dikerjakan(nilai 5 per nomor, untuk hari pertama) :

1. Jelaskan apa itu binary classification dilengkapi ilustrasi gambar sendiri
Binary Classification merupakan Klasifikasi biner yang berupa kelas positif atau negatif yang di tetapkan untuk tujuan yang praktis dari pada akurasi keseluruhan serta relatif dari berbagai jenis kesalahan yang menarik.

Contoh: mendeteksi penyakit ketika tidak ada(false positive), tidak mendeteksi penyakit ketika hadir(false negative)

2. Jelaskan apa itu supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar sendiri.
3. Jelaskan apa itu evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar sendiri.

Evaluasi dan Akurasi Evaluasi untuk mengukur akurasi model bekerja, dan akurasi persentase klasifikasi dengan benar.

4. Jelaskan bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix buatan sendiri.
5. Jelaskan bagaimana K-fold cross validation bekerja dengan gambar ilustrasi contoh buatan sendiri.
6. Jelaskan apa itu decision tree dengan gambar ilustrasi contoh buatan sendiri.

Decision Tree Digunakan untuk klasifikasi dan regresi dengan menggunakan metode pembelajaran non-parametrik yang dapat menghasilkan nilai puput berupa model yang memprediksi nilai variable target dengan aturan keputusan dari fitur data.

7. Jelaskan apa itu information gain dan entropi dengan gambar ilustrasi buatan sendiri.

Information Gain merupakan penurunan entropi setelah dataset di bagi pada atribut serta membangun decision tree untuk menemukan atribut yang mengembalikan informasi krusial(cabang paling homogen).

Entropi merupakan tingkat keacakan dalam informasi, semakin tinggi entropi semakin sulit menyimpulkan dari informasi yang acak. begitupun sebaliknya.

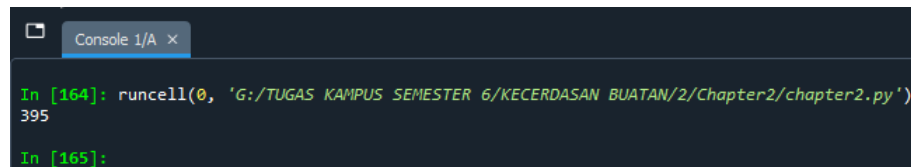
2.2 scikit-learn

Dataset ambil di <https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners> folder Chapter01. Tugas anda adalah, dataset ganti menggunakan **student-mat.csv** dan mengganti semua nama variabel dari kode di bawah ini dengan nama-nama makanan (NPM mod 3=0), kota (NPM mod 3=1), buah (NPM mod 3=2), . Jalankan satu per satu kode tersebut di spyder dengan menggunakan

textitRun current cell. Kemudian Jelaskan dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat dan wajib skrinsut dari komputer sendiri masing masing nomor di bawah ini(nilai 5 masing masing pada hari kedua).

Youtube :

1. # load dataset (student mat pakenya)
import pandas as pd
d = pd.read_csv('student-mat.csv', sep=';')
len(d)



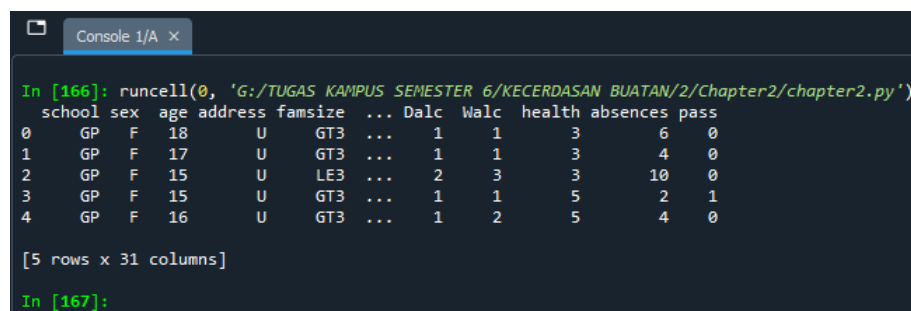
```

In [164]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
395
In [165]:

```

Figure 2.1: Load dataset

2. # generate binary label (pass/fail) based on G1+G2+G3
(test grades, each 0-20 pts); threshold for passing is sum>=30
d['pass'] = d.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3'])
>= 35 else 0, axis=1)
d = d.drop(['G1', 'G2', 'G3'], axis=1)
d.head()



```

In [166]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
school sex age address famsize ... Dalc Walc health absences pass
0 GP F 18 U GT3 ... 1 1 3 6 0
1 GP F 17 U GT3 ... 1 1 3 4 0
2 GP F 15 U LE3 ... 2 3 3 10 0
3 GP F 15 U GT3 ... 1 1 5 2 1
4 GP F 16 U GT3 ... 1 2 5 4 0

[5 rows x 31 columns]
In [167]:

```

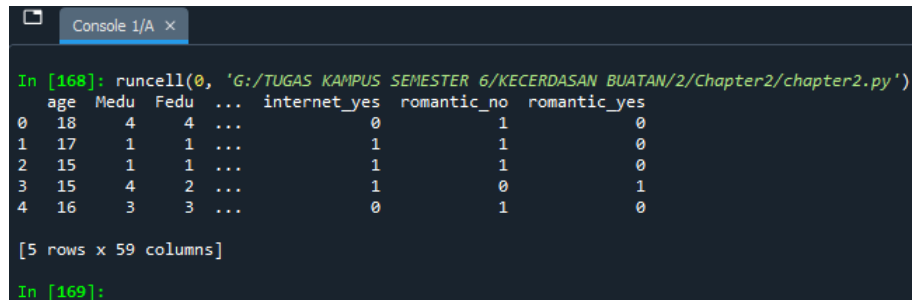
Figure 2.2: Generate binary label

3. # use one-hot encoding on categorical columns
d = pd.get_dummies(d, columns=['sex', 'school', 'address',
'famsize',
'Pstatus', 'Mjob', 'Fjob',

```

        'reason', 'guardian', 'schoolsup',
        'famsup', 'paid', 'activities',
        'nursery', 'higher', 'internet',
        'romantic'])
d.head()

```



The screenshot shows a Jupyter Notebook console window titled 'Console 1/A x'. It displays the execution of a `runcell` command in cell 168, which runs a script located at `'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py'`. The output shows the first five rows of a DataFrame with 59 columns. The columns visible are `age`, `Medu`, `Fedu`, `internet_yes`, `romantic_no`, and `romantic_yes`. The data is as follows:

	age	Medu	Fedu	...	internet_yes	romantic_no	romantic_yes
0	18	4	4	...	0	1	0
1	17	1	1	...	1	1	0
2	15	1	1	...	1	1	0
3	15	4	2	...	1	0	1
4	16	3	3	...	0	1	0

Below the data, it indicates the DataFrame shape: `[5 rows x 59 columns]`. The prompt for the next cell, `In [169]:`, is also visible.

Figure 2.3: Generate binary label

```

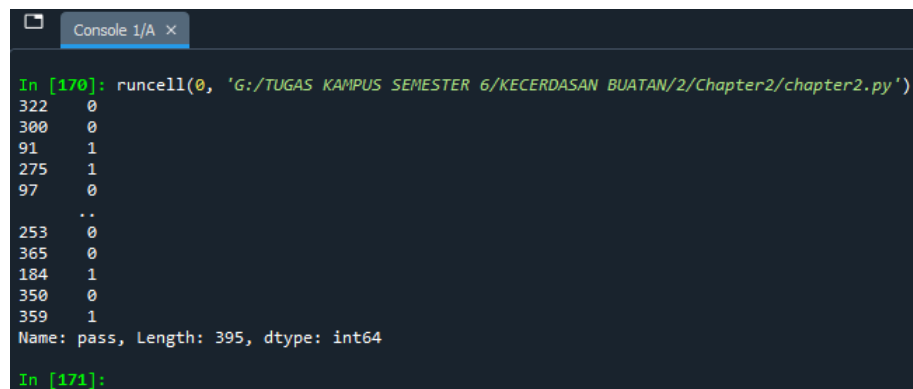
4. # shuffle rows
d = d.sample(frac=1)
# split training and testing data
d_train = d[:500]
d_test = d[500:]

d_train_att = d_train.drop(['pass'], axis=1)
d_train_pass = d_train['pass']

d_test_att = d_test.drop(['pass'], axis=1)
d_test_pass = d_test['pass']

d_att = d.drop(['pass'], axis=1)
d_pass = d['pass']

```



```

Console 1/A x
In [170]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
322      0
300      0
91       1
275      1
97       0
..
253      0
365      0
184      1
350      0
359      1
Name: pass, Length: 395, dtype: int64
In [171]:

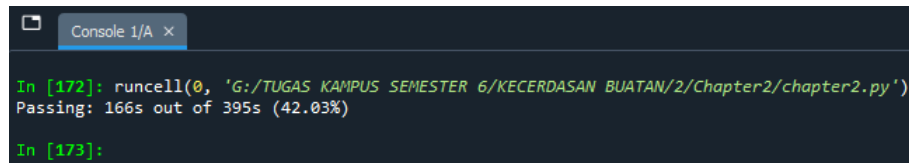
```

Figure 2.4: Shuffle row

```

# number of passing students in whole dataset:
import numpy as np
print("Passing: %d out of %d (%.2f%%)" % (np.sum(d_pass), len(d_pass),
    100*float(np.sum(d_pass)) / len(d_pass)))

```

```

In [172]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
Passing: 166s out of 395s (42.03%)

In [173]:

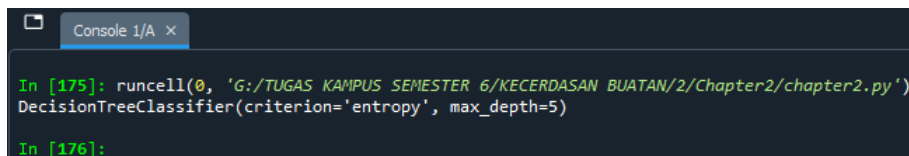
```

Figure 2.5: Number of passing

```

5. # fit a decision tree
from sklearn import tree
t = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
t = t.fit(d_train_att, d_train_pass)

```



```

In [175]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
DecisionTreeClassifier(criterion='entropy', max_depth=5)

In [176]:

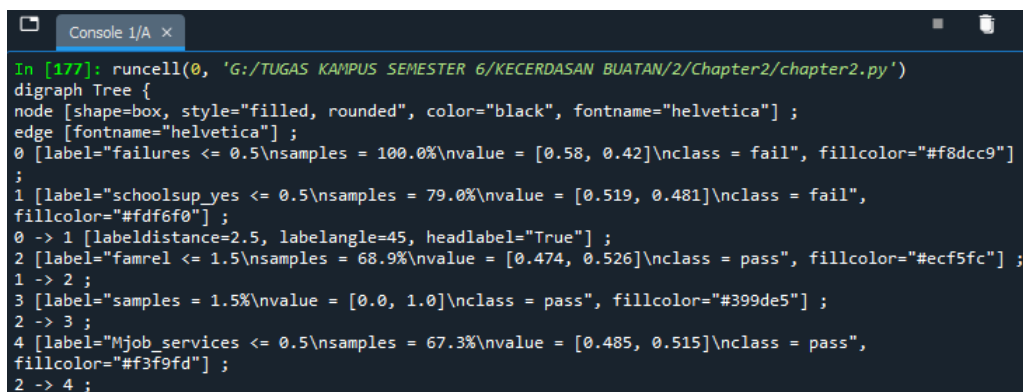
```

Figure 2.6: Fit a decision tree

```

6. # visualize tree
import graphviz
dot_data = tree.export_graphviz(t, out_file=None, label="all",
                                impurity=False, proportion=True,
                                feature_names=list(d_train_att),
                                class_names=["fail", "pass"],
                                filled=True, rounded=True)
graph = graphviz.Source(dot_data)
graph

```



```

In [177]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
digraph Tree {
  node [shape=box, style="filled, rounded", color="black", fontname="helvetica"] ;
  edge [fontname="helvetica"] ;
  0 [label="failures <= 0.5\nsamples = 100.0%\nvalue = [0.58, 0.42]\n\nclass = fail", fillcolor="#f8dcc9"] ;
  0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
  1 [label="schoolsup_yes <= 0.5\nsamples = 79.0%\nvalue = [0.519, 0.481]\n\nclass = fail", fillcolor="#fdf6f0"] ;
  0 -> 2 [label="famrel <= 1.5\nsamples = 68.9%\nvalue = [0.474, 0.526]\n\nclass = pass", fillcolor="#ecf5fc"] ;
  1 -> 2 ;
  3 [label="samples = 1.5%\nvalue = [0.0, 1.0]\n\nclass = pass", fillcolor="#399de5"] ;
  2 -> 3 ;
  4 [label="Mjob_services <= 0.5\nsamples = 67.3%\nvalue = [0.485, 0.515]\n\nclass = pass", fillcolor="#f3f9fd"] ;
  2 -> 4 ;
}

```

Figure 2.7: Visualize tree

7. # save tree

```
tree.export_graphviz(t, out_file="student-performance.dot",
    label="all", impurity=False,
    proportion=True,

    feature_names=list(d_train_att),
    class_names=["fail", "pass"],
    filled=True, rounded=True)
```

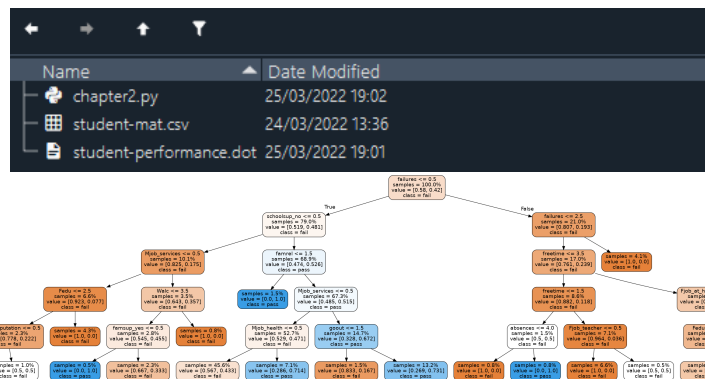


Figure 2.8: Save tree dot to png

8. t.score(d_test_att, d_test_pass)

```
Console 1/A x

In [179]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
0.7063291139240506

In [180]:
```

Figure 2.9: scores

9. from sklearn.model_selection import cross_val_score
scores = cross_val_score(t, d_att, d_pass, cv=5)

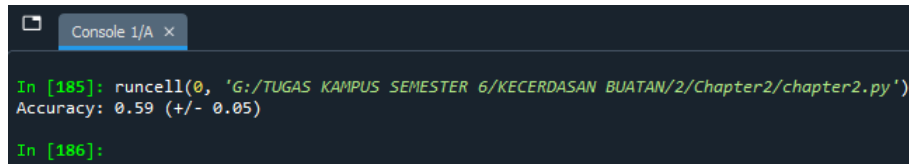
```
Console 1/A x

In [181]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
[0.62025316 0.5443038 0.55696203 0.50632911 0.55696203]

In [182]:
```

Figure 2.10: scores II

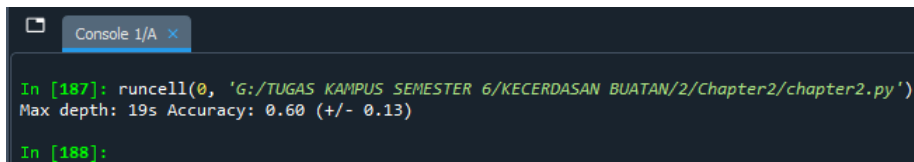
```
# show average score and +/- two standard deviations away
#(covering 95% of scores)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```



```
Console 1/A x
In [185]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
Accuracy: 0.59 (+/- 0.05)
In [186]:
```

Figure 2.11: Show average score

```
10. for max_depth in range(1, 20):
    t = tree.DecisionTreeClassifier(criterion="entropy",
    max_depth=max_depth)
    scores = cross_val_score(t, d_att, d_pass, cv=5)
    print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" %
    (max_depth, scores.mean(), scores.std() * 2)
    )
```



```
Console 1/A x
In [187]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
Max depth: 19s Accuracy: 0.60 (+/- 0.13)
In [188]:
```

Figure 2.12: DecisionTreeClassifier

```
11. depth_acc = np.empty((19,3), float)
    i = 0
    for max_depth in range(1, 20):
        t = tree.DecisionTreeClassifier(criterion="entropy",
        max_depth=max_depth)
        scores = cross_val_score(t, d_att, d_pass, cv=5)
        depth_acc[i,0] = max_depth
        depth_acc[i,1] = scores.mean()
        depth_acc[i,2] = scores.std() * 2
        i += 1

    depth_acc
```

```

Console 1/A x
In [188]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')
[[1.00000000e+000 1.01265823e-002 2.47032823e-323]
 [2.47032823e-323 5.43472210e-323 4.94065646e-323]
 [4.44659081e-323 3.45845952e-323 4.44659081e-323]
 [2.96439388e-323 2.47032823e-323 2.47032823e-323]
 [3.45845952e-323 4.44659081e-323 2.47032823e-323]
 [2.96439388e-323 2.96439388e-323 4.94065646e-323]
 [4.94065646e-323 4.94065646e-323 4.94065646e-323]
 [5.92878775e-323 5.92878775e-323 4.94065646e-323]
 [4.94065646e-323 6.42285340e-323 5.92878775e-323]
 [5.92878775e-323 6.42285340e-323 5.92878775e-323]

```

Figure 2.13: DecisionTreeClassifier depth

```

12. import matplotlib.pyplot as plt
    fig, ax = plt.subplots()
    ax.errorbar(depth_acc[:,0], depth_acc[:,1], yerr=depth_acc[:,2])
    plt.show()

```

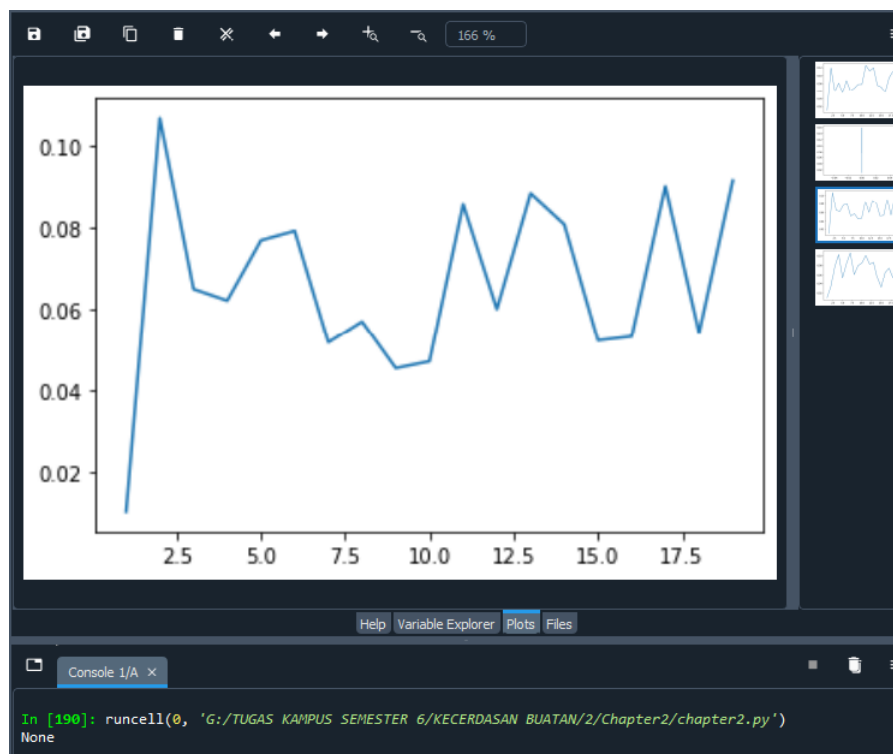


Figure 2.14: Subplots

2.3 Penanganan Error

Dari percobaan yang dilakukan di atas, error yang kita dapatkan di dokumentasikan dan di selesaikan(nilai 5 hari kedua):

1. skrinsut error
2. Tuliskan kode eror dan jenis errornya
3. Solusi pemecahan masalah error tersebut

Chapter 3

Prediksi dengan Random Forest

Untuk pratikum saati ini menggunakan buku *Python Artificial Intelligence Projects for Beginners*[?]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Kode program ada di <https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners> . Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti implementasi klasifikasi dan teknik evaluasi
2. Memprediksi spesies burung dengan random forest
3. Memahami Confusion Matrix.

Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten riset. Kode program menggunakan input listing ditaruh di folder src ekstensi .py dan dipanggil ke latex dengan input listings. Tulisan dan kode tidak boleh plagiat, menggunakan bahasa indonesia yang sesuai dengan gaya bahasa buku teks.

3.1 Teori

Random Forest adalah hasil voting dari beberapa decision tree yang masing-masing memegang atribut yang berbeda. Jadi setiap decision tree spesifik terhadap atribut tersebut yang merupakan bagian kecil dari keseluruhan atribut di data set. Hindari RF jika atribut terlalu sedikit untuk membentuk beberapa tree. Pada praktek kali ini menggunakan dataset spesies burung yang diambil dari situs (<http://www.vision.caltech.edu/visip200-2011.html>). Didalamnya terdapat 12.000 foto dari 200 spesies yang berbeda. Yang akan kita pakai untuk RF hanya atribut dari burungnya saja seperti ukuran,

bentuk dan warna. Data tersebut diberi label secara manual oleh manusia dengan memanfaatkan jasa dari Amazon's Mechanical Turk.

3.1.1 Random Forest

Pertama dataset kita baca terlebih dahulu.

```
1 import pandas as pd
2
3 # some lines have too many fields (?), so skip bad lines
4 imgatt = pd.read_csv("data/CUB_200_2011/attributes/
    image_attribute_labels.txt",
5                      sep='\s+', header=None, error_bad_lines=False,
6                      warn_bad_lines=False,
7                      usecols=[0,1,2], names=['imgid', 'attid', 'present'
8                      ])
9
```

Listing 3.1: Membaca data file txt

Melihat sebagian data awal, dengan menggunakan listing 3.2.

```
1 imgatt.head()
```

Listing 3.2: Melihat sebagian data awal

Melihat jumlah data menggunakan listing 3.3.

```
1 imgatt.shape
```

Listing 3.3: Mengetahui jumlah data

Merubah atribut menjadi kolom dengan menggunakan pivot layaknya excel. lalu kita cek isinya dengan menggunakan perintah pada listing 3.4.

```
1 imgatt2 = imgatt.pivot(index='imgid', columns='attid', values='present')
2
3 imgatt2.head()
4 imgatt2.shape
```

Listing 3.4: Pivot dataset

Sekarang kita akan meload jawabannya yang berisi apakah burung itu termasuk dalam spesies yang mana. Dua kolomnya adalah imgid dan label. Dan melakukan pivot yang mana imgid menjadi index yang artinya unik perintahnya ada di listing 3.5. Lalu kita cek kembali datanya.

```
1 imglabels = pd.read_csv("data/CUB_200_2011/image_class_labels.txt",
2                          sep=' ', header=None, names=['imgid', 'label'])
3
4 imglabels = imglabels.set_index('imgid')
5
6
7 imglabels.head()
8 imglabels.shape
```

Listing 3.5: membaca dataset label file txt

Karena isinya sama kita bisa melakukan join antara dua data. Sehingga kita akan mendapatkan data ciri dan data jawabannya atau labelnya sehingga bisa dikategorikan supervised learning. maka perintah untuk menggabungkan kedua data dan kemudian kita melakukan pemisahan antara data set untuk training dan test dengan perintah di listing 3.6.

```
1 df = imgatt2.join(imglabels)
2 df = df.sample(frac=1)
```

Listing 3.6: Menggabungkan field dari dua file terpisah

Kemudian drop label yang didepan, dan gunakan label yang paling belakang yang baru di join dengan perintah listing 3.7.

```
1 df_att = df.iloc[:, :312]
2 df_label = df.iloc[:, 312:]
```

Listing 3.7: Memisahkan dan memilih label

Kita bisa mengecek isinya dengan perintah listing 3.8.

```
1 df_att.head()
2 df_label.head()
```

Listing 3.8: Melihat isi masing masing data frame

Kita bagi menjadi dua bagian, 8000 row pertama sebagai data training sisanya sebagai data testing dengan perintah listing 3.9.

```
1 df_train_att = df_att[:8000]
2 df_train_label = df_label[:8000]
3 df_test_att = df_att[8000:]
4 df_test_label = df_label[8000:]
5
6 df_train_label = df_train_label['label']
7 df_test_label = df_test_label['label']
```

Listing 3.9: Pembagian data training dan test

Kita panggil kelas RandomForestClassifier. max features diartikan sebagai berapa banyak kolom pada setiap tree dengan perintah listing 3.10.

```
1 from sklearn.ensemble import RandomForestClassifier
2 clf = RandomForestClassifier(max_features=50, random_state=0,
    n_estimators=100)
```

Listing 3.10: Instansiasi kelas Random Forest

Kemudian lakukan fit untuk membangun random forest yang sudah ditentukan dengan maksimum fitur sebanyak 50 untuk perpohonnya dengan perintah listing 3.11.

```
1 clf.fit(df_train_att, df_train_label)
```

Listing 3.11: Fitting random forest dengan dataset training

Hasilnya bisa kita dapatkan dengan perintah predict dengan perintah listing 3.12.

```
1 print(clf.predict(df_train_att.head()))
```

Listing 3.12: Melihat Hasil prediksi

Untuk besaran akurasi dengan perintah listing 3.13

```
1 clf.score(df_test_att, df_test_label)
```

Listing 3.13: Score perolehan dari klasifikasi

3.1.2 Confusion Matrix

Dari RF kita coba petakan ke dalam Confusion Matrix dan lihat hasilnya dengan perintah listing 3.14.

```
1 from sklearn.metrics import confusion_matrix
2 pred_labels = clf.predict(df_test_att)
3 cm = confusion_matrix(df_test_label, pred_labels)
4
5 cm
```

Listing 3.14: Membuat Confusion Matrix

Kemudian kita plot dengan perintah

```
1 import matplotlib.pyplot as plt
2 import itertools
3 def plot_confusion_matrix(cm, classes,
4                           normalize=False,
5                           title='Confusion matrix',
6                           cmap=plt.cm.Blues):
7     """
8     This function prints and plots the confusion matrix.
9     Normalization can be applied by setting 'normalize=True'.
10    """
11     if normalize:
12         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
13         print("Normalized confusion matrix")
14     else:
15         print('Confusion matrix, without normalization')
16
17     print(cm)
18
19     plt.imshow(cm, interpolation='nearest', cmap=cmap)
20     plt.title(title)
21     #plt.colorbar()
22     tick_marks = np.arange(len(classes))
23     plt.xticks(tick_marks, classes, rotation=90)
24     plt.yticks(tick_marks, classes)
25
26     fmt = '.2f' if normalize else 'd'
27     thresh = cm.max() / 2.
```

```

28     #for i, j in itertools.product(range(cm.shape[0]), range(cm.shape
    [1])):
29     #     plt.text(j, i, format(cm[i, j], fmt),
30     #             horizontalalignment="center",
31     #             color="white" if cm[i, j] > thresh else "black")
32
33     plt.tight_layout()
34     plt.ylabel('True label')
35     plt.xlabel('Predicted label')

```

Listing 3.15: Plotting Confusion Matrix

Agar plot sumbunya sesuai dengan nama datanya maka kita set dengan perintah

```

1 birds = pd.read_csv("data/CUB_200_2011/classes.txt",
2                     sep='\s+', header=None, usecols=[1], names=['
    birdname'])
3 birds = birds['birdname']
4 birds

```

Listing 3.16: Membaca file classes.txt

Lalu kita plot

```

1 import numpy as np
2 np.set_printoptions(precision=2)
3 plt.figure(figsize=(60,60), dpi=300)
4 plot_confusion_matrix(cm, classes=birds, normalize=True)
5 plt.show()

```

Listing 3.17: Plot hasil perubahan label

3.1.3 Mencoba dengan metode Decission Tree dan SVM

Kita coba menggunakan Decission tree

```

1 from sklearn import tree
2 clftree = tree.DecisionTreeClassifier()
3 clftree.fit(df_train_att, df_train_label)
4 clftree.score(df_test_att, df_test_label)

```

Listing 3.18: Mencoba klasifikasi dengan decission tree dengan dataset yang sama

Kita coba menggunakan SVM

```

1 from sklearn import svm
2 clfsvm = svm.SVC()
3 clfsvm.fit(df_train_att, df_train_label)
4 clfsvm.score(df_test_att, df_test_label)

```

Listing 3.19: Mencoba klasifikasi dengan SVM dengan dataset yang sama

3.1.4 Pengecekan Cross Validation

Pengecekan Cross Validation untuk random forest

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

Listing 3.20: Hasil Cross Validation random forest

untuk decision tree

```
1 scorestree = cross_val_score(clftree, df_train_att, df_train_label, cv
  =5)
2 print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(), scorestree.std
  () * 2))
```

Listing 3.21: Hasil Cross Validation Decision Tree

untuk SVM

```
1 scoressvm = cross_val_score(clfsvm, df_train_att, df_train_label, cv=5)
2 print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(), scoressvm.std()
  * 2))
```

Listing 3.22: Hasil Cross Validation SVM

3.1.5 Pengamatan komponen informasi

Untuk mengetahui berapa banyak tree yang dibuat, berapa banyak atribut yang dipakai dan informasi lainnya menggunakan kode

```
1 max_features_opts = range(5, 50, 5)
2 n_estimators_opts = range(10, 200, 20)
3 rf_params = np.empty((len(max_features_opts)*len(n_estimators_opts),4),
  float)
4 i = 0
5 for max_features in max_features_opts:
6     for n_estimators in n_estimators_opts:
7         clf = RandomForestClassifier(max_features=max_features,
  n_estimators=n_estimators)
8         scores = cross_val_score(clf, df_train_att, df_train_label, cv
  =5)
9         rf_params[i,0] = max_features
10        rf_params[i,1] = n_estimators
11        rf_params[i,2] = scores.mean()
12        rf_params[i,3] = scores.std() * 2
13        i += 1
14        print("Max features: %d, num estimators: %d, accuracy: %0.2f
  (+/- %0.2f)" % (max_features, n_estimators, scores.
  mean(), scores.std() * 2))
```

Listing 3.23: Melakukan Pengamatan komponen informasi

Dan kita bisa melakukan plot informasi ini dengan kode

```
1 import matplotlib.pyplot as plt
2 from mpl_toolkits.mplot3d import Axes3D
3 from matplotlib import cm
4 fig = plt.figure()
5 fig.clf()
6 ax = fig.gca(projection='3d')
7 x = rf_params[:,0]
8 y = rf_params[:,1]
9 z = rf_params[:,2]
10 ax.scatter(x, y, z)
11 ax.set_zlim(0.2, 0.5)
12 ax.set_xlabel('Max features')
13 ax.set_ylabel('Num estimators')
14 ax.set_zlabel('Avg accuracy')
15 plt.show()
```

Listing 3.24: Plot Komponen informasi agar bisa dibaca

3.2 Soal Teori

Praktek teori penunjang yang dikerjakan(nilai 5 per nomor, untuk hari pertama) :

1. Jelaskan apa itu random forest, sertakan gambar ilustrasi buatan sendiri.
2. Jelaskan cara membaca dataset kasus dan artikan makna setiap file dan isi field masing masing file.
3. Jelaskan apa itu cross validation
4. Jelaskan apa arti score 44% pada random forest, 27% pada decision tree dan 29%dari SVM.
5. Jelaskan bagaimana cara membaca confusion matriks dan contohnya memakai gambar atau ilustrasi sendiri.
6. Jelaskan apa itu voting pada random forest disertai dengan ilustrasi gambar sendiri.

3.3 Praktek Program

Tugas anda adalah,praktekkan dan jelaskan dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat dan wajib skrinsut dari komputer sendiri masing masing nomor di bawah ini(nilai 5 masing masing pada hari kedua).

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

```
In [6]: runcell('1. Buat aplikasi sederhana menggunakan pandas', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/praktek_program.py')
      Nama  Umur  Alamat
0  Helmi  Salsabila  21  Cirebon
1      Helmi  20  Bandung
2      Salsabila  19  Jakarta
3      Hells  18  Bogor
4      Sabil  21  Sukabumi
```

Figure 3.1: Aplikasi Pandas - Chapter3

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

```
In [7]: runcell('2. Buat aplikasi sederhana menggunakan numpy', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/praktek_program.py')
      [1. 1.22222222 1.44444444 1.66666667 1.88888889 2.11111111
      2.33333333 2.55555556 2.77777778 3.]
```

Figure 3.2: Aplikasi Numpy - Chapter3

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

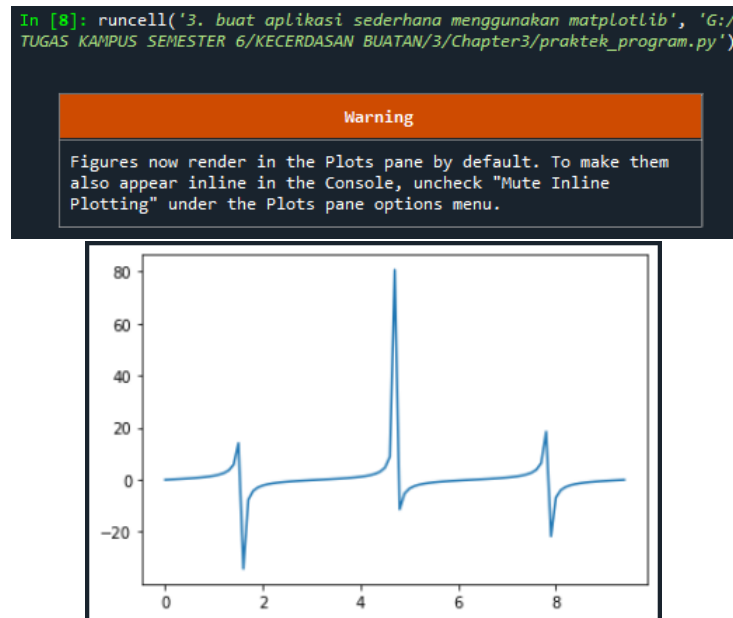


Figure 3.3: Aplikasi Matplotlib - Chapter3

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

```
In [12]: runcell('Membaca file txt', 'G:/TUGAS KAMPUS SEMESTER 6/
KECERDASAN BUATAN/3/Chapter3/chapter3.py')
G:\TUGAS KAMPUS SEMESTER 6\KECERDASAN BUATAN\3\Chapter3\chapter3.py:25:
FutureWarning: The error_bad_lines argument has been deprecated and will
be removed in a future version. Use on_bad_lines in the future.

imgatt = pd.read_csv("data/CUB_200_2011/attributes/
image_attribute_labels.txt",
G:\TUGAS KAMPUS SEMESTER 6\KECERDASAN BUATAN\3\Chapter3\chapter3.py:25:
FutureWarning: The warn_bad_lines argument has been deprecated and will
be removed in a future version. Use on_bad_lines in the future.

imgatt = pd.read_csv("data/CUB_200_2011/attributes/
image_attribute_labels.txt",
imgid attid present
0      1      1      0
1      1      2      0
2      1      3      0
3      1      4      0
4      1      5      1
...
3677851 11788 308      1
3677852 11788 309      0
3677853 11788 310      0
3677854 11788 311      0
3677855 11788 312      1
[3677856 rows x 3 columns]

In [14]: runcell('Melihat sebagian data awal/teratas', 'G:/TUGAS KAMPUS
SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/chapter3.py')
imgid attid present
0      1      1      0
1      1      2      0
2      1      3      0
3      1      4      0
4      1      5      1

In [16]: runcell('Mengetahui jumlah data', 'G:/TUGAS KAMPUS SEMESTER 6/
KECERDASAN BUATAN/3/Chapter3/chapter3.py')
(3677856, 3)

In [18]: runcell('Pivot dataset', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN
BUATAN/3/Chapter3/chapter3.py')
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      0      0      0      0      0      0      ... 0      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] (11788, 312)
```

```

In [20]: runcell('Membaca dataset label file txt', 'G:/TUGAS KAMPUS
SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/chapter3.py')
label
imgid
1      1
2      1
3      1
4      1
5      1 (11788, 1)

In [21]: runcell('Menggabungkan field dari dua file terpisah', 'G:/TUGAS
KAMPUS SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/chapter3.py')
1 2 3 4 5 6 7 8 9 ... 305 306 307 308 309 310 311
312 label
imgid
6362 0 0 0 0 0 0 1 0 0 ... 1 0 0 0 0 0 0
1 109
3102 0 0 0 0 0 0 0 0 1 ... 0 0 0 0 0 0 0
0 54
3654 0 1 0 0 0 0 0 0 0 ... 0 1 0 0 1 0 0
0 63
5105 0 0 0 0 0 0 1 0 0 ... 0 0 0 1 0 0 1
0 88
1928 0 0 0 0 0 0 0 1 0 ... 0 0 0 0 0 1 0
0 34
... ..
... ...
3675 0 0 0 0 1 0 0 0 0 ... 0 1 0 0 1 0 0
0 64
5810 1 0 0 0 0 0 0 0 0 ... 1 0 0 0 0 0 0
0 100
3386 0 0 0 0 0 1 0 0 0 ... 0 1 0 0 1 0 0
0 59
1008 0 0 0 0 0 0 1 0 0 ... 1 0 0 0 0 0 0
1 19
11177 0 1 0 0 0 0 0 0 0 ... 1 0 0 0 0 1 0
0 190

[11788 rows x 313 columns]

```

```

In [23]: runcell('Memisahkan dan memilih label', 'G:/TUGAS KAMPUS SEMESTER 6/
KECERDASAN BUATAN/3/Chapter3/chapter3.py')

```

	1	2	3	4	5	6	7	...	306	307	308	309	310	311	312
imgid								...							
6362	0	0	0	0	0	0	1	...	0	0	0	0	0	0	1
3102	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3654	0	1	0	0	0	0	0	...	1	0	0	1	0	0	0
5105	0	0	0	0	0	0	1	...	0	0	1	0	0	1	0
1928	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0
...
3675	0	0	0	0	1	0	0	...	1	0	0	1	0	0	0
5810	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3386	0	0	0	0	0	1	0	...	1	0	0	1	0	0	0
1008	0	0	0	0	0	0	1	...	0	0	0	0	0	0	1
11177	0	1	0	0	0	0	0	...	0	0	0	0	1	0	0

```

[11788 rows x 312 columns]
label
imgid
6362    109
3102     54
3654     63
5105     88
1928     34
...
3675     64
5810    100
3386     59
1008     19
11177    190

[11788 rows x 1 columns]

```

```

In [25]: runcell('Melihat isi masing masing data frame', 'G:/TUGAS KAMPUS
SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/chapter3.py')

```

	1	2	3	4	5	6	7	...	306	307	308	309	310	311	312
imgid								...							
6362	0	0	0	0	0	0	1	...	0	0	0	0	0	0	1
3102	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3654	0	1	0	0	0	0	0	...	1	0	0	1	0	0	0
5105	0	0	0	0	0	0	1	...	0	0	1	0	0	1	0
1928	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0

```

[5 rows x 312 columns]
label
imgid
6362    109
3102     54
3654     63
5105     88
1928     34

```



```

In [27]: runcell('Pembagian data training dan test', 'G:/TUGAS KAMPUS SEMESTER 6/
KECERDASAN BUATAN/3/Chapter3/chapter3.py')
imgid
6362    109
3102     54
3654     63
5105     88
1928     34
...
1351     25
3928     68
4659     80
435       9
6191    106
Name: label, Length: 8000, dtype: int64 imgid
3198     56
9742    166
11155   190
10961   186
8923    152
...
3675     64
5810    100
3386     59
1008     19
11177   190
Name: label, Length: 3788, dtype: int64

In [32]: runcell('Instansiasi kelas Random Forest & Fitting random forest
dengan dataset training & Melihat Hasil prediksi', 'G:/TUGAS KAMPUS SEMESTER
6/KECERDASAN BUATAN/3/Chapter3/chapter3.py')
[109  54  63  88  34] RandomForestClassifier(max_features=50, random_state=0)
0.44165786694825765

```

Figure 3.4: Klasifikasi Random Fores - Chapter3

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

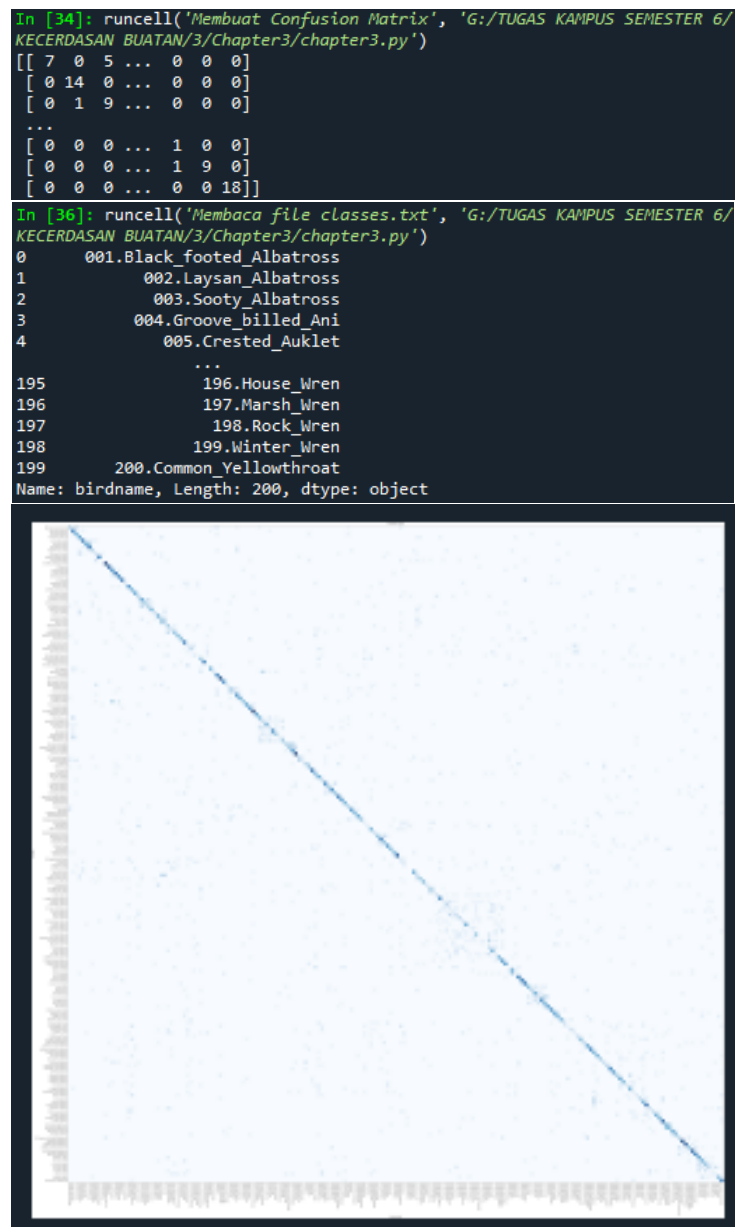


Figure 3.5: Confusion Matrix - Chapter3

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.

```
In [40]: runcell('Mencoba klasifikasi dengan decission tree dengan dataset  
yang sama', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/  
chapter3.py')  
0.2721752903907075  
In [41]: runcell('Mencoba klasifikasi dengan SVM dengan dataset yang sama',  
'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/chapter3.py')  
0.47729672650475186
```

Figure 3.6: Klasifikasi SVM - Chapter3

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

```
In [43]: runcell('Hasil Cross Validation random forest', 'G:/TUGAS KAMPUS  
SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/chapter3.py')  
Accuracy: 0.44 (+/- 0.01)  
In [44]: runcell('Hasil Cross Validation Decission Tree', 'G:/TUGAS KAMPUS  
SEMESTER 6/KECERDASAN BUATAN/3/Chapter3/chapter3.py')  
Accuracy: 0.26 (+/- 0.02)  
In [45]: runcell('Hasil Cross Validation SVM', 'G:/TUGAS KAMPUS SEMESTER 6/  
KECERDASAN BUATAN/3/Chapter3/chapter3.py')  
Accuracy: 0.47 (+/- 0.01)
```

Figure 3.7: Cross Validaiton - Chapter3

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

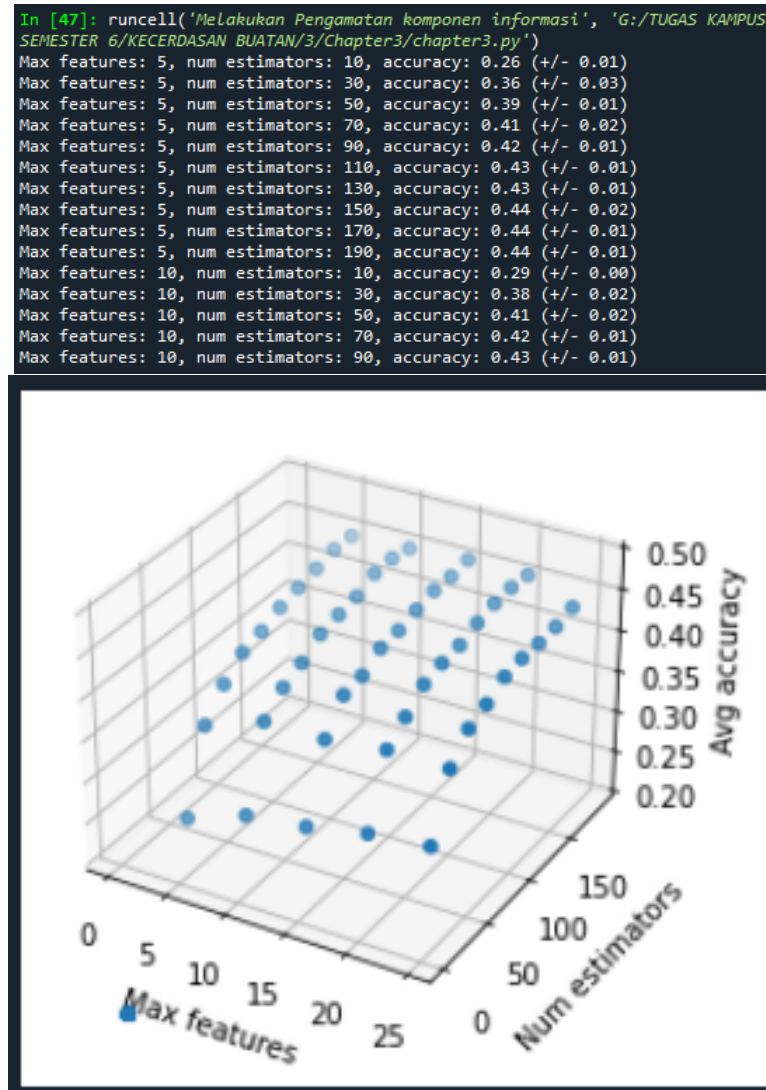


Figure 3.8: Komponen Informasi - Chapter3

3.4 Penanganan Error

Dari percobaan yang dilakukan di atas, error yang kita dapatkan di dokumentasikan dan di selesaikan (nilai 5 per error yang ditangani. Untuk hari kedua):

1. skrinshot error
2. Tuliskan kode error dan jenis errornya

3. Solusi pemecahan masalah error tersebut

3.5 Presentasi Tugas

Pada pertemuan ketiga ini, diadakan tiga penilaiain yaitu penilaian untuk tugas mingguan seperti sebelumnya dengan nilai maksimal 100. Kemudian dalam satu minggu kedepan maksimal sebelum waktu mata kuliah kecerdasan buatan. Ada presentasi tugas bab ini dan bab sebelumnya dengan nilai presentasi yang terpisah masing-masing 100. Jadi ada tiga komponen penilaiain pada pertemuan ini yaitu :

1. tugas minggu hari ini dan besok (maks 100). pada chapter ini
2. presentasi decission tree (maks 100). Mempraktekkan kode python dan menjelaskan cara kerjanya.
3. presentasi Random Forest (maks 100).Mempraktekkan kode python dan menjelaskan cara kerjanya.

Waktu presentasi pada jam kerja di IRC. Kriteria penilaian presentasi sangat sederhana, jika presenter tidak bisa menjawab pertanyaan asisten maka nilai nol. Jika semua pertanyaan bisa dijawab maka nilai 100. Presentasi bisa diulang apabila nilai nol sampai bisa mendapatkan nilai 100 dalam waktu satu minggu kedepan.

Chapter 4

Klasifikasi Teks

Untuk pratikum saati ini menggunakan buku *Python Artificial Intelligence Projects for Beginners*[?]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Kode program ada di <https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners> . Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti implementasi klasifikasi pada teks
2. Mengerti teknik machine learning
3. Memahami Bag of Words

Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten riset. Kode program menggunakan input listing ditaruh di folder src ekstensi .py dan dipanggil ke latex dengan input listings. Tulisan dan kode tidak boleh plagiat, menggunakan bahasa indonesia yang sesuai dengan gaya bahasa buku teks.

4.1 Teori

Menggunakan teknik bag-of-words pada klasifikasi berbasis text dan kata untuk mengklasifikasikan komentar yang ada di internet sebagai spam atau bukan. Atau bisa juga untuk melakukan identifikasi sebuah review apakah positive atau negatif.

4.1.1 Vektorisasi data

Pertama kita lakukan vektorisasi dari dataset. Lankah pertama kita baca terlebih dahulu dengan perintah 4.1.

```

1 import pandas as pd
2 d = pd.read_csv("Youtube01-Psy.csv")

```

Listing 4.1: Membaca data file txt

Memanggil library vektorisasi dari sci-kit lern, dengan menggunakan listing 4.2.

```

1 from sklearn.feature_extraction.text import CountVectorizer
2 vectorizer = CountVectorizer()

```

Listing 4.2: Instansiasi Vektorizer

Memilih kolom CONTENT dari dataframe d untuk di vektorisasi kemudian menam-pungnya pada variabel dvec menggunakan listing 4.3.

```

1 dvec = vectorizer.fit_transform(d['CONTENT'])
2 dvec

```

Listing 4.3: Vektorisasi data dari atribut CONTENT

Melihat daftar kata yang di vektorisasi. lalu kita simpan isinya pada variabel daptarkata dengan menggunakan perintah pada listing 4.4.

```

1 daptarkata=vectorizer.get_feature_names()

```

Listing 4.4: Mendapatkan Daftar Kata

Lakukan pengocokan data sehingga data terlihat random, perintahnya ada di list-ing 4.5. Lalu kita cek kembali datanya pada variabel dshuf.

```

1 dshuf = d.sample(frac=1)

```

Listing 4.5: Mengocok Data Frame

kemudian kita melakukan pemisahan antara data set untuk training dan test dengan perintah di listing 4.6.

```

1 d_train=dshuf[:300]
2 d_test=dshuf[300:]

```

Listing 4.6: Memisahkan data frame

Kita lakukan training perintah listing 4.7.

```

1 d_train_att=vectorizer.fit_transform(d_train['CONTENT'])
2 d_train_att

```

Listing 4.7: Training pada vektorisasi atau yang disebut transform dan fit

Lalu kita lakukan transformasi saja tanpa training pada data testing dengan per-intah listing 4.8.

```

1 d_test_att=vectorizer.transform(d_test['CONTENT'])
2 d_test_att

```

Listing 4.8: Transform tanpa fit dari data testing

Pengambilan label klasifikasi spam dari kolom CLASS dengan perintah listing 4.9.

```

1 d_train_label=d_train[ 'CLASS' ]
2 d_test_label=d_test[ 'CLASS' ]

```

Listing 4.9: Pengambilan label dari data testing dan training

4.1.2 Klasifikasi dengan Random Forest

Setelah lakukan vektorisasi. Kita panggil kelas RandomForestClassifier. dengan n estimators sebanyak 80 yang artinya kita akan membuat 80 tree dengan tanpa batasan pengambilan atribut atau kolom dengan perintah listing 4.10.

```

1 from sklearn.ensemble import RandomForestClassifier
2 clf=RandomForestClassifier(n_estimators=80)

```

Listing 4.10: Instansiasi kelas Random Forest

Kemudian lakukan fit untuk membangun random forest yang sudah ditentukan dengan banyak tree sebanyak 80 dengan perintah listing 4.11.

```

1 clf.fit(d_train_att, d_train_label)

```

Listing 4.11: Fitting random forest dengan dataset training

Hasilnya bisa kita lakukan prediksi dari data testing dengan perintah listing 4.12.

```

1 clf.predict(d_test_att)

```

Listing 4.12: Melihat Hasil prediksi

Untuk besaran skornya dengan perintah listing 4.13

```

1 clf.score(d_test_att, d_test_label)

```

Listing 4.13: Score perolehan dari klasifikasi

4.1.3 Confusion Matrix

Dari RF kita coba petakan ke dalam Confusion Matrix dan lihat hasilnya dengan perintah listing 4.14.

```

1 from sklearn.metrics import confusion_matrix
2 pred_labels = clf.predict(d_test_att)
3 cm=confusion_matrix(d_test_label, pred_labels)

```

Listing 4.14: Membuat Confusion Matrix

4.1.4 Pengecekan Cross Validation

Pengecekan Cross Validation untuk random forest dengan perintah 4.15.


```

1 from sklearn.model_selection import cross_val_score
2 scores = cross_val_score(clf, d_train_att, d_train_label, cv=5)
3
4 skorrata2=scores.mean()
5 skoresd=scores.std()

```

Listing 4.15: Hasil Cross Validation random forest

4.2 Soal Teori

Praktek teori penunjang yang dikerjakan(nilai 5 per nomor, untuk hari pertama) :

1. Jelaskan apa itu klasifikasi teks, sertakan gambar ilustrasi buatan sendiri.
2. Jelaskan mengapa klasifikasi bunga tidak bisa menggunakan machine learning, sertakan ilustrasi sendiri.
3. Jelaskan bagaimana teknik pembelajaran mesin pada teks pada kata-kata yang digunakan di youtube,jelaskan arti per atribut data csv dan sertakan ilustrasi buatan sendiri.
4. Jelaskan apa yang dimaksud vektorisasi data.
5. Jelaskan apa itu bag of words dengan kata-kata yang sederhana dan ilustrasi sendiri.
6. Jelaskan apa itu TF-IDF, ilustrasikan dengan gambar sendiri.

4.3 Praktek Program

Tugas anda adalah,praktekkan dan jelaskan dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat dan wajib skrinsut dari komputer sendiri masing masing nomor di bawah ini(nilai 5 masing masing pada hari kedua).

1. buat aplikasi sederhana menggunakan pandas, buat data dummy format csv sebanyak 500 baris dan melakukan load ke dataframe panda.jelaskan arti setiap baris kode yang dibuat(harus beda dengan teman sekelas)

```
In [53]: runcell('Aplikasi sederhana dengan menggunakan pandas, data dummy
csv 500 / 7391 baris', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/4/
Chapter04/praktek_program.py')
classification,"report_date","city","distribution_pattern","openfda_brand_n
ame","openfda_generic_name","state"
0 Class III,"6/20/2012","Chadds Ford","Nationwid...
1 Class II,"6/20/2012","Miami","Nationwide and P...
2 Class II,"6/20/2012","Miami","Nationwide and P...
3 Class II,"6/20/2012","Miami","Nationwide and P...
4 Class II,"6/20/2012","Bedford","Nationwide and...
```

Figure 4.1: Aplikasi Pandas - Chapter4

2. dari dataframe tersebut dipecah menjadi dua dataframe yaitu 450 row pertama dan 50 row sisanya(harus beda dengan teman sekelas)

```
In [54]: runcell('Dataframe 450 row pertama', 'G:/TUGAS KAMPUS SEMESTER 6/
KECERDASAN BUATAN/4/Chapter04/praktek_program.py')
classification,"report_date","city","distribution_pattern","openfda_brand
_name","openfda_generic name","state"
0 Class III,"6/20/2012","Chadds Ford","Nationwid...
1 Class II,"6/20/2012","Miami","Nationwide and P...
2 Class II,"6/20/2012","Miami","Nationwide and P...
3 Class II,"6/20/2012","Miami","Nationwide and P...
4 Class II,"6/20/2012","Bedford","Nationwide and...
...
445 Class II,"1/9/2013","Framingham","Nationwide",...
446 Class II,"1/9/2013","Framingham","Nationwide",...
447 Class II,"1/9/2013","Framingham","Nationwide",...
448 Class I,"1/9/2013","Framingham","Nationwide",...
449 Class II,"1/9/2013","Framingham","Nationwide",...
...
[450 rows x 1 columns]
classification,"report_date","city","distribution_pattern","openfda_brn
d_name","openfda_generic name","state"
450 Class II,"1/9/2013","Framingham","Nationwide",...
451 Class II,"1/9/2013","Framingham","Nationwide",...
452 Class II,"1/9/2013","Framingham","Nationwide",...
453 Class II,"1/9/2013","Framingham","Nationwide",...
454 Class I,"1/9/2013","Framingham","Nationwide",...
...
7385 Class II,"2/21/2018","Cary","Distributed Natio...
7386 Class II,"2/21/2018","Cary","Distributed Natio...
7387 Class II,"2/21/2018","Cary","Distributed Natio...
7388 Class II,"2/21/2018","Cary","Distributed Natio...
7389 Class II,"2/21/2018","Cary","Distributed Natio...
...
[6940 rows x 1 columns]
```

Figure 4.2: Dataframe dipecah menjadi dua - Chapter4

- pratekkan vektorisasi dan klasifikasi dari data (NPM mod 4, jika 0 maka katty perry, 1 LMFAO, 2 Eminem, 3 Shakira) dengan Decission Tree. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```
In [7]: runcell('Membaca Data File 1194018 mod 4 = 2 (eminem)', 'G:/TUGAS
KAMPUS SEMESTER 6/KECERDASAN BUATAN/4/Chapter04/chapter04.py')
COMMENT_ID ... CLASS
0 z12rwnfnyrbsefonb232i5ehdxzkjzjs2 ... 1
1 z130wpmwmyuetyxcn23xf5k5ynmkdpjrj04 ... 0
2 z13vsfqirtavju0t22ezrgzyorwxhpf3 ... 1
3 z12wjzc4eprnvja4304cgbbizuved35wxc ... 0
4 z13xjfr42z3uxdz2223gx5rrzs3dt5hna ... 1
.. ... ..
443 LneaDw26bFu3RCmyrWp9S6wh1h9dBv3X95g1HzKAb4 ... 1
444 LneaDw26bFsD65dtIvAE0bWYIYnFTqQDKBek_Ypz3J8 ... 1
445 LneaDw26bFuvs-8okwLpAfa6g3QHpwD8k7sbbMP3Bg8 ... 1
446 z12hfp2wmyuqztkw504cgblyxtbsxjuzeow0k ... 0
447 z13tsbc5vvn0hdozz04chjt51lq1cvris0k ... 0
[448 rows x 5 columns]
```

Figure 4.3: Vektorisasi dan Klasifikasi 1194018 = mod 2 - Chapter4

- Cobalah klasifikasikan dari data vektorisasi yang di tentukan di nomor sebelumnya dengan klasifikasi SVM. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```
In [8]: runcell('Instalasi Vektorizer & Vektorisasi data dari
atribut CONTENT', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/4/
Chapter04/chapter04.py')
(0, 58) 1
(0, 839) 1
(0, 574) 1
(0, 1358) 1
(0, 1404) 1
(0, 879) 1

In [10]: runcell('Mendapatkan Daftar Kata', 'G:/TUGAS KAMPUS
SEMESTER 6/KECERDASAN BUATAN/4/Chapter04/chapter04.py')
['00', '000', '047000', '09', '10', '100', '1000',
'100877300245414', '11', '12', '12year', '1337', '14', '15', '16',
'17', '17yr', '18', '19', '1990', '1b', '1billion', '200', '2005',
'2008', '200k', '2010', '2013', '2014', '2015', '21', '23',
'23active', '23awesome', '23eminem', '23king',
'23lovethewayyoulie', '23rapgod', '23rt', '23share', '25', '26',
'26t22', '28', '2asfn9shghk', '2b4wywphi8c', '2nd', '2zme8f',
'30', '31st', '365', '36loseweight', '39', '3rd', '4000',
'4000dollars', '41', '43', '447935454150', '45', '46', '476000',
'482', '490', '4e', '4th', '50', '500', '500k', '5tu9gn11310',
'60', '600', '600m', '613000', '6_h0m5sayho', '710000', '7k',
'800', '832000', '857', '87', '90', '940', 'abbas', 'ablaze',
'about', 'above', 'absolutely', 'abuse', 'abuses', 'abusive',
'abusue', 'acaer', 'acceptance', 'accidental', 'accomplished',
'achieve', 'across', 'act', 'activates', 'active', 'actual',
'actually', 'ad', 'adam', 'adapt', 'adhoc', 'admitting', 'adult',
'advance', 'advertise', 'advertisements', 'afraid', 'aftermath',
'again', 'ago', 'agree', 'agreeable', 'aimbwbfbqzbg', 'album',
```

```

In [11]: runcell('Mengocok Data Frame & Memisahkan data frame',
               'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/4/Chapter04/
               chapter04.py')

```

	COMMENT_ID	...	CLASS
172	z13kch2hprqkjhr5d04cdlwzkpeuwxljhhs	...	0
436	LneaDw26bFsfwQ8cbaeKwCs8B7EA1zRf2nBgT9NyY	...	1
168	z13kfpwrhzeuvfavb221yxoj3ynsi5vj004	...	0
371	z13ihhrakmndxb1zo04chpp53nn2xbja1fc	...	0
276	LneaDw26bFvaB1GMY4DfxNzifyIU7G0Wp8Br9oySEk	...	1
..
65	z13cxvn4hyjbyjiab04cepiynzaihpmpp24	...	0
59	LneaDw26bFshLT1-9YrPxWqBLzAdCIz2qsKDoIM2SDY	...	1
71	z124ipdjxtumsz10c23schdzqpb0xvczx04	...	1
82	z13xcrw5xrregt5gu04cg5rrt13mwdkp33o	...	0
132	z12vilga4u3yjpvs23pst05vsv0xtikh	...	0

```

[300 rows x 5 columns]
COMMENT_ID ... CLASS
66          z12yzzqoozqieho4x04ci12qbrzhzyyvvg ... 0
209         z12uxprwxwalxm10knzuxzxo5nyichlk1 ... 1
431         z13zjp3povzssp3xi04cgvtgwpfiy1faigo ... 0
8          z12gxdortqzwhhqas04cfjrwituzghb5tvk0k ... 1
205        LneaDw26bFsgggeU_vbp6eynA9RVJ0qto20Mt6tcgjk ... 1
..         ... ..
427        LneaDw26bFtn6YS8GRfHBXoaZZI5newNd0njN0SeF84 ... 1
155        z121i1eqppzph3eod04cixfgwknydnfzq3k ... 0
343        LneaDw26bFukZRaKv1u8jqanxsDr6RJraSjf2Ffe1_Q ... 1
311        z13ucdtaSub5djl1d235wndifoybg5lkt ... 1
370        z12kf5dyltrijhepb04cc1tibzijd5walmc0k ... 0

[148 rows x 5 columns]
In [13]: runcell('Training pada vektorisasi atau yang disebut
               transform dan fit', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/
               4/Chapter04/chapter04.py')

```

(0, 668)	1
(0, 1112)	1
(0, 1021)	1
(1, 1021)	1
(1, 520)	1
(1, 558)	1
(1, 9)	1

```

In [15]: runcell('Transform tanpa fit dari data testing', 'G:/
               TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/4/Chapter04/
               chapter04.py')

```

(0, 25)	1
(0, 205)	1
(0, 558)	1
(0, 641)	1
(0, 1051)	1
(0, 1112)	1
(0, 1122)	1

```

In [17]: runcell('Pengambilan label dari data testing dan
training', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/4/
Chapter04/chapter04.py')
172    0
436    1
168    0
371    0
276    1
..
65     0
59     1
71     1
82     0
132    0
Name: CLASS, Length: 300, dtype: int64 66      0
209    1
431    0
8      1
205    1
..
427    1
155    0
343    1
311    1
370    0
Name: CLASS, Length: 148, dtype: int64

In [19]: runcell('SVM', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN
BUATAN/4/Chapter04/chapter04.py')
0.918918918918919

```

Figure 4.4: Vektorisasi dan Klasifikasi - Chapter4

5. Cobalah klasifikasikan dari data vektorisasi yang di tentukan di nomor sebelumnya dengan klasifikasi Decission Tree. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```

In [18]: runcell('Decision Tree', 'G:/TUGAS KAMPUS SEMESTER 6/
KECERDASAN BUATAN/4/Chapter04/chapter04.py')
0.9527027027027027

```

Figure 4.5: Vektorisasi dan Klasifikasi Decission Tree - Chapter4

6. Plotlah confusion matrix dari praktek modul ini menggunakan matplotlib. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```

In [28]: runcell('Melihat Hasil prediksi & Membuat Confusion
Matrix', 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/4/
Chapter04/chapter04.py')
[[69  1]
 [ 8 70]]

```

Figure 4.6: Confusion Matrix - Chapter4

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

```

In [21]: runcell('Cross validation', 'G:/TUGAS KAMPUS SEMESTER 6/
KECERDASAN BUATAN/4/Chapter04/chapter04.py')
Accuracy: 0.93 (+/- 0.02)

In [22]: runcell('cv untuk dt', 'G:/TUGAS KAMPUS SEMESTER 6/
KECERDASAN BUATAN/4/Chapter04/chapter04.py')
Accuracy: 0.94 (+/- 0.02)

In [23]: runcell('cv untuk svm', 'G:/TUGAS KAMPUS SEMESTER 6/
KECERDASAN BUATAN/4/Chapter04/chapter04.py')
Accuracy: 0.94 (+/- 0.02)

In [29]: runcell('Hasil Cross Validation random forest', 'G:/TUGAS
KAMPUS SEMESTER 6/KECERDASAN BUATAN/4/Chapter04/chapter04.py')
0.9433333333333334 0.02494438257849293

```

Figure 4.7: Cross Validaiton - Chapter4

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

```

In [30]: runcell('Komponen informasi', 'G:/TUGAS KAMPUS SEMESTER
6/KECERDASAN BUATAN/4/Chapter04/chapter04.py')
Max features: 1, num estimators: 2, accuracy: 0.80 (+/- 0.10)
Max features: 1, num estimators: 6, accuracy: 0.88 (+/- 0.06)
Max features: 1, num estimators: 10, accuracy: 0.87 (+/- 0.08)
Max features: 1, num estimators: 14, accuracy: 0.90 (+/- 0.06)
Max features: 1, num estimators: 18, accuracy: 0.94 (+/- 0.07)
Max features: 1, num estimators: 22, accuracy: 0.93 (+/- 0.04)
Max features: 1, num estimators: 26, accuracy: 0.93 (+/- 0.06)
Max features: 1, num estimators: 30, accuracy: 0.94 (+/- 0.06)
Max features: 1, num estimators: 34, accuracy: 0.94 (+/- 0.03)
Max features: 1, num estimators: 38, accuracy: 0.93 (+/- 0.03)

```

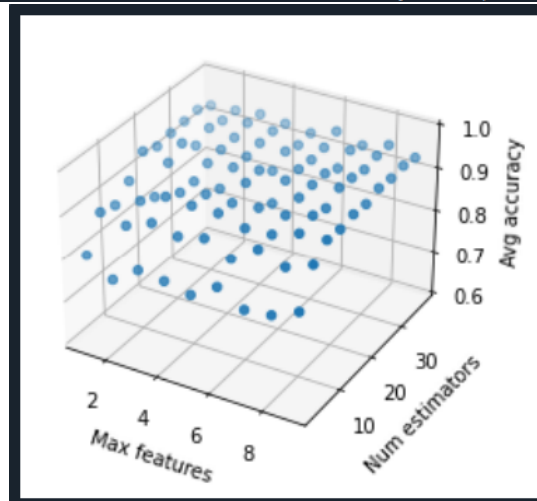


Figure 4.8: Komponen Informasi - Chapter4

4.4 Penanganan Error

Dari praktek pemrograman yang dilakukan di modul ini, error yang kita dapatkan (hasil komputer sendiri) di dokumentasikan dan di selesaikan (nilai 5 per error yang ditangani. Untuk hari kedua):

1. skreensut error
2. Tuliskan kode eror dan jenis errornya
3. Solusi pemecahan masalah error tersebut

4.5 Presentasi Tugas

Pada pertemuan ketiga ini, diadakan tiga penilaian yaitu penilaian untuk tugas mingguan seperti sebelumnya dengan nilai maksimal 100. Kemudian dalam satu minggu kedepan maksimal sebelum waktu mata kuliah kecerdasan buatan. Ada presentasi kematerian dengan nilai presentasi yang terpisah masing-masing 100. Jadi ada tiga komponen penilaian pada pertemuan ini yaitu :

1. tugas minggu hari ini dan besok (maks 100). pada chapter ini
2. presentasi Vektorisasi (maks 100). Mempraktekkan kode python dan menjelaskan cara kerjanya.
3. presentasi cara kerja Data Frame di Pandas (maks 100).Mempraktekkan kode python dan menjelaskan cara kerjanya.

Waktu presentasi pada jam kerja di IRC. Kriteria penilaian presentasi sangat sederhana, presenter akan ditanyai 20 pertanyaan tentang pemahamannya menggunakan python untuk kecerdasan buatan. jika presenter tidak bisa menjawab satu pertanyaan asisten maka nilai nol. Jika semua pertanyaan bisa dijawab maka nilai 100. Presentasi bisa diulang apabila gagal, sampai bisa mendapatkan nilai 100 dalam waktu satu minggu kedepan.