

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

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

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

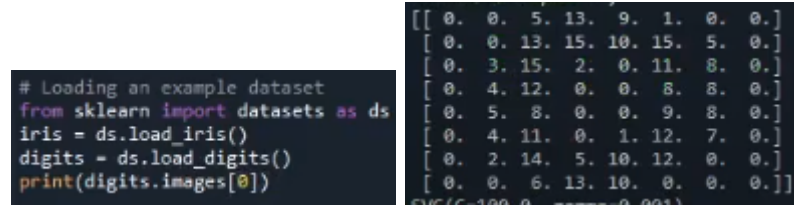
(SesiBelajar) C:\Users\Helmi Salsabila>

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)

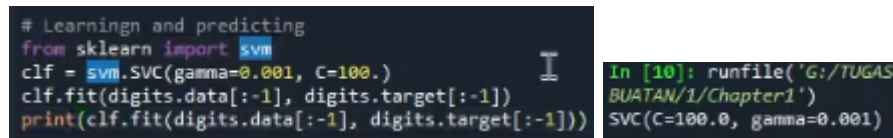


```
# Loading an example dataset
from sklearn import datasets as ds
iris = ds.load_iris()
digits = ds.load_digits()
print(digits.images[0])
```

```
[[ 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)

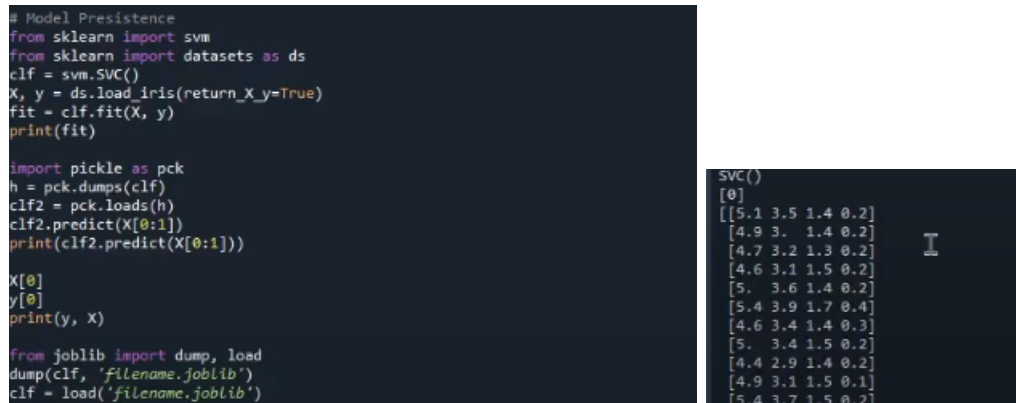


```
# Learningn and predicting
from sklearn import svm
clf = svm.SVC(gamma=0.001, C=100.)
clf.fit(digits.data[:-1], digits.target[:-1])
print(clf.fit(digits.data[:-1], digits.target[:-1]))
```

```
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]))

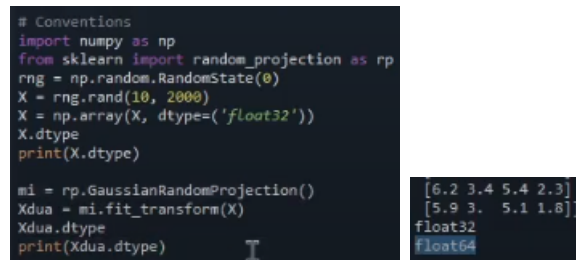
X[0]
y[0]
print(y, x)
```

```
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:

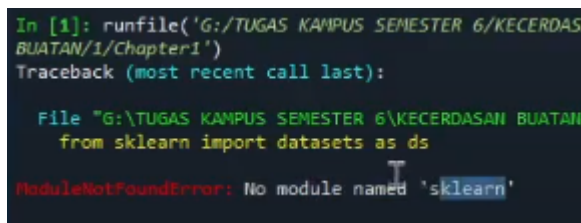
```
[[6.2 3.4 5.4 2.3]
 [5.9 3. 5.1 1.8]]
```

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.

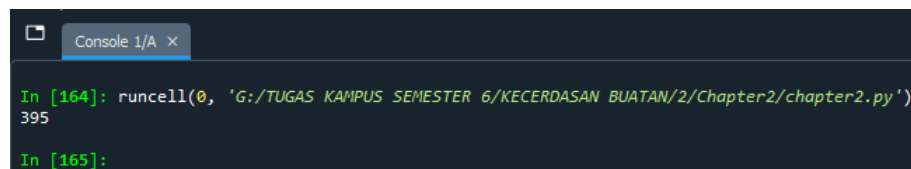
2.2 scikit-learn

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)



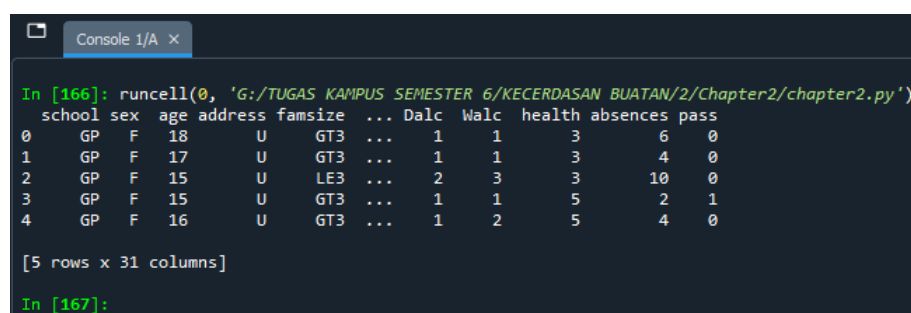
```

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]:

```

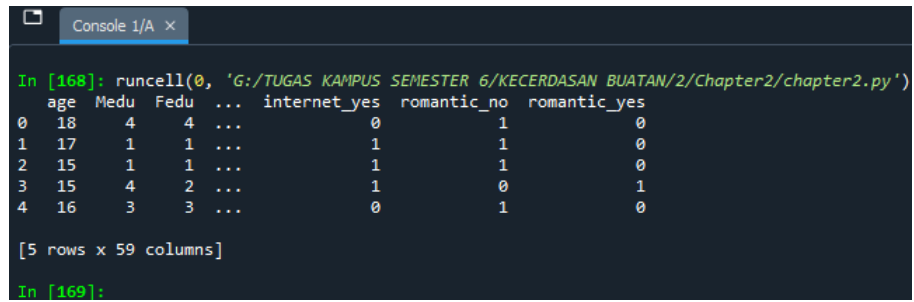
Figure 2.2: Generate binary label

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()

```



```

In [168]: runcell(0, 'G:/TUGAS KAMPUS SEMESTER 6/KECERDASAN BUATAN/2/Chapter2/chapter2.py')

```

	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

```

[5 rows x 59 columns]

In [169]:

```

Figure 2.3: Generate binary label

```

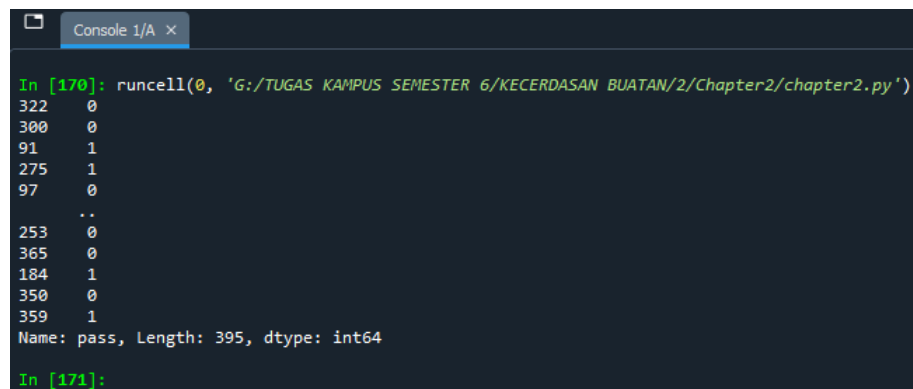
4. # shuffle rows
d = d.sample(frac=1)
# split training and testing data
d_train = d[:500]
d_test = d[500:]

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)))

```

```
Console 1/A x
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)
```

```
Console 1/A x
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
```

```
Console 1/A x
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]\nnclass = fail", fillcolor="#f8dcc9"]
  ;
  1 [label="schoolsup_yes <= 0.5\nsamples = 79.0%\nvalue = [0.519, 0.481]\nnclass = fail",
  fillcolor="#fdf6f0"] ;
  0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
  2 [label="famrel <= 1.5\nsamples = 68.9%\nvalue = [0.474, 0.526]\nnclass = pass", fillcolor="#ecf5fc"] ;
  1 -> 2 ;
  3 [label="samples = 1.5%\nvalue = [0.0, 1.0]\nnclass = pass", fillcolor="#399de5"] ;
  2 -> 3 ;
  4 [label="Mjob_services <= 0.5\nsamples = 67.3%\nvalue = [0.485, 0.515]\nnclass = 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)
```

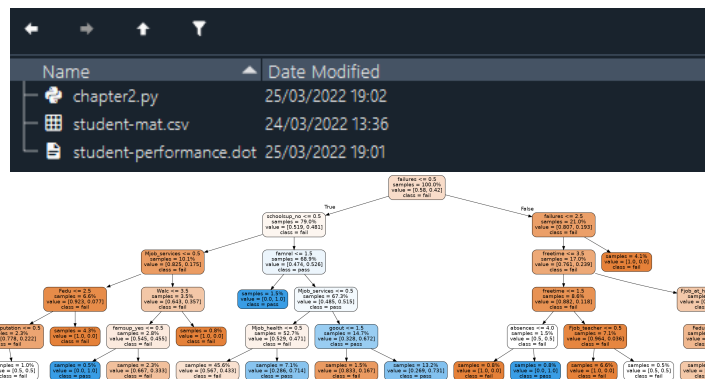


Figure 2.8: Save tree dot to png

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)

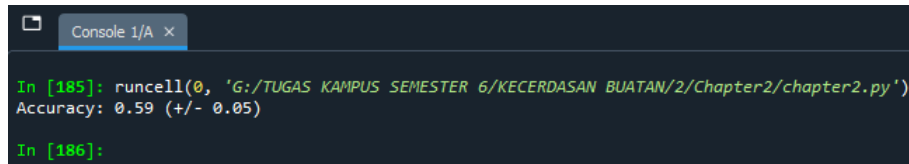
```
Console 1/A x

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

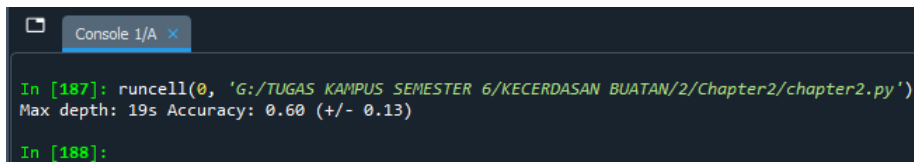

```
# show average score and +/- two standard deviations away
#(covering 95% of scores)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```



```
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()

```

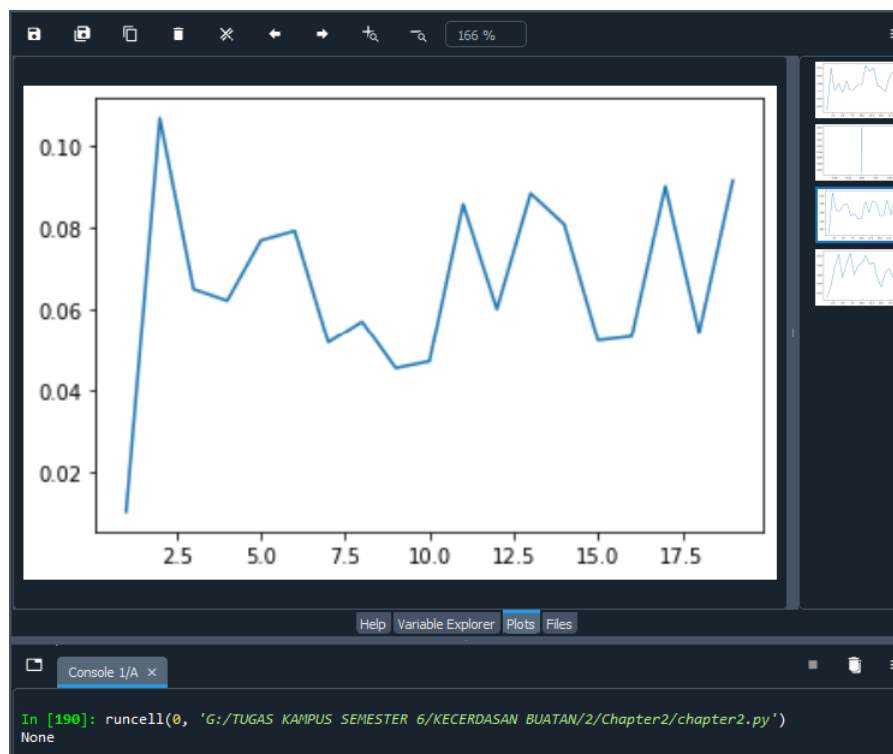


Figure 2.14: Subplots

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