

Modul Praktikum Kecerdasan Buatan



Burhanudin Zuhri

1194008

Applied Bachelor of Informatics Engineering

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

Politeknik Pos Indonesia

Bandung 2021/2022

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

Acknowledgements

Pertama-tama kami panjatkan puji dan syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga Buku Pedoman Tingkat Akhir ini dapat diselesaikan.

Abstract

Buku Pedoman ini dibuat dengan tujuan memberikan acuan, bagi mahasiswa Tingkat Akhir dan dosen Pembimbing. Pada intinya buku ini menjelaskan secara lengkap tentang Standar pengerjaan Intership dan Tugas Akhir di Program Studi D4 Teknik Informatika, dan juga mengatur mekanisme, teknik penulisan, serta penilaiannya. Dengan demikian diharapkan semua pihak yang terlibat dalam aktivitas Bimbingan Mahasiswa Tingkat Akhir berjalan lancar dan sesuai dengan standar.

Contents

1	Mengenal Kecerdasan Buatan dan Scikit-Learn	1
1.1	Teori	1
1.2	Instalasi	4
1.3	Penanganan Error	8
2	Membangun Model Prediksi	9
2.1	Teori	9
2.2	scikit-learn	12
2.3	Penanganan Error	17

List of Figures

Chapter 1

Mengenal Kecerdasan Buatan dan Scikit-Learn

1.1 Teori

Praktek teori penunjang yang dikerjakan :

1. Definisi, Sejarah, dan Perkembangan Kecerdasan Buatan.

- (a) Definisi Kecerdasan Buatan

Kecerdasan Buatan (Artificial Intelligence) adalah salah satu cabang ilmu pada bidang komputer dalam memodelkan atau mensimulasikan kecerdasan manusia ke dalam komputer yang bertujuan memungkinkan suatu sistem untuk belajar dari pengalaman, mengumpulkan dan menyesuaikan input-input data baru, melaksanakan tugas, serta menyelesaikan permasalahan.

- (b) Sejarah dan Perkembangan Kecerdasan Buatan

Sejarah kecerdasan buatan dimulai sejak abad 20 pada tahun 1940-1950, yaitu ditandai dengan mulai terbentuknya komputer modern. Pada tahun 1943, McMulloh dan Pitts mengusulkan model matematis yang diberi nama Perceptron dari neuron di dalam otak otak manusia. Pada tahun 1950, Alan Turing dalam tulisannya yang berjudul Computing Machinery and Intelligence mengeluarkan pernyataan untuk meningkatkan pengembangan Artificial Intelligenc. Pada akhir tahun 1955 program Artifical Intelligence pertama kali muncul berkat adanya perkembangan The Logic Theorist oleh Newell dan Simon. Pada tahun 1956, para ilmuwan mulai berdiskusi mengenai bidang sybernetics, matematika, algoritma dan teori jaringan dan pada tahun yang sama, McCarthy mendirikan Konferensi

Dartmouth di Hanover, New Hampshire dan menemukan beberapa teori kompleks mengenai jaringan saraf dan pemikiran kreatif pada komputer. Pada tahun 1960, terjadi perkembangan pesat yaitu berupa komputer telah dapat menampung lebih banyak informasi dan lebih mudah untuk mendapat akses yang cepat dan murah. Selain itu, beberapa algoritma machine learning sudah mulai digunakan untuk menyelesaikan permasalahan spesifik. Pada tahun 1971-1990, terdapat banyak milestone yang dicapai AI yaitu seperti penggunaan speech recognition software pada Dragon Systems yang diciptakan pada Windows dan disusul dengan munculnya beberapa robot yang mengimplementasikan Artificial Intelligence seperti Deep Blue, Furby, dan RoBOt (AIBO). Pada abad 21 ini, AI terus meningkat, dan informasi seputar AI semakin banyak disebarkan dan korporat juga semakin banyak menggunakan AI untuk mengembangkan machine learning.

2. Definisi Supervised Learning, Klasifikasi, Regresi dan Unsupervised learning. Data Set, Training Set dan Testing Set.

(a) Supervised Learning

Supervised Learning merupakan sebuah pendekatan yang ditentukan berdasarkan penggunaan training set berlabel atau labeled dataset untuk membangun sebuah model yang tingkat akurasi dapat ditingkatkan dari waktu ke waktu. Dengan kata lain, semakin banyak model tersebut mengolah data, maka tingkat keakurasiannya akan semakin tinggi juga. Supervised learning ini juga digunakan untuk melakukan klasifikasi data atau memprediksi hasil secara akurat sesuai dengan output berdasarkan pola yang ada di dalam data training dan berupa data yang memiliki label yang sudah ditentukan.

(b) Unsupervised Learning

Unsupervised Learning merupakan sebuah pendekatan yang ditentukan berdasarkan penggunaan training set yang tidak berlabel yang digunakan untuk menganalisa dan juga mengelompokkan kumpulan data yang tidak berlabel. Unsupervised Learning ini juga digunakan untuk menarik kesimpulan dari dataset dan untuk mempelajari suatu data berdasarkan kedekatannya saja atau yang biasa disebut dengan clustering.

(c) Klasifikasi

Klasifikasi merupakan teknik untuk mengidentifikasi beberapa data yang belum berlabel untuk dikategorikan menjadi sebuah bagian dari kelas diskrit. Klasifikasi ini mempelajari hubungan antara kumpulan variabel fitur dan variabel target.

(d) Regresi

Regresi merupakan suatu teknik untuk mendefinisikan relasi antara dua variabel maupun lebih seperti variabel terikat dan variabel bebas yang bertujuan untuk menemukan suatu fungsi yang dapat memodelkan data dengan meminimalkan error atau selisih antara nilai prediksi dengan nilai yang sebenarnya.

(e) Data Set

Data set merupakan kumpulan data yang berisi informasi-informasi lama, dan dapat dikelola sehingga menjadi sebuah informasi baru.

(f) Training Set

Training set merupakan bagian dari data set yang berfungsi untuk melatih suatu algoritma agar dapat memprediksi sesuatu atau menjalankan fungsi dari algoritma tersebut.

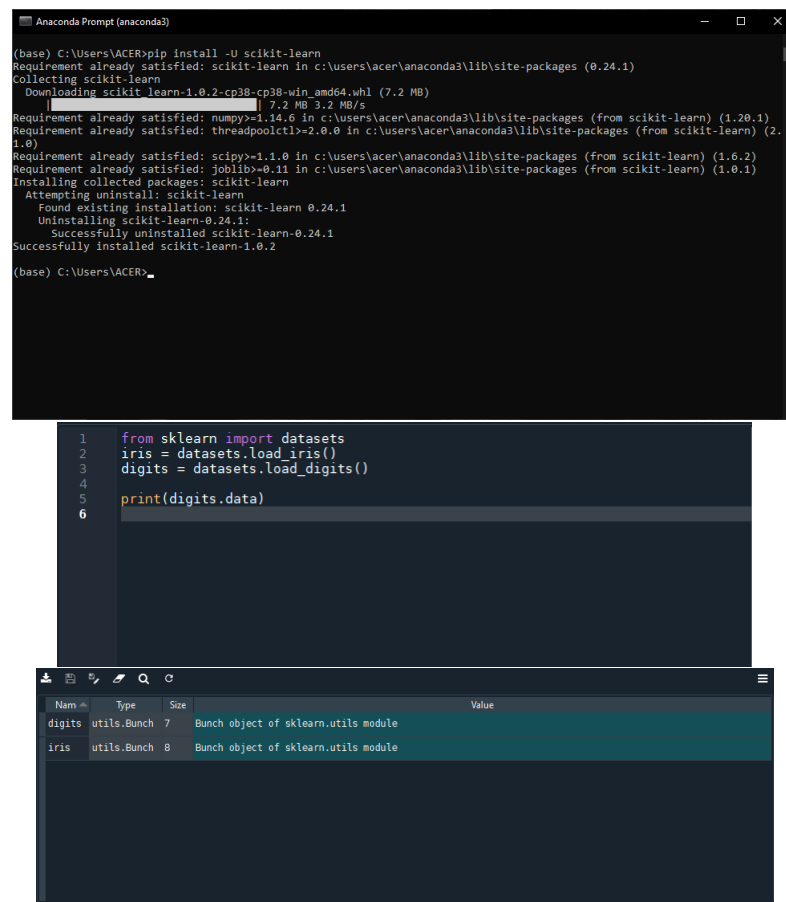
(g) Testing Set

Testing set merupakan bagian dari data set yang digunakan untuk mengetahui akurasi dan performa dari algoritma yang sudah dilatih oleh training set sebelumnya.

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.

1. Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer. Gunakan perintah "pip install -U scikit-learn".



The image shows a sequence of three screenshots from a computer screen. The top screenshot is an Anaconda Prompt window showing the command `pip install -U scikit-learn` being executed. It shows the process of downloading and installing scikit-learn 1.0.2, including dependency checks for numpy, threadpoolctl, and scipy. The middle screenshot is a Jupyter Notebook cell with the following code: `from sklearn import datasets; iris = datasets.load_iris(); digits = datasets.load_digits(); print(digits.data)`. The bottom screenshot is the Jupyter Notebook's variable explorer, showing two variables: `digits` and `iris`, both of type `utils.Bunch` and containing data from the `sklearn.utils` module.

```
(base) C:\Users\VACER>pip install -U scikit-learn
Requirement already satisfied: scikit-learn in c:\users\acer\anaconda3\lib\site-packages (0.24.1)
Collecting scikit-learn
  Downloading scikit_learn-1.0.2-cp38-cp38-win_amd64.whl (7.2 MB)
    | 7.2 MB 3.2 MB/s
Requirement already satisfied: numpy>=1.14.6 in c:\users\acer\anaconda3\lib\site-packages (from scikit-learn) (1.20.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\acer\anaconda3\lib\site-packages (from scikit-learn) (2.1.0)
Requirement already satisfied: scipy>=1.1.0 in c:\users\acer\anaconda3\lib\site-packages (from scikit-learn) (1.6.2)
Requirement already satisfied: joblib>=0.11 in c:\users\acer\anaconda3\lib\site-packages (from scikit-learn) (1.0.1)
Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 0.24.1
    Uninstalling scikit-learn-0.24.1:
      Successfully uninstalled scikit-learn-0.24.1
Successfully installed scikit-learn-1.0.2

(base) C:\Users\VACER>
```

```
1 from sklearn import datasets
2 iris = datasets.load_iris()
3 digits = datasets.load_digits()
4
5 print(digits.data)
6
```

Name	Type	Size	Value
digits	utils.Bunch	7	Bunch object of sklearn.utils module
iris	utils.Bunch	8	Bunch object of sklearn.utils module

2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris.

```
1 from sklearn import datasets
2 iris = datasets.load_iris()
3 digits = datasets.load_digits()
4 #print(digits.data)
5
6 digits.target
7 #print(digits.target)
8
9 digits.images[0]
10 print(digits.images[0])
```

Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.
IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[[0. 0. 5. ... 0. 0. 0.]
 [0. 0. 0. ... 10. 0. 0.]
 [0. 0. 0. ... 16. 9. 0.]
 ...
 [0. 0. 1. ... 6. 0. 0.]
 [0. 0. 2. ... 12. 0. 0.]
 [0. 0. 10. ... 12. 1. 0.]]

In [2]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[0 1 2 ... 8 9 8]

In [3]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[[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.]]

In [4]:

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris.

```
1 from sklearn import datasets, svm
2
3 digits = datasets.load_digits()
4
5 clf = svm.SVC(gamma=0.001, C=100.)
6 clf.fit(digits.data[:-1], digits.target[:-1])
7 clf.fit(digits.data[:-1], digits.target[:-1])
8 #print(clf.fit(digits.data[:-1], digits.target[:-1]))
9 print(clf.predict(digits.data[-1:]))
```

```
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/
Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
SVC(C=100.0, gamma=0.001)

In [2]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/
Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[8]

In [3]:
```

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris. Terdapat dua cara yaitu menggunakan pickle atau menggunakan joblib.

[illegible]

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris.

```

1 import numpy as np
2 from sklearn import random_projection, datasets
3 from sklearn.svm import SVC
4 from sklearn.datasets import load_iris
5 from sklearn.multiclass import OneVsRestClassifier
6 from sklearn.preprocessing import LabelBinarizer, MultiLabelBinarizer
7
8 iris = datasets.load_iris()
9
10 rng = np.random.RandomState(0)
11 X = rng.rand(10, 2000)
12 X = np.array(X, dtype='float32')
13 X.dtype
14 #print(X.dtype)
15
16 transformer = random_projection.GaussianRandomProjection()
17 X_new = transformer.fit_transform(X)
18 X_new.dtype
19 #print(X_new.dtype)
20
21 clf = SVC()
22 clf.fit(iris.data, iris.target)
23 #print(clf.fit(iris.data, iris.target))
24
25 list(clf.predict(iris.data[:3]))
26 #print(list(clf.predict(iris.data[:3])))
27
28 clf.fit(iris.data, iris.target_names[iris.target])
29 #print(clf.fit(iris.data, iris.target_names[iris.target]))
30
31 list(clf.predict(iris.data[:3]))
32 #print(list(clf.predict(iris.data[:3])))
33
34 X, y = load_iris(return_X_y=True)
35 clf = SVC()
36 clf.set_params(kernel='linear').fit(X, y)
37 #print(clf.set_params(kernel='linear').fit(X, y))
38
39 clf.predict(X[:5])
40 #print(clf.predict(X[:5]))
41
42 clf.set_params(kernel='rbf').fit(X, y)
43 #print(clf.set_params(kernel='rbf').fit(X, y))
44
45 clf.predict(X[:5])
46 #print(clf.predict(X[:5]))
47
48 X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
49 y = [0, 0, 1, 1, 2]
50
51 classifier = OneVsRestClassifier(estimator=SVC(random_state=0))
52 classifier.fit(X, y).predict(X)
53 #print(classifier.fit(X, y).predict(X))
54
55 y = LabelBinarizer().fit_transform(y)
56 classifier.fit(X, y).predict(X)
57 #print(classifier.fit(X, y).predict(X))
58
59 y = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
60 y = MultiLabelBinarizer().fit_transform(y)
61 classifier.fit(X, y).predict(X)
62 #print(classifier.fit(X, y).predict(X))

```

```

In [4]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[0, 0, 0]

In [5]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
SVC()

In [6]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
['setosa', 'setosa', 'setosa']

In [7]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
SVC(kernel='linear')

In [8]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[0 0 0 0]

In [9]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
SVC()

In [10]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[0 0 0 0]

In [11]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[0 0 1 1 2]

In [12]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[[1 0 0]
 [1 0 0]
 [0 1 0]
 [0 0 0]
 [0 0 0]]

In [13]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
[[1 1 0 0 0]
 [1 0 1 0 0]
 [0 1 0 1 0]
 [1 0 1 0 0]
 [1 0 1 0 0]]

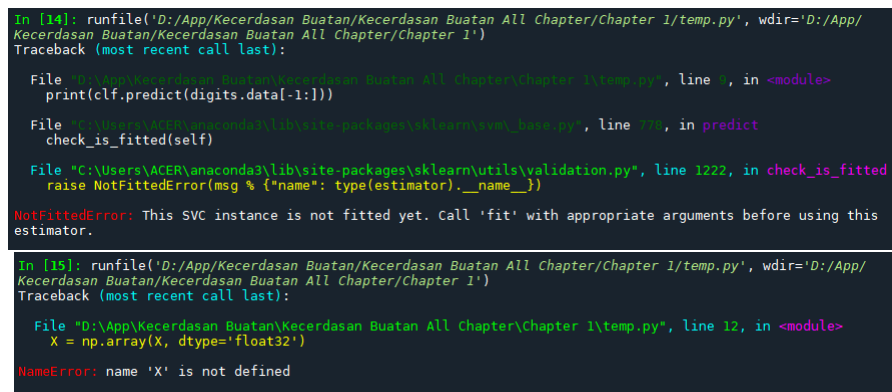
In [14]:

```

1.3 Penanganan Error

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

1. Screenshot error.

The image shows two screenshots of Jupyter Notebook error messages. The first screenshot shows a 'NotFittedError' where a SVC instance has not been fitted before attempting to predict. The second screenshot shows a 'NameError' where a variable 'X' has not been defined before being used in a numpy array creation.

```
In [14]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/
Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
Traceback (most recent call last):
  File "D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py", line 9, in <module>
    print(clf.predict(digits.data[-1:]))
  File "C:/Users/ACER/anaconda3/lib/site-packages/sklearn/base.py", line 178, in predict
    check_is_fitted(self)
  File "C:/Users/ACER/anaconda3/lib/site-packages/sklearn/utils/validation.py", line 1222, in check_is_fitted
    raise NotFittedError(msg % {"name": type(estimator).__name__})
NotFittedError: This SVC instance is not fitted yet. Call 'fit' with appropriate arguments before using this
estimator.

In [15]: runfile('D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py', wdir='D:/App/
Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1')
Traceback (most recent call last):
  File "D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 1/temp.py", line 12, in <module>
    X = np.array(X, dtype='float32')
NameError: name 'X' is not defined
```

2. Tuliskan kode error dan jenis errornya.

- (a) NotFittedError = This SVC instance is not fitted yet. Call 'fit' with appropriate arguments before using this estimator.
- (b) NameError = name 'X' is not defined

3. Solusi pemecahan masalah error tersebut.

- (a) NotFittedError = Solusinya yaitu memanggil parameter dengan method fit, sebelum menggunakan method predict.
- (b) NameError = Solusinya yaitu membuat variabel dengan nama X.

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.

Jawab:

Binary classification merupakan proses pengklasifikasian elemen-elemen himpunan berdasarkan aturan klasifikasi yang ouputnya dibagi menjadi 2 kelompok.

2. Jelaskan apa itu supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar sendiri.

Jawab:

Supervised Learning merupakan sebuah pendekatan yang ditentukan berdasarkan penggunaan training dataset yang berlabel/labeled dataset dan digunakan untuk melakukan klasifikasi data atau memprediksi hasil secara akurat.

Unsupervised Learning merupakan sebuah pendekatan yang ditentukan berdasarkan penggunaan training dataset yang tidak berlabel dan digunakan untuk menganalisa serta mengelompokkan kumpulan data yang tidak berlabel.

Clustering merupakan sebuah proses pengelompokan data ke dalam beberapa cluster yang dimana data-data di suatu cluster tersebut memiliki kemiripan.

3. Jelaskan apa itu evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar sendiri.

Jawab:

Evaluasi merupakan kegiatan untuk mengukur seberapa baik nilai performa dari suatu model dengan cara mengukur akurasinya.

Akurasi merupakan tingkat ketepatan yang diklasifikasikan dengan benar dari suatu model.

4. Jelaskan bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix buatan sendiri.

Jawab:

Cara membuat dan membaca confusion matrix:

- (a) Menentukan studi kasusnya.
- (b) Membuat ke dalam Decision Tree.
- (c) Menyiapkan data testing.
- (d) Mencari value dari variabel, misalnya yaitu a,b,c,d.
- (e) Mencari value dari recall, precision, accuracy, dan error state.

5. Jelaskan bagaimana K-fold cross validation bekerja dengan gambar ilustrasi contoh buatan sendiri.

Jawab:

Cara kerja dari K-fold cross validation:

- (a) Mengacak dataset secara random.

- (b) Membagi dataset tersebut kedalam k-group, yaitu sebagai test dataset dan sisanya sebagai training dataset.
 - (c) Memasang model pada training set dan evaluasi pada test dataset.
 - (d) Menyimpan skor evaluasi dan buang modelnya.
 - (e) Meringkas keterampilan model menggunakan sampel skor menggunakan model evaluasi.
6. Jelaskan apa itu decision tree dengan gambar ilustrasi contoh buatan sendiri.
Jawab:
Decision Tree merupakan suatu metode yang digunakan untuk membantu dalam pengambilan keputusan dengan menggunakan model pohon.
7. Jelaskan apa itu information gain dan entropi dengan gambar ilustrasi buatan sendiri.
Jawab:
Information gain merupakan kumpulan informasi yang didapatkan dari variable acak.
Entropi merupakan tingkat keacakan pada informasi yang diperloh dari information gain dan sedang diproses.

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).

1. # load dataset (student mat pakenya)

```
import pandas as pd
d = pd.read_csv('student-mat.csv', sep=';')
len(d)
```

Output:

```
In [2]: runcell('1 Load dataset', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/
chapter2.py')
395
```

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()
```

Output:

```
In [3]: runcell('2 Generate binary label (pass)', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/
Chapter 2/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]
```

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()

```

Output:

```

In [4]: runcell('3 use one-hot encoding on categorical columns', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan
All Chapter/Chapter 2/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 57 columns]

```

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']

```

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)))

```

Output:

```

In [5]: runcell('4 shuffle rows', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/
chapter2.py')
Passing: 166 out of 395 (42.03%)

```

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)
```

Output:

```
In [6]: runcell('5 fit a decision tree', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/
chapter2.py')
```

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),
```

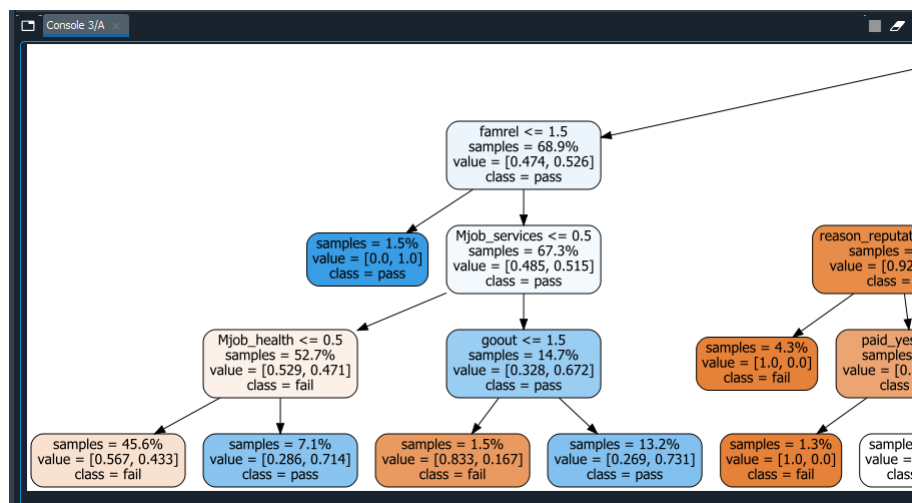
```
class_names=["fail", "pass"],
```

```
filled=True, rounded=True)
```

```
graph = graphviz.Source(dot_data)
```

```
graph
```

Output:



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)
```

Output:

```
In [8]: runcell('7 save tree', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/chapter2.py')
```

```
8. t.score(d_test_att, d_test_pass)
```

Output:

```
In [9]: runcell('8', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/chapter2.py')
0.7063291139240506
```

```
9. from sklearn.model_selection import cross_val_score
scores = cross_val_score(t, d_att, d_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))
```

Output:

```
In [10]: runcell('9', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/chapter2.py')
Accuracy: 0.54 (+/- 0.06)
```

```
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)
    )
```

Output:

```
In [11]: runcell('10', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/chapter2.py')
Max depth: 1, Accuracy: 0.58 (+/- 0.01)
Max depth: 2, Accuracy: 0.58 (+/- 0.15)
Max depth: 3, Accuracy: 0.55 (+/- 0.09)
Max depth: 4, Accuracy: 0.52 (+/- 0.09)
Max depth: 5, Accuracy: 0.54 (+/- 0.06)
Max depth: 6, Accuracy: 0.57 (+/- 0.08)
Max depth: 7, Accuracy: 0.53 (+/- 0.11)
Max depth: 8, Accuracy: 0.53 (+/- 0.05)
Max depth: 9, Accuracy: 0.54 (+/- 0.12)
Max depth: 10, Accuracy: 0.57 (+/- 0.05)
Max depth: 11, Accuracy: 0.56 (+/- 0.08)
Max depth: 12, Accuracy: 0.55 (+/- 0.07)
Max depth: 13, Accuracy: 0.56 (+/- 0.12)
Max depth: 14, Accuracy: 0.56 (+/- 0.07)
Max depth: 15, Accuracy: 0.54 (+/- 0.09)
Max depth: 16, Accuracy: 0.55 (+/- 0.09)
Max depth: 17, Accuracy: 0.54 (+/- 0.11)
Max depth: 18, Accuracy: 0.55 (+/- 0.07)
Max depth: 19, Accuracy: 0.56 (+/- 0.07)
```

```
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
```

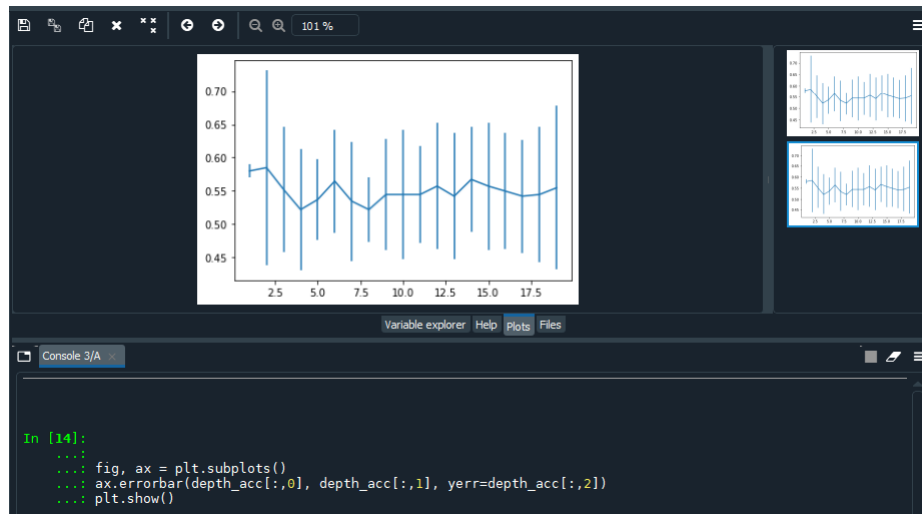
depth_acc

Output:

```
In [12]: runcell('11', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/chapter2.py')
[[1.00000000e+00 5.79746835e-01 1.01265823e-02]
[2.00000000e+00 5.84810127e-01 1.46223079e-01]
[3.00000000e+00 5.51898734e-01 9.41826594e-02]
[4.00000000e+00 5.21518987e-01 9.11392405e-02]
[5.00000000e+00 5.36708861e-01 6.11799796e-02]
[6.00000000e+00 5.64556962e-01 7.77837544e-02]
[7.00000000e+00 5.34177215e-01 8.97217476e-02]
[8.00000000e+00 5.21518987e-01 4.90904289e-02]
[9.00000000e+00 5.44303797e-01 8.31983632e-02]
[1.00000000e+01 5.44303797e-01 9.73943497e-02]
[1.10000000e+01 5.44303797e-01 7.33740595e-02]
[1.20000000e+01 5.56962025e-01 9.47255035e-02]
[1.30000000e+01 5.41772152e-01 9.52652543e-02]
[1.40000000e+01 5.67088608e-01 7.90911360e-02]
[1.50000000e+01 5.56962025e-01 9.60691947e-02]
[1.60000000e+01 5.49367089e-01 8.71121546e-02]
[1.70000000e+01 5.41772152e-01 8.53280990e-02]
[1.80000000e+01 5.44303797e-01 1.02523832e-01]
[1.90000000e+01 5.54430380e-01 1.23403115e-01]]
```

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

Output:



2.3 Penanganan Error

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

1. Screenshot error.

```

In [1]: runcell('1 Load dataset', 'D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/
chapter2.py')
Traceback (most recent call last):
  File "D:/App/Kecerdasan Buatan/Kecerdasan Buatan All Chapter/Chapter 2/chapter2.py", line 15, in <module>
    nanas = pd.read_csv('student-mat.csv', sep=';')
NameError: name 'pd' is not defined

```

2. Tuliskan kode error dan jenis errornya.

(a) NameError = name 'np' is not defined

3. Solusi pemecahan masalah error tersebut.

(a) NameError = Solusinya yaitu mengimport numpy dengan menginisialisasi-
saikannya sebagai np.