

Modul Praktikum Kecerdasan Buatan



Muhammad Syiarul Amrullah
1194026

Applied Bachelor of Informatics Engineering
Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering
Politeknik Pos Indonesia

Bandung 2019

‘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	3
1.3	Penanganan Error	7
2	Membangun Model Prediksi	8
2.1	Teori	8
2.2	scikit-learn	12
2.3	Penanganan Error	21
3	Prediksi dengan Random Forest	22
3.1	Teori	22
3.1.1	Random Forest	23
3.1.2	Confusion Matrix	25
3.1.3	Mencoba dengan metode Decission Tree dan SVM	26
3.1.4	Pengecekan Cross Validation	27
3.1.5	Pengamatan komponen informasi	27
3.2	Soal Teori	28
3.3	Praktek Program	28
3.4	Penanganan Error	29
3.5	Presentasi Tugas	30
4	Klasifikasi Teks	31
4.1	Teori	31
4.1.1	Vektorisasi data	31
4.1.2	Klasifikasi dengan Random Forest	33
4.1.3	Confusion Matrix	33
4.1.4	Pengecekan Cross Validation	33
4.2	Soal Teori	34

4.3	Praktek Program	34
4.4	Penanganan Error	35
4.5	Presentasi Tugas	36
5	Vektorisasi kata dan dokumen	37
5.1	Teori	37
5.2	Soal Teori	37
5.3	Praktek Program	38
5.4	Penanganan Error	39
5.5	Presentasi Tugas	39
6	MFCC dan Neural Network	41
6.1	Teori	41
6.2	Soal Teori	41
6.3	Praktek Program	42
6.4	Penanganan Error	43
6.5	Presentasi Tugas	44
7	CNN	45
7.1	Teori	45
7.2	Soal Teori	46
7.3	Praktek Program	47
7.4	Penanganan Error	49
7.5	Presentasi Tugas	50
8	Perkenalan Generative Adversarial Network	51
8.1	Teori	51
8.2	Soal Teori	51
8.3	Praktek Program	53
8.4	Penanganan Error	58
8.5	Presentasi Tugas	58
9	Conditional Generative Adversarial Network	60
9.1	Soal Teori	60
9.2	Praktek Program	61
9.3	Penanganan Error	62
10	Discussion	63

11 Discussion	64
12 Discussion	65
13 Discussion	66
14 Discussion	67
A Form Penilaian Jurnal	68
B FAQ	71
Bibliography	73

List of Figures

1.1	Instalasi Sklearn Melalui Anaconda Prompt	4
1.2	Variable Explorer	4
1.3	Source Code Loading an example dataset	4
1.4	Output Example Dataset	4
1.5	Variable Explorer dari output dataset	4
1.6	Source Code Learning and Predicting	5
1.7	Output Learning and Predicting	5
1.8	Source Code Model Persistence	5
1.9	Output Menggunakan Pickle	5
1.10	Output Menggunakan Joblib	5
1.11	Source Code Convention	6
1.12	Output Source Code	6
1.13	Error Could Not Install Packages due to an OSError Acces Is Denied	7
2.1	Binary Classification using dataset breast cancer	9
2.2	Contoh Hasil dari predict dari supervised learning menggunakan SVC	10
2.3	Source Code fitting dan predict menggunakan dataset iris pada model SVC	10
2.4	Binary Classification using dataset breast cancer	11
2.5	Binary Classification using dataset breast cancer	11
2.6	Binary Classification using dataset breast cancer	11
2.7	Source Code Task 1	12
2.8	Output Task 1	12
2.9	Source Code Task 2	13
2.10	Output Task 2	13
2.11	Source Code Task 3	14
2.12	Output Task 3	14
2.13	Source Code Task 4	15
2.14	Output Task 4	15

2.15 Source Code Task 5	16
2.16 Output Task 5	16
2.17 Source Code Task 6	16
2.18 Output Task 6	17
2.19 Source Code Task 7	17
2.20 Output Task 7	17
2.21 Source Code Task 8	18
2.22 Output Task 8	18
2.23 Source Code Task 9	18
2.24 Output Task 9	19
2.25 Source Code Task 10	19
2.26 Output Task 10	19
2.27 Source Code Task 11	20
2.28 Output Task 11	20
2.29 Source Code Task 12	21
2.30 Output Task 12	21
A.1 Form nilai bagian 1.	69
A.2 form nilai bagian 2.	70

Chapter 1

Mengenai Kecerdasan Buatan dan Scikit-Learn

Buku umum teori lengkap yang digunakan memiliki judul *Artificial intelligence: a modern approach*[4]. Untuk pratikum sebelum UTS menggunakan buku *Python Artificial Intelligence Projects for Beginners*[2]. Buku pelengkap penunjang penggunaan python menggunakan buku *Python code for Artificial Intelligence: Foundations of Computational Agents*[3]. 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. Resume Definisi, Sejarah dan perkembangan Kecerdasan Buatan

- (a) Definisi Kecerdasan Buatan

Kecerdasan Buatan merupakan teknologi yang memiliki kecerdasan layaknya manusia namun kecerdasan yang dimiliki oleh AI merupakan hasil dari permodelan oleh manusia yang dibuat agar bisa berpikir seperti manusia.

(b) Sejarah Kecerdasan Buatan

Kecerdasan Buatan (Artificial Intelligence) bermula dari adanya kemunculan komputer pada tahun 1940, dengan salah satu penemuan model matematis bernama perceptron dari neuron didalam otak yang diusulkan oleh McCulloch dan Pitts di tahun 1943, dengan menunjukkan bagaimana cara neuron menjadi aktif seperti saklar on/off dan neuron tersebut mampu memahami dan memberikan aksi terhadap dari waktu dari input yang diberikan. di beberapa tahun kemudian ada beberapa penemuan dari beberapa ahli lainnya, seperti;

- pada tahun 1950 - Paper berjudul Computing Machinery and Intelligence - dirancang oleh Alan Turing
- pada tahun 1955 - The Logic Theorist - dirancang oleh Newell dan Simon
- pada tahun 1956 s/d 1980 - pengembangan pemahaman mengenai AI dengan diadakannya konferensi dan diskusi dengan para ahli komputer

(c) Perkembangan Kecerdasan Buatan

Perkembangan Kecerdasan Buatan, berkembang semenjak tahun 1956 - 1970, dengan dilakukan pemahaman dengan berdiskusi dengan para ahli komputer, banyak universitas juga yang berpartisipasi di MIT, seperti (universitas di Edinburgh, Standford dan Carnegie Mellon) yang mana komputer pemrograman mulai membuktikan masalah aljabar, teorema geometris yang menggunakan pemahaman sintaks dan tata bahasa inggris.

2. Resume mengenai definisi supervised learning, klasifikasi, regresi dan unsupervised learning Data set, training set dan testing set.

(a) Definisi Supervised Learning, klasifikasi, regresi

- Definisi Supervised Learning
merupakan *Machine Learning* model yang mempelajari data dengan label atau target dimana evaluasi model tersebut akan berdasarkan target. contoh: pengenalan jenis bunga berdasarkan ukuran kelopak dan warna kelopak.
- Definisi Klasifikasi
merupakan *Machine Learning* model yang mempelajari data dengan label atau target dimana evaluasi model tersebut akan berdasarkan

target. label merupakan pengenalan dari sebuah data. contoh: pengenalan jenis bunga berdasarkan ukuran kelopak dan warna kelopak.

- Definisi Regresi

metode pada supervised learning yang mengembalikan target numerik untuk setiap sampel sedangkan klasifikasi adalah metode supervised learning yang bekerja dengan cara memberikan label pada setiap sampel dengan memilih dua atau lebih kelas atau kelompok yang berbeda.

(b) Definisi Unsupervised Learning, Dataset, Training Set dan Testing Set

- Definisi Unsupervised Learning

Unsupervised learning adalah salah satu tipe algoritma machine learning yang digunakan untuk menarik kesimpulan dari dataset

- Definisi Dataset

Dataset adalah istilah informal yang merujuk pada kumpulan . Secara umum, dataset berisi lebih dari satu variabel dan menyangkut suatu topik tertentu yang akan digunakan untuk klasifikasi dengan metode data mining.

- Definisi Training Set

Training Set merupakan bagian dataset yang disiapkan untuk melatih model dalam membuat prediksi atau menjalankan fungsi dari sebuah algoritma ML. Kita memberikan petunjuk melalui algoritma agar mesin yang kita latih bisa mencari korelasinya sendiri atau belajar pola dari data yang diberikan.

- Definisi Testing Set

Testing Set merupakan bagian dataset yang disiapkan untuk melakukan tes pada model yang telah dilatih untuk melihat keakuratannya, atau dengan kata lain melihat performanya.

1.2 Instalasi

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

Link Livestreaming: Klik Disini

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

Figure 1.1: Instalasi Sklearn Melalui Anaconda Prompt

```
(base) C:\Users\yulrpsip install -U scikit-learn -user
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda\lib\site-packages)
Requirement already satisfied: scikit-learn in c:\programdata\anaconda\lib\site-packages (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: joblib>=0.12 in c:\programdata\anaconda\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: numpy>=1.14.6 in c:\programdata\anaconda\lib\site-packages (from scikit-learn) (1.20.3)
Requirement already satisfied: scipy>=1.0 in c:\programdata\anaconda\lib\site-packages (from scikit-learn) (1.7.1)
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda\lib\site-packages)
```

Figure 1.2: Variable Explorer

Name	Type	Size	Value
a	list	5	[(1, 2), (2, 4), (4, 5), (3, 2), (3, 1)]
b	list	5	[0, 0, 1, 1, 2]
B	Array of int32	(5, 3)	[[1 0 0] [1 0 0]]
c	list	5	[[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
C	Array of int32	(5, 5)	[[1 1 0 0 0] [1 0 1 0 0]]
classif	multiclass.OneVsRestClassifier	1	OneVsRestClassifier object of sklearn.multiclass module
clf	svm._classes.SVC	1	SVC object of sklearn.svm._classes module
clf2	svm._classes.SVC	1	SVC object of sklearn.svm._classes module
clf3	svm._classes.SVC	1	SVC object of sklearn.svm._classes module
clf4	svm._classes.SVC	1	SVC object of sklearn.svm._classes module
clf_iris	svm._classes.SVC	1	SVC object of sklearn.svm._classes module
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[hari ke 1](10)

```
from sklearn import datasets
from sklearn import svm

import matplotlib
import numpy as np
from sklearn import random_projection
from sklearn import dump, load
from sklearn.multiclass import OneVsRestClassifier
from sklearn.preprocessing import LabelBinarizer
from sklearn.preprocessing import MultiLabelBinarizer

iris = datasets.load_iris()
digits = datasets.load_digits()
print(digits.data)
print(digits.target)
print(iris.target_names)
```

Figure 1.3: Source Code Loading an example dataset

```
In [2]: runfile('E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1/source code.py', wdir='E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1')
[[ 0. 0. 5. ... 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.]]
Output digits.data
Output digits.target
Output iris.target_names
[0 1 2 ... 8 9 8]
['setosa' 'versicolour' 'virginica']
```

Figure 1.4: Output Example Dataset

Key	Type	Size	Value
data	Array of float64	(1797, 64)	[[0. 0. 5. ... 0. 0. 0.] [0. 0. 0. ... 10. 0. 0.] ... [0. 0. 1. ... 6. 0. 0.] [0. 0. 2. ... 12. 0. 0.] [0. 0. 10. ... 12. 1. 0.]]
DESCR	str	2028	..._digits_dataset
feature_names	list	64	['pixel_0_0', 'pixel_0_1', 'pixel_0_2', 'pixel_0_3', 'pixel_0_4', 'pin ...]
frame	NoneType	1	NoneType object
images	Array of float64	(1797, 64)	[[[0. 0. 5. ... 1. 0. 0.] [0. 0. 13. ... 13. 0. 0.] ... [0. 0. 1. ... 6. 0. 0.] [0. 0. 2. ... 12. 0. 0.] [0. 0. 10. ... 12. 1. 0.]]]
target	Array of int32	(1797,)	[0 1 2 ... 8 9 8]
target_names	Array of int32	(10,)	[0 1 2 3 4 5 6 7 8 9]

Key	Type	Size	Value
data	Array of float64	(150, 4)	[[5.1 3.5 1.4 0.2] [4.9 3.1 1.4 0.2]]
data_module	str	21	sklearn.datasets.data
DESCR	str	2782	..._iris_dataset
feature_names	list	4	['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal ...]
filename	str	8	iris.csv
frame	NoneType	1	NoneType object
target	Array of int32	(150,)	[0 0 0 ... 2 2 2]
target_names	Array of str320	(3,)	ndarray object of numpy module

Figure 1.5: Variable Explorer dari output dataset

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

```
clf = svm.SVC(gamma=0.001, C=100.)
print(clf.fit(digits.data[:-1], digits.target[:-1]))
print(clf.predict(digits.data[-1:]))
```

Figure 1.6: Source Code Learning and Predicting

```
In [4]: runfile('E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1/source code.py', wdir='E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1')
SVC(C=100.0, gamma=0.001)
[8]
```

Figure 1.7: Output Learning and Predicting

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

```
clf2 = svm.SVC()
X, y = datasets.load_iris(return_X_y=True)
clf2.fit(X, y)
svm.SVC()

s = pickle.dumps(clf2)
clf3 = pickle.loads(s)
print(clf3.predict(X[0:1]))

dump(clf2, 'filename.joblib')
clf4 = load('filename.joblib')
```

Figure 1.8: Source Code Model Persistence

```
In [5]: runfile('E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1/source code.py', wdir='E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1')
[0]
```

Figure 1.9: Output Menggunakan Pickle

```
In [7]: runfile('E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1/source code.py', wdir='E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1')
[0]
```

Figure 1.10: Output Menggunakan Joblib

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

```
rng = np.random.RandomState(0)
u = rng.rand(10, 2000)
U = np.array(u, dtype='float32')
print(U.dtype)

transformer = random_projection.GaussianRandomProjection()
U_new = transformer.fit_transform(U)
print(U_new.dtype)
iris_clf = svm.SVC()
iris_clf.fit(iris.data, iris.target)
print(list(iris_clf.predict(iris.data[:3])))
iris_clf.fit(iris.data, iris.target_names[iris.target])
print(list(iris_clf.predict(iris.data[:3])))

clf_iris = svm.SVC()
clf_iris.set_params(kernel='linear').fit(X, y)
print(clf_iris.predict(X[:5]))
clf_iris.set_params(kernel='rbf').fit(X, y)
print(clf_iris.predict(X[:5]))

a = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
b = [0, 0, 1, 1, 2]

classif = OneVsRestClassifier(estimator=svm.SVC(random_state=0))
print(classif.fit(a, b).predict(a))

B = LabelBinarizer().fit_transform(b)
print(classif.fit(a, B).predict(a))

c = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
C = MultiLabelBinarizer().fit_transform(c)
print(classif.fit(a, C).predict(a))
```

Figure 1.11: Source Code Convention

```
In [0]: runfile('E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1/source code.py', wdir='E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 1')
float32
float64
[0, 0, 0]
['setosa', 'setosa', 'setosa']
[0 0 0 0]
[0 0 0 0]
[0 0 1 1 2]
[[1 0 0]
 [1 0 0]
 [0 1 0]
 [0 0 0]
 [0 0 0]]
[[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]]
```

Figure 1.12: Output Source Code

1.3 Penanganan Error

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

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

```
(base) C:\Users\syiar>pip install -U scikit-learn
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda3\lib\site-packages)
Requirement already satisfied: scikit-learn in c:\programdata\anaconda3\lib\site-packages (0.24.2)
Collecting scikit-learn
  Using cached scikit_learn-1.0.2-cp39-cp39-win_amd64.whl (7.2 MB)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: scipy>=1.1.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.7.1)
Requirement already satisfied: joblib>=0.11 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: numpy>=1.14.6 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.20.3)
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda3\lib\site-packages)
Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda3\lib\site-packages)
    Found existing installation: scikit-learn 0.24.2
    Uninstalling scikit-learn-0.24.2:
      Successfully uninstalled scikit-learn-0.24.2
  ERROR: Could not install packages due to an OSError: [WinError 5] Access is denied: 'C:\\ProgramData\\Anaconda3\\Lib\\site-packages\\-klearn\\utils\\murmurhash.cp39-win_amd64.pyd'
  Consider using the '--user' option or check the permissions.
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\programdata\anaconda3\lib\site-packages)
```

Figure 1.13: Error Could Not Install Packages due to an OSError Acces Is Denied

2. Tuliskan Kode error dan Jenis Errornya [hari ke 2](10)
pip install -U scikit-learn (jenis error Could Not Install Packages due to an OSError Acces Is Denied)
3. Solusi pemecahan masalah error tersebut[hari ke 2](10)
tambahkan --user pada command install pip ” pip install -U scikit-learn --user ”

Chapter 2

Membangun Model Prediksi

Untuk pratikum saati ini menggunakan buku *Python Artificial Intelligence Projects for Beginners*[2]. 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 bentuk klasifikasi untuk proses memprediksi variable kategori dimana output dibatasi untuk kedua kelas.

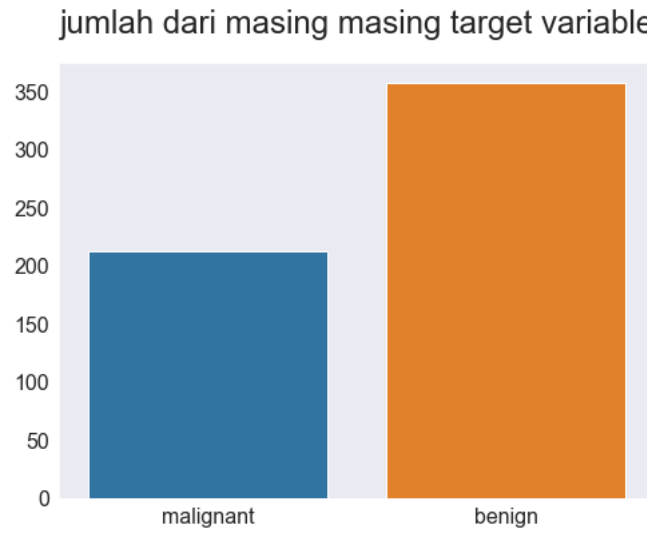


Figure 2.1: Binary Classification using dataset breast cancer

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

- supervised learning merupakan Machine Learning model yang mempelajari data dengan label atau target dimana evaluasi model tersebut akan berdasarkan target. contoh: pengenalan jenis bunga berdasarkan ukuran kelopak dan warna kelopak. seperti pada Figure 2.2 merupakan hasil predict dan ciri dari sample dataset iris yang didapatkan dari algoritma SVC untuk memprediksi target dari suatu objek berdasarkan beberapa ciri seperti sepal length, sepal width, petal width dan sepal length, akan diprediksi menjadi salah satu dari beberapa target name seperti setosa, versicolor dan virginica.

Index	sepal length	sepal width	petal length	petal width	result
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa

Figure 2.2: Contoh Hasil dari predict dari supervised learning menggunakan SVC

```
dataset_iris = load_iris()
iris_clf = svm.SVC()
iris_clf.fit(dataset_iris.data, dataset_iris.target)
iris_clf.fit(dataset_iris.data, dataset_iris.target_names[dataset_iris.target])

hasil_predict = list(iris_clf.predict(dataset_iris.data[:3]))

data_result_iris = {'sepal length' : dataset_iris.data[:3,0],
                    'sepal width': dataset_iris.data[:3,1],
                    'petal length': dataset_iris.data[:3,2],
                    'petal width': dataset_iris.data[:3,3],
                    'result': hasil_predict}
df_iris = pd.DataFrame(data_result_iris)
```

Figure 2.3: Source Code fitting dan predict menggunakan dataset iris pada model SVC

- Unsupervised learning adalah salah satu tipe algoritma machine learning yang digunakan untuk menarik kesimpulan dari dataset. Metode ini hanya akan mempelajari suatu data berdasarkan kedekatannya saja atau yang biasa disebut dengan clustering. Metode unsupervised learning yang paling umum adalah analisis cluster, yang digunakan pada analisa data untuk mencari pola-pola tersembunyi atau pengelompokan dalam data.

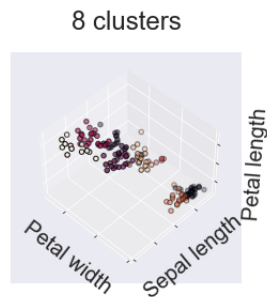


Figure 2.4: Binary Classification using dataset breast cancer

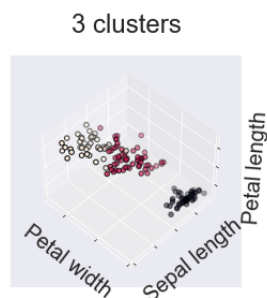


Figure 2.5: Binary Classification using dataset breast cancer

3 clusters, bad initialization

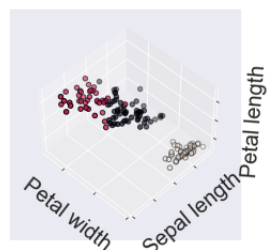


Figure 2.6: Binary Classification using dataset breast cancer

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

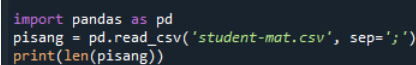
4. Jelaskan bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix buatan sendiri.
5. Jelaskan bagaimana K-fold cross validation bekerja dengan gambar ilustrasi contoh buatan sendiri.
6. Jelaskan apa itu decision tree dengan gambar ilustrasi contoh buatan sendiri.
7. Jelaskan apa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

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).

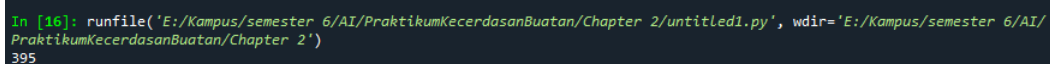
1. # load dataset (student mat pakenya)


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



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

Figure 2.7: Source Code Task 1



```
In [16]: runfile('E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2/untitled1.py', wdir='E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2')
395
```

Figure 2.8: Output Task 1

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()
```

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

Figure 2.9: Source Code Task 2

```
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]
```

Figure 2.10: Output Task 2

```
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()
```

```
# use one-hot encoding on categorical columns
pisang = pd.get_dummies(pisang, columns=['sex', 'school', 'address',
                                         'famsize',
                                         'Pstatus', 'Mjob', 'Fjob',
                                         'reason', 'guardian', 'schoolsup',
                                         'famsup', 'paid', 'activities',
                                         'nursery', 'higher', 'internet',
                                         'romantic'])
# print(pisang.head())
```

Figure 2.11: Source Code Task 3

```
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]
```

Figure 2.12: Output Task 3

```

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

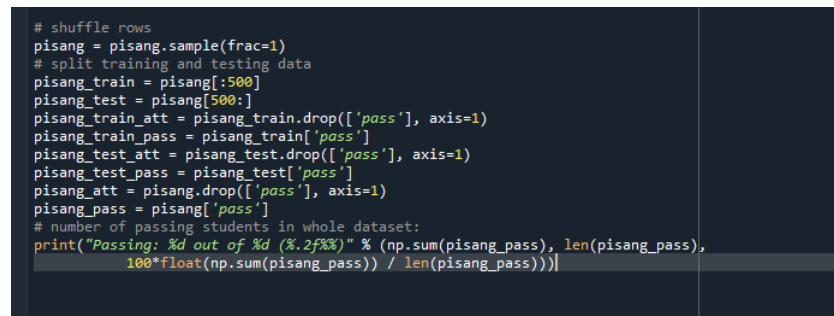
d_train_att = d_train.drop(['pass'], axis=1)
d_train_pass = d_train['pass']

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)))

```

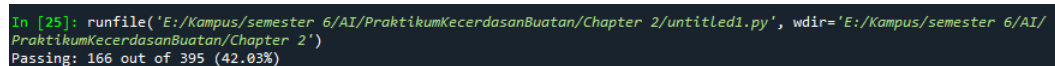


```

# shuffle rows
pisang = pisang.sample(frac=1)
# split training and testing data
pisang_train = pisang[:500]
pisang_test = pisang[500:]
pisang_train_att = pisang_train.drop(['pass'], axis=1)
pisang_train_pass = pisang_train['pass']
pisang_test_att = pisang_test.drop(['pass'], axis=1)
pisang_test_pass = pisang_test['pass']
pisang_att = pisang.drop(['pass'], axis=1)
pisang_pass = pisang['pass']
# number of passing students in whole dataset:
print("Passing: %d out of %d (%.2f%%)" % (np.sum(pisang_pass), len(pisang_pass),
    100*float(np.sum(pisang_pass)) / len(pisang_pass)))

```

Figure 2.13: Source Code Task 4



```

In [25]: runfile('E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2/untitled1.py', wdir='E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2')
Passing: 166 out of 395 (42.03%)

```

Figure 2.14: Output Task 4

```

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)

```



```
# fit a decision tree
jeruk = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
jeruk = jeruk.fit(pisang_train_att, pisang_train_pass)
print(jeruk)
```

Figure 2.15: Source Code Task 5

```
In [37]: runfile('E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2/untitled1.py', wdir='E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2')
DecisionTreeClassifier(criterion='entropy', max_depth=5)
```

Figure 2.16: Output Task 5

```
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
```

```
# visualize tree
import graphviz
semangka = tree.export_graphviz(jeruk, out_file=None, label="all",
                                impurity=False, proportion=True,
                                feature_names=list(pisang_train_att),
                                class_names=["fail", "pass"],
                                filled=True, rounded=True)
apel = graphviz.Source(semangka)
print(apel)
```

Figure 2.17: Source Code Task 6

```

In [38]: runfile('E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2/untitled1.py', wdir='E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2')
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_no <= 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="Mjob_services <= 0.5\nsamples = 10.1%\nvalue = [0.825, 0.175]\nnclass = fail", fillcolor="#eb9c63"];
  1 -> 2;
  3 [label="Fedu <= 2.5\nsamples = 6.6%\nvalue = [0.923, 0.077]\nnclass = fail", fillcolor="#e78c49"];
  2 -> 3;
  4 [label="famsize_LE3 <= 0.5\nsamples = 2.3%\nvalue = [0.778, 0.222]\nnclass = fail", fillcolor="#eca572"];
  3 -> 4;
  5 [label="samples = 1.0%\nvalue = [0.5, 0.5]\nnclass = fail", fillcolor="ffffff"];
  4 -> 5;
  6 [label="samples = 1.3%\nvalue = [1.0, 0.0]\nnclass = fail", fillcolor="#e58139"];
  4 -> 6;
  7 [label="samples = 4.3%\nvalue = [1.0, 0.0]\nnclass = fail", fillcolor="#e58139"];
  3 -> 7;
  8 [label="health <= 2.5\nsamples = 3.5%\nvalue = [0.643, 0.357]\nnclass = fail", fillcolor="#f3c7a7"];
  2 -> 8;
}

```

Figure 2.18: Output Task 6

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)

```

```

# save tree
tree.export_graphviz(jeruk, out_file="student-performance.dot",
  label="all", impurity=False,
  proportion=True,
  feature_names=list(pisang_train_att),
  class_names=["fail", "pass"],
  filled=True, rounded=True)

```

Figure 2.19: Source Code Task 7

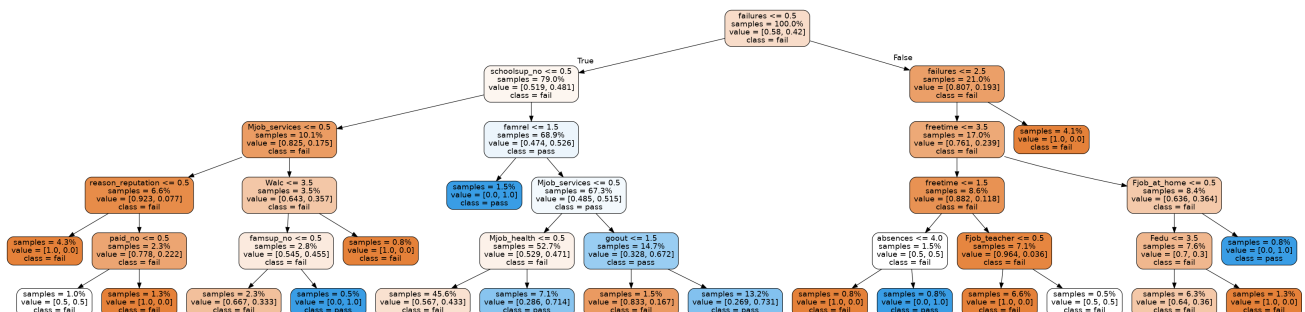


Figure 2.20: Output Task 7

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

```
###  
print(jeruk.score(pisang_test_att, pisang_test_pass))  
###
```

Figure 2.21: Source Code Task 8

```
In [13]: runcell(8, 'E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2/sklearnPraktikum.py')  
0.7063291139240506
```

Figure 2.22: Output Task 8

```
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))
```

```
###  
durian = cross_val_score(jeruk, pisang_att, pisang_pass, cv=5)  
# show average score and +/- two standard deviations away  
#(covering 95% of scores)  
print("Accuracy: %0.2f (+/- %0.2f)" % (durian.mean(), durian.std() * 2))
```

Figure 2.23: Source Code Task 9

```
In [14]: runcell(9, 'E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2/sklearnPraktikum.py')
Accuracy: 0.57 (+/- 0.09)
```

Figure 2.24: Output Task 9

```
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)
    )
```

```
#####
for lemon in range(1,20):
    jeruk_2 = tree.DecisionTreeClassifier(criterion="entropy", max_depth = lemon)
    durian_2 = cross_val_score(jeruk_2, pisang_att, pisang_pass, cv=5)
    print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" % (lemon, durian_2.mean(), durian_2.std()*2))
```

Figure 2.25: Source Code Task 10

```
In [15]: runcell(10, 'E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2/sklearnPraktikum.py')
Max depth: 1, Accuracy: 0.58 (+/- 0.01)
Max depth: 2, Accuracy: 0.60 (+/- 0.09)
Max depth: 3, Accuracy: 0.56 (+/- 0.09)
Max depth: 4, Accuracy: 0.57 (+/- 0.06)
Max depth: 5, Accuracy: 0.56 (+/- 0.09)
Max depth: 6, Accuracy: 0.57 (+/- 0.10)
Max depth: 7, Accuracy: 0.57 (+/- 0.08)
Max depth: 8, Accuracy: 0.58 (+/- 0.09)
Max depth: 9, Accuracy: 0.58 (+/- 0.08)
Max depth: 10, Accuracy: 0.60 (+/- 0.04)
Max depth: 11, Accuracy: 0.61 (+/- 0.08)
Max depth: 12, Accuracy: 0.59 (+/- 0.07)
Max depth: 13, Accuracy: 0.62 (+/- 0.08)
Max depth: 14, Accuracy: 0.59 (+/- 0.07)
Max depth: 15, Accuracy: 0.59 (+/- 0.07)
Max depth: 16, Accuracy: 0.58 (+/- 0.08)
Max depth: 17, Accuracy: 0.58 (+/- 0.10)
Max depth: 18, Accuracy: 0.58 (+/- 0.05)
Max depth: 19, Accuracy: 0.61 (+/- 0.08)
```

Figure 2.26: Output Task 10

```
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

```
###
kelengkeng = np.empty((19,3), float)
melon = 0
for lemon in range(1,20):
    jeruk_3 = tree.DecisionTreeClassifier(criterion="entropy", max_depth=lemon)
    durian_3 = cross_val_score(jeruk_3, pisang_att, pisang_pass, cv=5)
    kelengkeng[melon, 0]=lemon
    kelengkeng[melon, 1]=durian_3.mean()
    kelengkeng[melon, 2]=durian_3.std() * 2
    melon += 1
print(kelengkeng)
```

Figure 2.27: Source Code Task 11

```
In [16]: runcell(11, 'E:/Kampus/semester 6/AI/PraktikumKecerdasanBuatan/Chapter 2/sklearnPraktikum.py')
[[1.00000000e+00 5.79746835e-01 1.01265823e-02]
 [2.00000000e+00 5.97468354e-01 8.97217476e-02]
 [3.00000000e+00 5.67088608e-01 9.25350222e-02]
 [4.00000000e+00 5.74683544e-01 5.68353021e-02]
 [5.00000000e+00 5.69620253e-01 8.31983632e-02]
 [6.00000000e+00 5.82278481e-01 8.31983632e-02]
 [7.00000000e+00 5.84810127e-01 6.07594937e-02]
 [8.00000000e+00 5.77215190e-01 1.00758221e-01]
 [9.00000000e+00 5.87341772e-01 7.08860759e-02]
 [1.00000000e+01 5.89873418e-01 3.78902014e-02]
 [1.10000000e+01 6.07594937e-01 8.31983632e-02]
 [1.20000000e+01 5.79746835e-01 1.01770892e-01]
 [1.30000000e+01 5.89873418e-01 1.09301434e-01]
 [1.40000000e+01 6.15189873e-01 5.21297729e-02]
 [1.50000000e+01 5.94936709e-01 1.02523832e-01]
 [1.60000000e+01 6.02531646e-01 9.28116596e-02]
 [1.70000000e+01 6.15189873e-01 6.11799796e-02]
 [1.80000000e+01 6.00000000e-01 9.41826594e-02]
 [1.90000000e+01 5.92405063e-01 1.54741335e-01]]
```

Figure 2.28: Output Task 11

```

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

```

```

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

```

Figure 2.29: Source Code Task 12

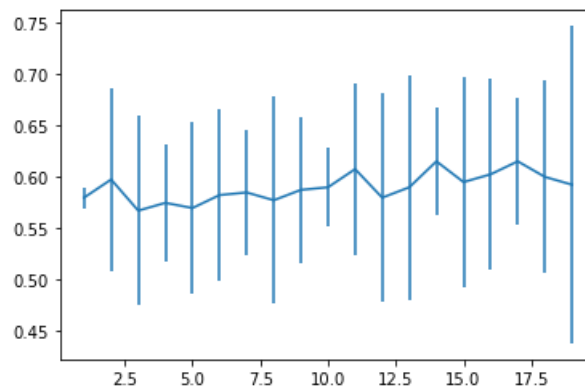


Figure 2.30: Output Task 12

2.3 Penanganan Error

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

1. skreensut 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*[2]. 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,
8         n_estimators=n_estimators)
9         scores = cross_val_score(clf, df_train_att, df_train_label, cv
10         =5)
11         rf_params[i,0] = max_features
12         rf_params[i,1] = n_estimators
13         rf_params[i,2] = scores.mean()
14         rf_params[i,3] = scores.std() * 2
15         i += 1
16         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)
2. buat aplikasi sederhana menggunakan numpy dan jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas)
3. buat aplikasi sederhana menggunakan matplotlib dan jelaskan arti dari setiap baris kode(harus beda dengan teman sekelas)
4. jalankan program klasifikasi Random Forest pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap keluaran yang didapatkan.
5. jalankan program confusion matrix pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap keluaran yang didapatkan.
6. jalankan program klasifikasi SVM dan Decision Tree pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap keluaran yang didapatkan.
7. jalankan program cross validation pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap keluaran yang didapatkan.
8. jalankan program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap keluaran yang didapatkan.

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 eror 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*[2]. 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 menampungnya 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 listing 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 perintah 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)
2. dari dataframe tersebut dipecah menjadi dua dataframe yaitu 450 row pertama dan 50 row sisanya(harus beda dengan teman sekelas)

3. praktekkan vektorisasi dan klasifikasi dari data (NPM mod 4, jika 0 maka katty perry, 1 LMFAO, 2 Eminem, 3 Shakira) dengan Decision Tree. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
4. 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.
5. Cobalah klasifikasikan dari data vektorisasi yang di tentukan di nomor sebelumnya dengan klasifikasi Decision Tree. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
6. Plotlah confusion matrix dari praktek modul ini menggunakan matplotlib. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
7. jalankan program cross validaiton pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
8. Buatlah program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

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

4.5 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 kematerian 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 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.

Chapter 5

Vektorisasi kata dan dokumen

Untuk pratikum saati ini menggunakan buku *Python Artificial Intelligence Projects for Beginners*[2]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Kode program ada di <https://github.com/awangga/Python-Artificial-Intelligence-Projects-for-Beginners> . Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti konsep dasar vektorisasi pada kata dan dokumen
2. Mengerti teknik machine learning untuk similaritas kata dan dokumen
3. Memahami score dari berbagai teknik klasifikasi

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. Tidak menyertakan pdf kompilasi diskon 50% nilainya.

5.1 Teori

Teori diambil dari buku referensi mengenai apa vektorisasi dari kata dan dokumen. Dan bagaimana konsep vektorisasi dan similaritas. Kode dan praktek bisa dilihat di youtube dosen.

5.2 Soal Teori

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

1. Jelaskan kenapa kata-kata harus di lakukan vektorisasi. dilengkapi dengan ilustrasi atau gambar.
2. Jelaskan mengapa dimensi dari vektor dataset google bisa sampai 300.dilengkapi dengan ilustrasi atau gambar.
3. Jelaskan konsep vektorisasi untuk kata.dilengkapi dengan ilustrasi atau gambar.
4. Jelaskan konsep vektorisasi untuk dokumen.dilengkapi dengan ilustrasi atau gambar.
5. Jelaskan apa mean dan standar deviasi,dilengkapi dengan ilustrasi atau gambar.
6. Jelaskan apa itu skip-gram,dilengkapi dengan ilustrasi atau gambar.

5.3 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. Cobalah dataset google, dan jelaskan vektor dari kata love, faith, fall, sick, clear, shine, bag, car, wash, motor, cycle dan cobalah untuk melakukan perbandingan similirati dari masing-masing kata tersebut. jelaskan arti dari outputan similaritas dan setiap baris kode yang dibuat(harus beda dengan teman sekelas). (Nilai 5 untuk setiap perbandingan, disini ada 5 perbandingan similaritas)
2. jelaskan dengan kata dan ilustrasi fungsi dari extract_words dan PermuteSentences (harus beda dengan teman sekelas)
3. Jelaskan fungsi dari librari gensim TaggedDocument dan Doc2Vec disertai praktek pemakaiannya. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
4. Jelaskan dengan kata dan praktek cara menambahkan data training dari file yang dimasukkan kepada variabel dalam rangka melatih model doc2vac. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

5. Jelaskan dengan kata dan praktek kenapa harus dilakukan pengocokan dan pembersihan data. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
6. Jelaskan dengan kata dan praktek kenapa model harus di save dan kenapa temporari training harus dihapus. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
7. Jalankan dengan kta dan praktek maksud dari `infer_code`. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
8. Jelaskan dengan praktek dan kata maksud dari `cosine_similarity`. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
9. Jelaskan dengan praktek score dari cross validation masing-masing metode. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

5.4 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. Screenshot error
2. Tuliskan kode error dan jenis errornya
3. Solusi pemecahan masalah error tersebut

5.5 Presentasi Tugas

Pada pertemuan ini, diadakan dua 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 ke-materian dengan nilai presentasi yang terpisah masing-masing 100. Jadi ada dua komponen penilaian pada pertemuan ini yaitu :

1. tugas minggu hari ini dan besok (maks 100). pada chapter ini

2. presentasi tugas kode word2vec dan doc2vec (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(10 praktek dan 10 teori) pertanyaan tentang pema-
hamannya menggunakan python untuk kecerdasan buatan. jika presenter tidak bisa menjawab satu pertanyaan asisten maka nilai nol. Jika semua pertanyaan bisa di-
jawab maka nilai 100. Presentasi bisa diulang apabila gagal, sampai bisa mendap-
atkan nilai 100 dalam waktu satu minggu kedepan.

Chapter 6

MFCC dan Neural Network

Untuk pratikum saat ini menggunakan buku *Python Artificial Intelligence Projects for Beginners*[2]. Dengan praktek menggunakan python 3 dan editor spyder dan library python keras dan librosa. Kode program ada di <https://github.com/awangga/Python-Artificial-Intelligence-Projects-for-Beginners> . Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti konsep dasar MFCC untuk vektorisasi suara
2. Mengerti teknik Neural Network dari hasil MFCC
3. Memahami konsep pembobotan dan fungsi aktivasi

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. Tidak menyertakan **pdf** kompilasi **diskon 50%** nilainya.

6.1 Teori

Teori diambil dari buku referensi mengenai Neural Network dan MFCC dari dataset suara. Materi dan praktek bisa dilihat di youtube dosen.

6.2 Soal Teori

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

1. Jelaskan kenapa file suara harus di lakukan MFCC. dilengkapi dengan ilustrasi atau gambar.
2. Jelaskan konsep dasar neural network.dilengkapi dengan ilustrasi atau gambar.
3. Jelaskan konsep pembobotan dalam neural network.dilengkapi dengan ilustrasi atau gambar.
4. Jelaskan konsep fungsi aktivasi dalam neural network. dilengkapi dengan ilustrasi atau gambar.
5. Jelaskan cara membaca hasil plot dari MFCC,dilengkapi dengan ilustrasi atau gambar.
6. Jelaskan apa itu one-hot encoding,dilengkapi dengan ilustrasi kode dan atau gambar.
7. Jelaskan apa fungsi dari np.unique dan to_categorical dalam kode program,dilengkapi dengan ilustrasi atau gambar.
8. Jelaskan apa fungsi dari Sequential dalam kode program,dilengkapi dengan ilustrasi atau gambar.

6.3 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. Jelaskan isi dari data GTZAN Genre Collection dan data dari freesound. Buat kode program untuk meload data tersebut untuk digunakan pada MFCC. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas).
2. Jelaskan perbaris kode program dengan kata-kata dan dilengkapi ilustrasi gambar fungsi dari `display_mfcc()` .
3. Jelaskan perbaris kode program dengan kata-kata dan dilengkapi ilustrasi gambar fungsi dari `extract_features_song()`. Jelaskan juga mengapa yang diambil 25.000 baris pertama?

4. Jelaskan perbaris kode program dengan kata-kata dan dilengkapi ilustrasi gambar fungsi dari `generate_features_and_labels()`.
5. Jelaskan dengan kata dan praktek kenapa penggunaan fungsi `generate_features_and_labels()` sangat lama ketika meload dataset genre. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
6. Jelaskan kenapa harus dilakukan pemisahan data training dan data set sebesar 80 persen? Praktekkan dengan kode dan Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
7. Praktekkan dan jelaskan masing-masing parameter dari fungsi `Sequential()`.Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
8. Praktekkan dan jelaskan masing-masing parameter dari fungsi `compile()`.Tunjukkan keluarannya dengan fungsi `summary` dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
9. Praktekkan dan jelaskan masing-masing parameter dari fungsi `fit()`.Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
10. Praktekkan dan jelaskan masing-masing parameter dari fungsi `evaluate()`.Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
11. Praktekkan dan jelaskan masing-masing parameter dari fungsi `predict()`.Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

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

6.5 Presentasi Tugas

Pada pertemuan ini, diadakan dua 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 ke-materian dengan nilai presentasi yang terpisah masing-masing 100. Jadi ada dua komponen penilaiain pada pertemuan ini yaitu :

1. tugas minggu hari ini dan besok (maks 100). pada chapter ini
2. presentasi tugas kode MFCC dan Neural Network (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(10 praktek dan 10 teori) pertanyaan tentang pema-hamannya menggunakan python untuk kecerdasan buatan. jika presenter tidak bisa menjawab satu pertanyaan asisten maka nilai nol. Jika semua pertanyaan bisa di-jawab maka nilai 100. Presentasi bisa diulang apabila gagal, sampai bisa mendap-atkan nilai 100 dalam waktu satu minggu kedepan.

Chapter 7

CNN

Untuk pratikum saat ini menggunakan buku *Python Artificial Intelligence Projects for Beginners*[2]. Dengan praktek menggunakan python 3 dan editor spyder dan library python keras dan algoritma konvolusi. Kode program ada di <https://github.com/awangga/Python-Artificial-Intelligence-Projects-for-Beginners> . Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti konsep dasar Neural Network pada implementasi vektorisasi(Tokenize) di teks
2. Mengerti teknik Konvolusi
3. Memahami konsep deep learning

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. Tidak menyertakan **pdf** kompilasi **diskon 50%** nilainya.

7.1 Teori

Teori diambil dari buku referensi mengenai Neural Network dan deep learning dari dataset komentar di youtube dan HASYv2. Materi dan praktek bisa dilihat di youtube dosen.

7.2 Soal Teori

Praktek teori penunjang yang dikerjakan total nilai 100 sebagai nilai terpisah dari praktek pada modul ini (nilai 5 per nomor kecuali nomor terakhir 30 sisanya merupakan penanganan error, untuk hari pertama) :

1. Jelaskan kenapa file teks harus dilakukan tokenizer. dilengkapi dengan ilustrasi atau gambar.
2. Jelaskan konsep dasar K Fold Cross Validation pada dataset komentar Youtube pada kode listing 7.1. dilengkapi dengan ilustrasi atau gambar.

```
1 kfold = StratifiedKFold(n_splits=5)
2 splits = kfold.split(d, d['CLASS'])
```

Listing 7.1: K Fold Cross Validation

3. Jelaskan apa maksudnya kode program *for train, test in splits*. dilengkapi dengan ilustrasi atau gambar.
4. Jelaskan apa maksudnya kode program *train_content = d['CONTENT'].iloc[train_idx]* dan *test_content = d['CONTENT'].iloc[test_idx]*. dilengkapi dengan ilustrasi atau gambar.
5. Jelaskan apa maksud dari fungsi *tokenizer = Tokenizer(num_words=2000)* dan *tokenizer.fit_on_texts(train_content)*, dilengkapi dengan ilustrasi atau gambar.
6. Jelaskan apa maksud dari fungsi *d_train_inputs = tokenizer.texts_to_matrix(train_content, mode='tfidf')* dan *d_test_inputs = tokenizer.texts_to_matrix(test_content, mode='tfidf')*, dilengkapi dengan ilustrasi kode dan atau gambar.
7. Jelaskan apa maksud dari fungsi *d_train_inputs = d_train_inputs/np.amax(np.absolute(d_train_inputs))* dan *d_test_inputs = d_test_inputs/np.amax(np.absolute(d_test_inputs))*, dilengkapi dengan ilustrasi atau gambar.
8. Jelaskan apa maksud fungsi dari *d_train_outputs = np_utils.to_categorical(d['CLASS'].iloc[train_idx])* dan *d_test_outputs = np_utils.to_categorical(d['CLASS'].iloc[test_idx])* dalam kode program, dilengkapi dengan ilustrasi atau gambar.
9. Jelaskan apa maksud dari fungsi di listing 7.2. Gambarkan ilustrasi Neural Network nya dari model kode tersebut.

```

1 model = Sequential()
2 model.add(Dense(512, input_shape=(2000,)))
3 model.add(Activation('relu'))
4 model.add(Dropout(0.5))
5 model.add(Dense(2))
6 model.add(Activation('softmax'))

```

Listing 7.2: Membuat model Neural Network

10. Jelaskan apa maksud dari fungsi di listing 7.3 dengan parameter tersebut.

```

1 model.compile(loss='categorical_crossentropy', optimizer='adamax',
2               metrics=['accuracy'])

```

Listing 7.3: Compile model

11. Jelaskan apa itu Deep Learning
12. Jelaskan apa itu Deep Neural Network, dan apa bedanya dengan Deep Learning
13. Jelaskan dengan ilustrasi gambar buatan sendiri (langkah per langkah) bagaimana perhitungan algoritma konvolusi dengan ukuran stride $(NPM \bmod 3 + 1) \times (NPM \bmod 3 + 1)$ yang terdapat max pooling. (nilai 30)

7.3 Praktek Program

Tugas nilai terpisah dari teori maksimal 100. 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). Buka kode program pada repo Python-Artificial-Intelligence-Projects-for-Beginners pada github awangga. Buka folder Chapter04 file MathSymbols.py

1. Jelaskan kode program pada blok # In[1]. Jelaskan arti dari setiap baris kode yang dibuat (harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
2. Jelaskan kode program pada blok # In[2]. Jelaskan arti dari setiap baris kode yang dibuat (harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
3. Jelaskan kode program pada blok # In[3]. Jelaskan arti dari setiap baris kode yang dibuat (harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

4. Jelaskan kode program pada blok # In[4]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
5. Jelaskan kode program pada blok # In[5]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
6. Jelaskan kode program pada blok # In[6]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
7. Jelaskan kode program pada blok # In[7]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
8. Jelaskan kode program pada blok # In[8]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
9. Jelaskan kode program pada blok # In[9]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
10. Jelaskan kode program pada blok # In[10]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
11. Jelaskan kode program pada blok # In[11]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
12. Jelaskan kode program pada blok # In[12]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
13. Jelaskan kode program pada blok # In[13]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

14. Jelaskan kode program pada blok # In[14]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
15. Jelaskan kode program pada blok # In[15]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
16. Jelaskan kode program pada blok # In[16]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
17. Jelaskan kode program pada blok # In[17]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
18. Jelaskan kode program pada blok # In[18]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
19. Jelaskan kode program pada blok # In[19]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.
20. Jelaskan kode program pada blok # In[20]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

7.4 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

7.5 Presentasi Tugas

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

1. tugas teori hari pertama(maks 100) modul ini
2. tugas praktek hari kedua modul ini(maks 100)
3. Presentasi tugas penjelasan CNN dan deep learning, Mempraktekkan kode python dan menjelaskan cara kerjanya(maks 100).

Waktu presentasi pada jam kerja di IRC. Kriteria penilaian presentasi sangat sederhana, presenter akan ditanyai 20(10 praktek dan 10 teori) pertanyaan tentang pemahamannya menggunakan python untuk kecerdasan buatan dan teori konvolusi dan deep learning. 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.

Chapter 8

Perkenalan Generative Adversarial Network

Menggunakan sumber buku [1]. Dengan source code yang ada di github awangga. <https://github.com/awangga/Generative-Adversarial-Networks-Projects> Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti konsep dasar Generative Adversarial Network
2. Mengerti teknik Generator dan diskriminators
3. Memahami penggunaan keras dan tensorflow

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. Tidak menyertakan **pdf** kompilasi **diskon 50%** nilainya.

8.1 Teori

Teori diambil dari buku referensi [1] chapter 1.

8.2 Soal Teori

Praktek teori penunjang yang dikerjakan total nilai 100 sebagai nilai terpisah dari praktek pada modul ini(nilai 6,75 per nomor kecuali nomor terakhir 30 sisanya merupakan penanganan error, untuk hari pertama) :

1. Jelaskan dengan ilustrasi gambar sendiri apa itu generator dengan perumpamaan anda sebagai mahasiswa sebagai generatornya.
2. Jelaskan dengan ilustrasi gambar sendiri apa itu diskriminator dengan perumpamaan dosen anda sebagai diskriminatornya.
3. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur generator dibuat
4. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur diskriminator dibuat
5. Jelaskan dengan ilustrasi gambar apa itu latent space.
6. Jelaskan dengan ilustrasi gambar apa itu adversarial play
7. Jelaskan dengan ilustrasi gambar apa itu Nash equilibrium
8. Sebutkan dan jelaskan contoh-contoh implementasi dari GAN
9. Berikan contoh dengan penjelasan kode program beserta gambar arsitektur untuk membuat generator(neural network) dengan sebuah input layer, tiga hidden layer(dense layer), dan satu output layer(reshape layer)
10. Berikan contoh dengan ilustrasi dari arsitektur diskriminator dengan sebuah input layer, 3 buah hidden layer, dan satu output layer.
11. Jelaskan bagaimana kaitan output dan input antara generator dan diskriminator tersebut. Jelaskan kenapa inputan dan outputan seperti itu.
12. Jelaskan apa perbedaan antara Kullback-Leibler divergence (KL divergence)/relative entropy, Jensen-Shannon(JS) divergence / information radius(iRaD) / total divergence to the average dalam mengukur kualitas dari model.
13. Jelaskan apa itu fungsi objektif yang berfungsi untuk mengukur kesamaan antara gambar yang dibuat dengan yang asli.
14. Jelaskan apa itu scoring algoritma selain mean square error atau cross entropy seperti The Inception Score dan The Frechet Inception distance.
15. Jelaskan kelebihan dan kekurangan GAN

8.3 Praktek Program

Tugas nilai terpisah dari teori maksimal 100. Praktekkan dengan menjalankan kode program nya dan jelaskan (diperlihatkan di video youtube) dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat dan wajib diambil dari layar komputer sendiri masing masing nomor di bawah ini(nilai 3 masing masing pada hari kedua). Buka kode program pada repo Generative-Adversarial-Networks-Projects pada github awangga. Buka folder Chapter02. Kita praktekkan 3D Generative Adversarial Networks (3D-GANs) .

1. Jelaskan apa itu 3D convolutions
2. Jelaskan dengan kode program arsitektur dari generator networknya, beserta penjelasan input dan output dari generator network.
3. Jelaskan dengan kode program arsitektur dari diskriminator network, beserta penjelasan input dan outputnya.
4. Jelaskan proses training 3D-GANs
5. Jelaskan bagaimana melakukan settingan awal chapter 02 untuk memenuhi semua kebutuhan sebelum melanjutkan ke tahapan persiapan data.
6. Jelaskan tentang dataset yang digunakan, dari mulai tempat unduh, cara membuka dan melihat data. Sampai deskripsi dari isi dataset dengan detail penjelasan setiap folder/file yang membuat orang awam paham.
7. Jelaskan apa itu voxel dengan ilustrasi dan bahasa paling awam
8. Visualisasikan dataset tersebut dalam tampilan visual plot, jelaskan cara melakukan visualisasinya
9. buka file run.py jelaskan perbaris kode pada fungsi untuk membuat generator yaitu build_generator.
10. jelaskan juga fungsi untuk membangun diskriminator pada fungsi build_discriminator.
11. jelaskan apa maksud dari kode program `__name__ == '__main__'`

```
1 if __name__ == '__main__':
```

Listing 8.1: Kode program utama

12. jelaskan secara detail perbaris dan per parameter apa arti dari kode program :

```
1 object_name = "chair"
2 data_dir = "data/3DShapeNets/volumetric_data/" \
3           "{}/30/train/*.mat".format(object_name)
4 gen_learning_rate = 0.0025
5 dis_learning_rate = 10e-5
6 beta = 0.5
7 batch_size = 1
8 z_size = 200
9 epochs = 10
```

Listing 8.2: Setting Parameter

13. Jelaskan secara detail dari kode program pembuatan dan kompilasi arsitektur berikut :

```
1 gen_optimizer = Adam(lr=gen_learning_rate, beta_1=beta)
2 dis_optimizer = Adam(lr=dis_learning_rate, beta_1=beta)
3
4 discriminator = build_discriminator()
5 discriminator.compile(loss='binary_crossentropy', optimizer=
6 dis_optimizer)
7
8 generator = build_generator()
9 generator.compile(loss='binary_crossentropy', optimizer=
10 gen_optimizer)
```

Listing 8.3: Setting Parameter

14. Jelaskan secara detail kode program untuk membuat dan melakukan kompilasi model adversarial berikut:

```
1 discriminator.trainable = False
2
3 input_layer = Input(shape=(1, 1, 1, z_size))
4 generated_volumes = generator(input_layer)
5 validity = discriminator(generated_volumes)
6 adversarial_model = Model(inputs=[input_layer], outputs=[
7 validity])
8 adversarial_model.compile(loss='binary_crossentropy', optimizer=
9 gen_optimizer)
```

Listing 8.4: Membuat dan Kompilasi Model Adversarial

15. Jelaskan Ekstrak dan load data kursi dengan menggunakan fungsi getVoxels-Format dan get3DImages yang digunakan pada kode program berikut :

```
1 print("Loading data...")
2 volumes = get3DImages(data_dir=data_dir)
3 volumes = volumes[..., np.newaxis].astype(np.float)
4 print("Data loaded...")
```

Listing 8.5: Ekstraksi dan load dataset

16. Jelaskan maksud dari kode program instansiasi TensorBoard yang menambahkan generator dan diskriminator pada program berikut:

```
1  tensorboard = TensorBoard(log_dir="logs/{}".format(time.time()))
2  )
3  tensorboard.set_model(generator)
4  tensorboard.set_model(discriminator)
```

Listing 8.6: Instansiasi tensorboard

17. Jelaskan apa fungsi dari np.reshape ones zeros pada kode program berikut dengan parameteranya:

```
1  labels_real = np.reshape(np.ones((batch_size,)), (-1, 1, 1, 1, 1))
2  labels_fake = np.reshape(np.zeros((batch_size,)), (-1, 1, 1, 1, 1))
```

Listing 8.7: Pelabelan dataset

18. Jelaskan kenapa harus ada perulangan dalam meraih epoch. Dan jelaskan apa itu epoch terkait kode program berikut:

```
1  for epoch in range(epochs):
2      print("Epoch:", epoch)
3
4      gen_losses = []
5      dis_losses = []
```

Listing 8.8: Setting Epoch

19. Jelaskan apa itu batches dan kaitannya dengan kode program berikut, dan kenapa berada di dalam epoch:

```
1  number_of_batches = int(volumes.shape[0] / batch_size)
2  print("Number of batches:", number_of_batches)
3  for index in range(number_of_batches):
4      print("Batch:", index + 1)
```

Listing 8.9: Setting Batch

20. Berikut adalah kode program pengambilan gambar dan noise. Jelaskan apa fungsi np.random.normal serta astype, serta jelaskan apa arti parameter titik dua dan jelaskan isi dari z_sample dan volumes_batch:

```
1  z_sample = np.random.normal(0, 0.33, size=[
2  batch_size, 1, 1, 1, z_size]).astype(np.float32)
3  volumes_batch = volumes[index * batch_size:(index +
4  1) * batch_size, :, :, :]
```

Listing 8.10: Set real images dan vektor noise

21. Berikut adalah kode program generator gambar palsu. Jelaskan apa fungsi `generator.predict_on_batch`, serta jelaskan apa arti parameter `z_sample`:

```
1         # Next, generate volumes using the generate network
2         gen_volumes = generator.predict_on_batch(z_sample)
```

Listing 8.11: Generator Gambar Palsu

22. Berikut adalah kode program training diskriminator dengan gambar palsu dari generator dan gambar asil. Jelaskan apa maksudnya harus dilakukan training diskriminator secara demikian dan jelaskan apa isi `loss_fake` dan `loss_real` serta `d_loss` dan fungsi `train_on_batch`.

```
1         discriminator.trainable = True
2         if index % 2 == 0:
3             loss_real = discriminator.train_on_batch(
volumes_batch, labels_real)
4             loss_fake = discriminator.train_on_batch(
gen_volumes, labels_fake)
5
6             d_loss = 0.5 * np.add(loss_real, loss_fake)
7             print("d_loss:{}".format(d_loss))
8
9         else:
10            d_loss = 0.0
```

Listing 8.12: Training Diskriminator

23. Berikut adalah kode program training model adversarial yang terdapat generator dan diskriminator. Jelaskan apa bagaimana proses terbentuknya parameter `z` dan `g_loss`:

```
1         z = np.random.normal(0, 0.33, size=[batch_size, 1,
1, 1, z_size]).astype(np.float32)
2         g_loss = adversarial_model.train_on_batch(z,
labels_real)
3         print("g_loss:{}".format(g_loss))
4
5         gen_losses.append(g_loss)
6         dis_losses.append(d_loss)
```

Listing 8.13: Training adversarial model

24. Berikut adalah kode program generate dan menyimpan gambar 3D setelah beberapa saat setiap epoch. Jelaskan mengapa ada perulangan dengan parameter tersebut, serta jelaskan arti setiap variabel beserta perlihatkan isinya dan artikan isinya :

```

1          # Every 10th mini-batch, generate volumes and save
    them
2          if index % 10 == 0:
3              z_sample2 = np.random.normal(0, 0.33, size=[
    batch_size, 1, 1, 1, z_size]).astype(np.float32)
4              generated_volumes = generator.predict(z_sample2
    , verbose=3)
5              for i, generated_volume in enumerate(
    generated_volumes[:5]):
6                  voxels = np.squeeze(generated_volume)
7                  voxels[voxels < 0.5] = 0.
8                  voxels[voxels >= 0.5] = 1.
9                  saveFromVoxels(voxels, "results/img-{}-{}-
    {}".format(epoch, index, i))

```

Listing 8.14: Buat dan simpan gambar 3D

25. Berikut adalah kode program menyimpan average losses setiap epoch. Jelaskan apa itu tensorboard dan setiap parameter yang digunakan pada kode program ini :

```

1          # Write losses to Tensorboard
2          write_log(tensorboard, 'g_loss', np.mean(gen_losses),
    epoch)
3          write_log(tensorboard, 'd_loss', np.mean(dis_losses),
    epoch)

```

Listing 8.15: Simpan Average losses setiap epoch

26. Berikut adalah kode program menyimpan model. Jelaskan apa itu format h5 dan penjelasan dari kode program berikut :

```

1          generator.save_weights(os.path.join("models", "
    generator_weights.h5"))
2          discriminator.save_weights(os.path.join("models", "
    discriminator_weights.h5"))

```

Listing 8.16: Simpan model

27. Berikut adalah kode program testing model. Jelaskan dengan ilustrasi gambar dari mulai meload hingga membuat gambar 3D dengan menggunakan z_sample, bisakah parameter z_sample tersebut diubah? :

```

1          # Create models
2          generator = build_generator()
3          discriminator = build_discriminator()
4
5          # Load model weights
6          generator.load_weights(os.path.join("models", "
    generator_weights.h5"), True)
7          discriminator.load_weights(os.path.join("models", "
    discriminator_weights.h5"), True)

```

```

8
9      # Generate 3D models
10     z_sample = np.random.normal(0, 1, size=[batch_size, 1, 1,
11     1, z_size]).astype(np.float32)
12     generated_volumes = generator.predict(z_sample, verbose=3)
13
14     for i, generated_volume in enumerate(generated_volumes[:2]):
15         voxels = np.squeeze(generated_volume)
16         voxels[voxels < 0.5] = 0.
17         voxels[voxels >= 0.5] = 1.
18         saveFromVoxels(voxels, "results/gen_{}".format(i))

```

Listing 8.17: Testing model

8.4 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. skrintut error
2. Tuliskan kode eror dan jenis errornya
3. Solusi pemecahan masalah error tersebut

8.5 Presentasi Tugas

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

1. tugas teori hari pertama(maks 100) modul ini
2. tugas praktek hari kedua modul ini(maks 100)
3. Presentasi tugas penjelasan CNN dan deep learning, Mempraktekkan kode python dan menjelaskan cara kerjanya(maks 100).

Waktu presentasi pada jam kerja di IRC. Kriteria penilaian presentasi sangat sederhana, presenter akan ditanyai 20(10 praktek dan 10 teori) pertanyaan tentang pema-
hamannya menggunakan python untuk kecerdasan buatan dan teori konvolusi dan
deep learning. jika presenter tidak bisa menjawab satu pertanyaan asisten maka ni-
lai nol. Jika semua pertanyaan bisa dijawab maka nilai 100. Presentasi bisa diulang
apabila gagal, sampai bisa mendapatkan nilai 100 dalam waktu satu minggu kedepan.

Chapter 9

Conditional Generative Adversarial Network

Menggunakan sumber buku [1]. Dengan source code yang ada di github awangga. <https://github.com/awangga/Generative-Adversarial-Networks-Projects> Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti konsep Conditional Generative Adversarial Network
2. Mengerti teknik CGAN
3. Memahami penggunaan keras untuk cGAN

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. Tidak menyertakan **pdf** kompilasi **diskon 50%** nilainya.

9.1 Soal Teori

Teori diambil dari buku referensi [1] chapter 3. Praktek teori penunjang yang dikerjakan total nilai 100 sebagai nilai terpisah dari praktek pada modul ini (nilai 9,09 per nomor kecuali nomor terakhir 30 sisanya merupakan penanganan error, untuk hari pertama) :

1. Jelaskan dengan ilustrasi gambar sendiri apa perbedaan antara vanilla GAN dan cGAN.

2. Jelaskan dengan ilustrasi gambar sendiri arsitektur dari Age-cGAN.
3. Jelaskan dengan ilustrasi gambar sendiri arsitektur encoder network dari Age-cGAN.
4. Jelaskan dengan ilustrasi gambar sendiri arsitektur generator network dari Age-cGAN.
5. Jelaskan dengan ilustrasi gambar sendiri arsitektur discriminator network dari Age-cGAN.
6. Jelaskan dengan ilustrasi gambar apa itu pretrained Inception-ResNet-2 Model.
7. Jelaskan dengan ilustrasi gambar sendiri arsitektur Face recognition network Age-cGAN.
8. Sebutkan dan jelaskan serta di sertai contoh-contoh tahapan dari Age-cGAN
9. Berikan contoh perhitungan fungsi training objektif
10. Berikan contoh dengan ilustrasi penjelasan dari Initial latent vector approximation
11. Berikan contoh perhitungan latent vector optimization

9.2 Praktek Program

Tugas nilai terpisah dari teori maksimal 100. Praktekkan dengan menjalankan kode program nya dan jelaskan (diperlihatkan di video youtube) dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat dan wajib diambil dari layar komputer sendiri masing masing nomor di bawah ini(nilai 15 masing masing pada hari kedua). Buka kode program pada repo Generative-Adversarial-Networks-Projects pada github awangga. Buka folder Chapter03. Kita praktekkan Age Conditional Generative Adversarial Networks (Age-cGAN). Dataset ada di <https://drive.google.com/open?id=1NoV357Zveml>

1. Jelaskan bagaimana cara ekstrak file dataset Age-cGAN menggunakan google colab
2. Jelaskan bagaimana kode program bekerja untuk melakukan load terhadap dataset yang sudah di ekstrak, termasuk bagaimana penjelasan kode program perhitungan usia

3. Jelaskan bagaimana kode program The Encoder Network bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana
4. Jelaskan bagaimana kode program The Generator Network bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana
5. Jelaskan bagaimana kode program The Discriminator Network bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana
6. Jelaskan bagaimana kode program Training cGAN bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana
7. Jelaskan bagaimana kode program Initial dan latent vector approximation bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana

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

Chapter 10

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 11

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 12

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 13

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 14

Discussion

Please tell more about conclusion and how to the next work of this study.

Appendix A

Form Penilaian Jurnal

gambar A.1 dan A.2 merupakan contoh bagaimana reviewer menilai jurnal kita.

NO	UNSUR	KETERANGAN	MAKS	KETERANGAN
1	Keefektifan Judul Artikel	Maksimal 12 (dua belas) kata dalam Bahasa Indonesia atau 10 (sepuluh) kata dalam Bahasa Inggris	2	a. Tidak lugas dan tidak ringkas (0) b. Kurang lugas dan kurang ringkas (1) c. Ringkas dan lugas (2)
2	Pencantuman Nama Penulis dan Lembaga Penulis		1	a. Tidak lengkap dan tidak konsisten (0) b. Lengkap tetapi tidak konsisten (0,5) c. Lengkap dan konsisten (1)
3	Abstrak	Dalam Bahasa Indonesia dan Bahasa Inggris yang baik, jumlah 150-200 kata. Isi terdiri dari latar belakang, metode, hasil, dan kesimpulan. Isi tertuang dengan kalimat yang jelas.	2	a. Tidak dalam Bahasa Indonesia dan Bahasa Inggris (0) b. Abstrak kurang jelas dan ringkas, atau hanya dalam Bahasa Inggris, atau dalam Bahasa Indonesia saja (1) c. Abstrak yang jelas dan ringkas dalam Bahasa Indonesia dan Bahasa Inggris (2)
4	Kata Kunci	Maksimal 5 kata kunci terpenting dalam paper	1	a. Tidak ada (0) b. Ada tetapi kurang mencerminkan konsep penting dalam artikel (0,5) c. Ada dan mencerminkan konsep penting dalam artikel (1)
5	Sistematika Pembahasan	Terdiri dari pendahuluan, tinjauan pustaka, metode penelitian, hasil dan pembahasan, kesimpulan dan saran, daftar pustaka	1	a. Tidak lengkap (0) b. Lengkap tetapi tidak sesuai sistematika (0,5) c. Lengkap dan bersistem (1)
6	Pemanfaatan Instrumen Pendukung	Pemanfaatan Instrumen Pendukung seperti gambar dan tabel	1	a. Tidak dimanfaatkan (0) b. Kurang informatif atau komplementer (0,5) c. Informatif dan komplementer (1)
7	Cara Pengacuan dan Pengutipan		1	a. Tidak baku (0) b. Kurang baku (0,5) c. Baku (1)
8	Penyusunan Daftar Pustaka	Penyusunan Daftar Pustaka	1	a. Tidak baku (0) b. Kurang baku (0,5) c. Baku (1)
9	Peristilahan dan Kebahasaan		2	a. Buruk (0) b. Baik (1) c. Cukup (2)
10	Makna Sumbangan bagi Kemajuan		4	a. Tidak ada (0) b. Kurang (1) c. Sedang (2) d. Cukup (3) e. Tinggi (4)

Figure A.1: Form nilai bagian 1.

11	Dampak Ilmiah		7	a. Tidak ada (0) b. Kurang (1) c. Sedang (3) d. Cukup (5) e. Besar (7)
12	Nisbah Sumber Acuan Primer berbanding Sumber lainnya	Sumber acuan yang langsung merujuk pada bidang ilmiah tertentu, sesuai topik penelitian dan sudah teruji.	3	a. < 40% (1) b. 40-80% (2) c. > 80% (3)
13	Derajat Kemutakhiran Pustaka Acuan	Derajat Kemutakhiran Pustaka Acuan	3	a. < 40% (1) b. 40-80% (2) c. > 80% (3)
14	Analisis dan Sintesis	Analisis dan Sintesis	4	a. Sedang (2) b. Cukup (3) c. Baik (4)
15	Penyimpulan	Sangat jelas relevasinya dengan latar belakang dan pembahasan, dirumuskan dengan singkat	3	a. Kurang (1) b. Cukup (2) c. Baik (3)
16	Unsur Plagiat		0	a. Tidak mengandung plagiat (0) b. Terdapat bagian-bagian yang merupakan plagiat (-5) c. Keseluruhannya merupakan plagiat (-20)
TOTAL			36	
Catatan : Nilai minimal untuk diterima 25				

Figure A.2: form nilai bagian 2.

Appendix B

FAQ

M : Kalo Intership II atau TA harus buat aplikasi ? D : Ga harus buat aplikasi tapi harus ngoding

M : Pa saya bingung mau ngapain, saya juga bingung mau presentasi apa? D : Makanya baca de, buka jurnal topik ‘ganteng’ nah kamu baca dulu sehari 5 kali ya, 4 hari udah 20 tuh. Bingung itu tanda kurang wawasan alias kurang baca.

M : Pa saya sudah cari jurnal terindeks scopus tapi ga nemu. D : Kamu punya mata de? coba dicolok dulu. Kamu udah lakuin apa aja? tolong di list laporkan ke grup Tingkat Akhir. Tinggal buka google scholar klik dari tahun 2014, cek nama jurnalnya di scimagojr.com beres.

M : Pa saya belum dapat tempat intership, jadi ga tau mau presentasi apa? D : kamu kok ga nyambung, yang dipresentasikan itu yang kamu baca bukan yang akan kamu lakukan.

M : Pa ini jurnal harus yang terindex scopus ga bisa yang lain ? D : Index scopus menandakan artikel tersebut dalam standar semantik yang mudah dipahami dan dibaca serta bukan artikel asal jadi. Jika diluar scopus biasanya lebih sukar untuk dibaca dan dipahami karena tidak adanya proses review yang baik dan benar terhadap artikel.

M : Pa saya tidak mengerti D : Coba lihat standar alasan

M : Pa saya bingung D : Coba lihat standar alasan

M : Pa saya sibuk D : Mbahmu....

M : Pa saya ganteng D : Ndasmu....

M : Pa saya kece D : wes karepmu lah....

Biasanya anda memiliki alasan tertentu jika menghadapi kendala saat proses bimbingan, disini saya akan melakukan standar alasan agar persepsi yang diterima sama dan tidak salah kaprah. Penggunaan kata alasan tersebut antara lain :

1. Tidak Mengerti : anda boleh menggunakan alasan ini jika anda sudah melakukan tahapan membaca dan meresumekan 15 jurnal. Sudah mencoba dan mempraktekkan teorinya dengan mencari di youtube dan google minimal 6 jam sehari selama 3 hari berturut-turut.

2. Bingung : anda boleh mengatakan alasan bingung setelah maksimal dalam berusaha menyelesaikan tugas bimbingan dari dosen(sudah dilakukan semua). Anda belum bisa mengatakan alasan bingung jika anda masih belum menyelesaikan tugas bimbingan dan poin nomor 1 diatas. Setelah anda menyelesaikan tugas bimbingan secara maksimal dan tahap 1 poin diatas, tapi anda masih tetap bingung maka anda boleh memakai alasan ini.

Bibliography

- [1] Kailash Ahirwar. *Generative Adversarial Networks Projects: Build next-generation generative models using TensorFlow and Keras*. Packt Publishing Ltd, 2019.
- [2] Joshua Eckroth. *Python Artificial Intelligence Projects for Beginners: Get up and running with Artificial Intelligence using 8 smart and exciting AI applications*. Packt Publishing Ltd, 2018.
- [3] David L Poole and Alan K Mackworth. *Python code for artificial intelligence: Foundations of computational agents*, 2017.
- [4] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.