

Modul Praktikum Kecerdasan Buatan



Nur Ikhsani Suwandy Putri
1194029

Applied Bachelor of Informatics Engineering
Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering
Politeknik Pos Indonesia

Bandung 2022

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

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	9
2	Membangun Model Prediksi	10
2.1	Teori	10
2.2	scikit-learn	17
2.3	Penanganan Error	29
3	Prediksi dengan Random Forest	32
3.1	Teori	32
3.1.1	Random Forest	33
3.1.2	Confusion Matrix	35
3.1.3	Mencoba dengan metode Decission Tree dan SVM	36
3.1.4	Pengecekan Cross Validation	37
3.1.5	Pengamatan komponen informasi	37
3.2	Soal Teori	38
3.3	Praktek Program	38
3.4	Penanganan Error	39
3.5	Presentasi Tugas	40
4	Klasifikasi Teks	41
4.1	Teori	41
4.1.1	Vektorisasi data	41
4.1.2	Klasifikasi dengan Random Forest	43
4.1.3	Confusion Matrix	43
4.1.4	Pengecekan Cross Validation	43
4.2	Soal Teori	44

4.3	Praktek Program	44
4.4	Penanganan Error	45
4.5	Presentasi Tugas	46
5	Vektorisasi kata dan dokumen	47
5.1	Teori	47
5.2	Soal Teori	47
5.3	Praktek Program	48
5.4	Penanganan Error	49
5.5	Presentasi Tugas	49
6	MFCC dan Neural Network	51
6.1	Teori	51
6.2	Soal Teori	51
6.3	Praktek Program	52
6.4	Penanganan Error	53
6.5	Presentasi Tugas	54
7	CNN	55
7.1	Teori	55
7.2	Soal Teori	56
7.3	Praktek Program	57
7.4	Penanganan Error	59
7.5	Presentasi Tugas	60
8	Perkenalan Generative Adversarial Network	61
8.1	Teori	61
8.2	Soal Teori	61
8.3	Praktek Program	63
8.4	Penanganan Error	68
8.5	Presentasi Tugas	68
9	Conditional Generative Adversarial Network	70
9.1	Soal Teori	70
9.2	Praktek Program	71
9.3	Penanganan Error	72
10	Discussion	73

11 Discussion	74
12 Discussion	75
13 Discussion	76
14 Discussion	77
A Form Penilaian Jurnal	78
B FAQ	81
Bibliography	83

List of Figures

A.1	Form nilai bagian 1.	79
A.2	form nilai bagian 2.	80

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. Definisi Sejarah serta Perkembangan Kecerdasan Buatan

Definisi dari Kecerdasan buatan adalah suatu teknologi yang terdapat pada bidang ilmu komputer yang dapat melakukan manipulasi kecerdasan manusia yang dimasukkan pada mesin agar dapat menyelesaikan persoalan dan pekerjaan seperti yang dilakukan oleh manusia atau mungkin dapat lebih baik dari

manusia. Adapun sejarah dari kecerdasan buatan atau dapat disebut Artificial Intelligence bermula ketika munculnya komputer pada tahun 1940 pada saat itu banyak sekali perhatian yang dapat difokuskan pada kemampuan komputer dalam mengerjakan sesuatu yang dapat dilakukan oleh manusia. komputer dapat meniru kemampuan kecerdasan dan perilaku manusia. pada tahun 1956 ilmuwan seperti Alan Turing, Norbert Wiener dan yang lainnya melakukan kerja sama dalam beberapa ilmu seorang ilmuwan mendapatkan suatu ide untuk membuat kecerdasan buatan sehingga suatu kecerdasan tersebut dibuat agar dapat meniru perilaku manusia. adapun perkembangan dari kecerdasan buatan yaitu pada saat ini banyak sekali jenis mesin yang dapat berperilaku layaknya manusia dengan beragam kemampuan yang diberikan contohnya yaitu google assistant.

2. Supervised Learning dan Unsupervised Learning

Supervised Learning adalah suatu pembelajaran yang didalamnya terdapat pengawasan atau dapat diartikan supervisor. Supervisor merupakan suatu label yang ada di setiap data nya. Kemudian label tersebut berisi tag dari data yang ditambahkan kedalam model pembelajaran mesin atau lebih trend disebut dengan machine learning model. Unsupervised Learning Merupakan suatu pembelajaran tanpa adanya sebuah pengawasan dan tidak menggunakan label untuk bisa memprediksi target variabel

3. Klasifikasi dan Regresi

Klasifikasi merupakan sampel yang dimiliki oleh dua atau lebih kelas yang dikelompokkan yang disesuaikan berdasarkan ukuran kemiripan atau jarak yang melekat. Regresi merupakan sebuah prediksi apabila hasil atau output yang diinginkan terdiri dari satu atau lebih variabel. Teknik klasifikasi menyediakan model atau fungsi prediktif yang memprediksi data baru dalam kategori atau label tersendiri dengan bantuan data historis. Sebaliknya, metode regresi memodelkan fungsi bernilai kontinu yang berarti memprediksi data dalam data numerik kontinu.

4. Dataset Training set dan Testing set

Dataset merupakan kumpulan objek yang merepresentasikan data dan juga relasi yang ada di memory yang bersifat homogen. Trainingset Merupakan sebuah data yang digunakan untuk melakukan klasifikasi ataupun prediksi. Dengan

adanya data training maka akan didapatkan sebuah model regresi. Testingset Digunakan untuk menguji kebenaran dari sebuah model data.yang berisi unseen example merupakan contoh yang tidak ada didalam trainingset.

1.2 Instalasi

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

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

```
yunita@yunita: ~  
File Edit View Search Terminal Help  
yunita@yunita:~$ pip3 install -U scikit-learn  
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.6/dist-packages (0.24.2)  
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (from scikit-learn) (1.19.5)  
Requirement already satisfied: scipy>=0.19.1 in /usr/local/lib/python3.6/dist-packages (from scikit-learn) (1.5.4)  
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.6/dist-packages (from scikit-learn) (3.1.0)  
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.6/dist-packages (from scikit-learn) (1.1.0)  
yunita@yunita:~$
```

```
yunita@yunita: ~  
File Edit View Search Terminal Help  
yunita@yunita:~$ python3 -m pip show scikit-learn  
Name: scikit-learn  
Version: 0.19.1  
Summary: A set of python modules for machine learning and data mining  
Home-page: http://scikit-learn.org  
Author: Andreas Mueller  
Author-email: amueller@ais.uni-bonn.de  
License: new BSD  
Location: /home/yunita/anaconda3/lib/python3.6/site-packages  
Requires:  
Required-by:  
yunita@yunita:~$
```

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

```
from sklearn import datasets
... iris = datasets.load_iris()
... digits = datasets.load_digits()
... print(digits.data)

In [1]: ...: ...: ...: from sklearn import datasets
iris = datasets.load_iris()
digits = datasets.load_digits()
print(digits.data)
[[ 0.  0.  5. ...  0.  0.  0.]
 [ 0.  0.  0. ... 10.  0.  0.]
 [ 0.  0.  0. ... 16.  9.  0.]
 ...
 [ 0.  0.  1. ...  6.  0.  0.]
 [ 0.  0.  2. ... 12.  0.  0.]
 [ 0.  0. 10. ... 12.  1.  0.]]

9 from sklearn import datasets
10 iris = datasets.load_iris()
11 digits = datasets.load_digits()
12 print(digits.data)
```

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

```
5 from sklearn import svm
6 clf = svm.SVC(gamma=0.001, C=100.)
7 clf.fit(digits.data[:-1], digits.target[:-1])
8 clf.predict(digits.data[-1:])
9
In [2]: from sklearn import svm
...: clf = svm.SVC(gamma=0.001, C=100.)
...: clf.fit(digits.data[:-1], digits.target[:-1])
...: clf.predict(digits.data[-1:])
Out[2]: array([8])
```

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

```
0 from sklearn import svm
1 from sklearn import datasets
2 clf = svm.SVC()
3 X, y = datasets.load_iris(return_X_y=True)
4 clf.fit(X, y)
5
6 import pickle
7 s = pickle.dumps(clf)
8 clf2 = pickle.loads(s)
9 clf2.predict(X[0:1])
0
1 from joblib import dump, load
2 dump(clf, '1194029.joblib')
3 clf = load('1194029.joblib')
4
5 print(clf)
6
```

```
In [3]: from sklearn import svm
...: from sklearn import datasets
...: clf = svm.SVC()
...: X, y = datasets.load_iris(return_X_y=True)
...: clf.fit(X, y)
Out[3]: SVC()
```

```
In [4]: import pickle
...: s = pickle.dumps(clf)
...: clf2 = pickle.loads(s)
...: clf2.predict(X[0:1])
Out[4]: array([0])
```

```
In [5]: from joblib import dump, load
...: dump(clf, '1194029.joblib')
...: clf = load('1194029.joblib')
...:
...: print(clf)
SVC()
```

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

```
import numpy as np
from sklearn import random_projection

rng = np.random.RandomState(0)
X = rng.rand(10, 2000)
X = np.array(X, dtype='float32')
X.dtype

transformer = random_projection.GaussianRandomProjection()
X_new = transformer.fit_transform(X)
X_new.dtype

print(X_new)

...: print(X_new)
[[-0.16240584  1.50298719  0.10490307 ... -0.17604244 -0.25166146
 -0.78564333]
 [ 0.31146686  1.07127377 -0.40350508 ...  0.12386513 -0.52107496
 -1.02234101]
 [ 0.31035335  0.87644647  0.03159358 ... -0.35523213 -0.19171858
 -0.83131909]
 ...
 [ 0.38190054  1.23431236  0.45530212 ... -0.22952216 -0.06537192
 -0.87578668]
 [ 0.21711962  1.00946652 -0.83185103 ...  0.24009344  0.118436
 -0.34095977]
 [ 0.19595825  1.32622775  0.18703105 ... -0.23216707  0.09286661
 -0.52036458]]

In [7]:
```


1.3 Penanganan Error

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

1. skrinsut error[hari ke 2](10)
2. Tuliskan kode eror dan jenis errornya [hari ke 2](10)
3. Solusi pemecahan masalah error tersebut[hari ke 2](10)

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.

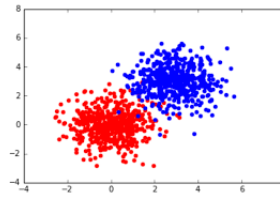
Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten riset. Kode program menggunakan input listing ditaruh di folder src ekstensi .py dan dipanggil ke latex dengan input listings. Tulisan dan kode tidak boleh plagiat, menggunakan bahasa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

2.1 Teori

Nur Ikhsani Suwandy Futri
1194029

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

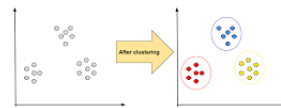
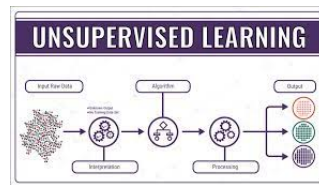
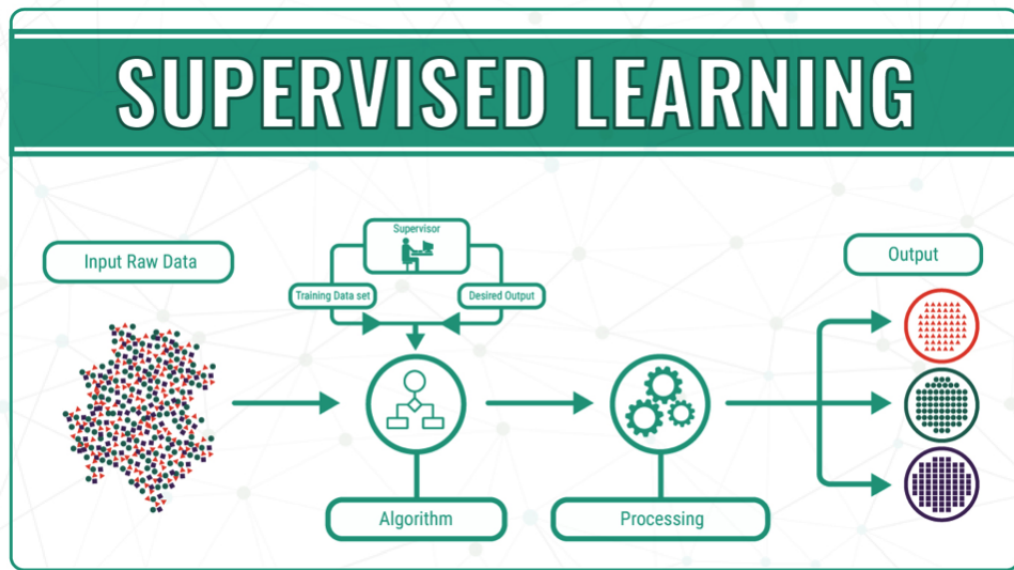
1. Jelaskan apa itu binary classification dilengkapi ilustrasi gambar sendiri



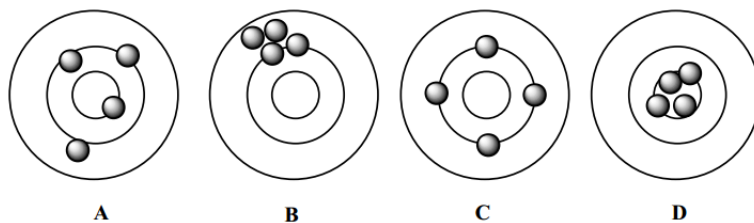
binary classification adalah tugas mengklasifikasikan suatu elemen-elemen dari himpunan yang diberikan ke dalam dua kelompok, dengan berdasarkan aturan klasifikasi. konteks yang dapat membutuhkan suatu keputusan pada suatu item yang memiliki sifat kualitatif atau tidak, adapun beberapa klasifikasi biner khas yaitu :

- Tes medis yang digunakan untuk melakukan keputusan pasien mempunyai penyakit tertentu atau tidak properti klasifikasi yaitu keberadaan penyakit.
- Mode uji yaitu lulus atau gagalnya suatu kontrol pabrik yang dapat memutuskan apakah suatu spesifikasi telah terpenuhi.
- Pengambilan informasi yaitu melakukan pemutusan suatu halaman atau artikel harus dalam hasil pencarian atau tidak properti klasifikasi yaitu relevan artikel.

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



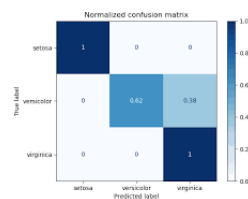
1. Supervised adalah suatu pendekatan machine learning yang dapat ditentukan dengan berdasarkan penggunaan dataset berlabel.
 2. Unsepervised learning adalah suatu metode pembelajaran dengan menggunakan algoritma machine learning yang dapat digunakan untuk menganalisis serta mengelompokkan kumpulan data yang tidak berlabel.
 3. Clustering adalah suatu metode penganalisaan data yang sering dimasukkan pada salah satu metode data maining yang mempunyai tujuan untuk mengelompokkan data dengan karakteristik yang sama pada satu wilayah yang sama dengan data.
3. Jelaskan apa itu evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar sendiri.



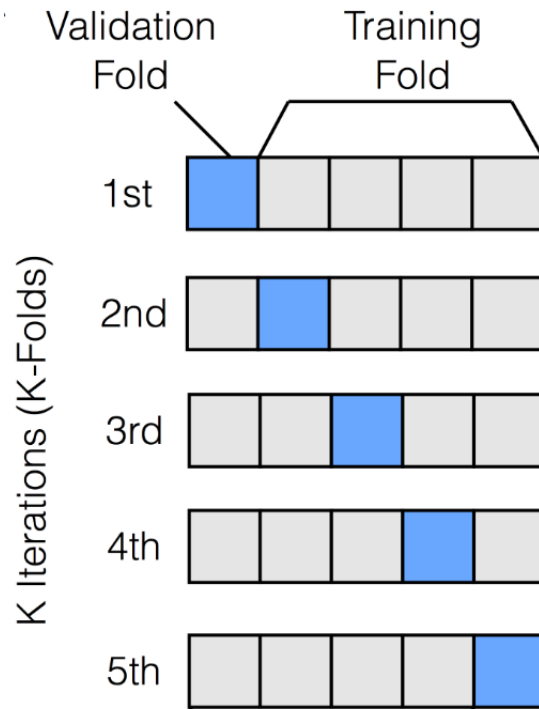
Evaluasi adalah suatu kegiatan yang terencana agar dapat mengukur menilai dan keberhasilan suatu program, evaluasi juga merupakan suatu cara yang terbaik untuk menguji efektivitas dan produktifitas.

Akurasi adalah suatu ukuran kedekatan dari hasil pengukuran dengan nilai yang sebenarnya atau nilai target.

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.



K-fold cross validation adalah suatu metode yang digunakan untuk mendapatkan kinerja classifier, metode ini digunakan dengan jumlah data yang terbatas. cara kerja K-fold cross validation yaitu

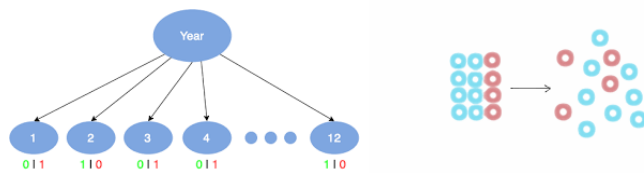
- total intance dibagi menjadi N bagian.
- fold-1 ketika bagian 1 manjadi data uji dan yang lainnya menjadi data pelatihan. sehingga dapat dihitung keakuratan berdasarkan porsi data. perhitungan akuasi menggunakan persamaan berikut Akurasi = $\frac{\text{sigma data uji benar klasifikasi}}{\text{sigma total data uji}} \times 100$.
- fold 2 pada bagian ke-2 menjadi data uji dan yang lainnya menjadi data pelatihan. sehingga hitung keakuratan berdasarkan porsi data 4 hingga seterusnya mencapai fold ke- K. pada hitungan rata-rata akursi dari K buah akurasi pada bagian atas. rata-rata akurasi ini menjadi akurasi final.

6. Jelaskan apa itu decision tree dengan gambar ilustrasi contoh buatan sendiri.



Decision tree adalah alat pendukung dengan struktur seperti pohon yang memodelkan kemungkinan hasil, biaya sumber daya, utilitas, dan kemungkinan konsekuensi. Decision tree menyediakan cara untuk menyajikan algoritma dengan pernyataan kontrol bersyarat. Mereka termasuk cabang yang mewakili langkah-langkah pengambilan keputusan yang dapat mengarah pada hasil yang menguntungkan.

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



Informasi gain adalah teknik seleksi fitur yang memakai metode scoring untuk nominal ataupun pembobotan atribut kontinu yang didiskretkan menggunakan maksimal entropy. Suatu entropy digunakan untuk mendefinisikan nilai Information Gain. Entropy menggambarkan banyaknya informasi yang dibutuhkan untuk mengkodekan suatu kelas.

Entropi adalah salah satu besaran termodinamika yang mengukur energi dalam sistem per satuan temperatur yang tak dapat digunakan untuk melakukan usaha.

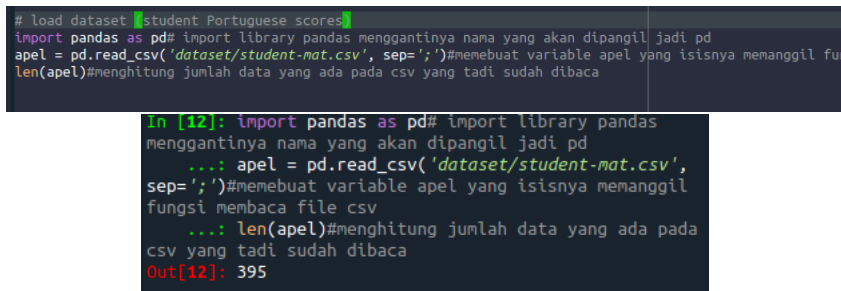
2.2 scikit-learn

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)//
```

saya Nur Ikhsani Suwandy Futri 1194029 mod 3 = 2 jadi menggunakan buah.//



```
# load dataset (student Portuguese scores)
import pandas as pd# import library pandas menggantinya nama yang akan dipanggil jadi pd
apel = pd.read_csv('dataset/student-mat.csv', sep=';')#membuat variable apel yang isisnya memanggil fun
len(apel)#menghitung jumlah data yang ada pada csv yang tadi sudah dibaca

In [12]: import pandas as pd# import library pandas
menggantinya nama yang akan dipanggil jadi pd
...: apel = pd.read_csv('dataset/student-mat.csv',
...: sep=';')#membuat variable apel yang isisnya memanggil
fungsi membaca file csv
...: len(apel)#menghitung jumlah data yang ada pada
csv yang tadi sudah dibaca
Out[12]: 395
```

```

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

```

```

# In[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()

```

	school	sex	age	address	famsize	...	Dalc	Walc
0	GP	F	18	U	GT3	...	1	1
1	GP	F	17	U	GT3	...	1	1
2	GP	F	15	U	LE3	...	2	3
3	GP	F	15	U	GT3	...	1	1
4	GP	F	16	U	GT3	...	1	2

Index	sup	paid	activities	nursery	higher	internet	romantic	famrel	freetime	qoout	D
0		no	no	yes	yes	no	no	4	3	4	1
1		no	no	no	yes	yes	no	5	3	3	1
2		yes	no	yes	yes	yes	no	4	3	2	2
3		yes	yes	yes	yes	yes	yes	3	2	2	1
4		yes	no	yes	yes	no	no	4	3	2	1
5		yes	yes	yes	yes	yes	no	5	4	2	1
6		no	no	yes	yes	yes	no	4	4	4	1
7		no	no	yes	yes	no	no	4	1	4	1
8		yes	no	yes	yes	yes	no	4	2	2	1
9		yes	yes	yes	yes	yes	no	5	5	1	1
10		yes	no	yes	yes	yes	no	3	3	3	1
11		no	yes	yes	yes	yes	no	5	2	2	1
12		yes	yes	yes	yes	yes	no	4	3	3	1
13		yes	no	yes	yes	yes	no	5	4	3	1
14		no	no	yes	yes	yes	yes	4	5	2	1
15		no	no	yes	yes	yes	no	4	4	4	1
16		yes	yes	yes	yes	yes	no	3	2	3	1
17		no	yes	yes	yes	no	no	5	3	2	1
18		no	yes	yes	yes	yes	no	5	5	5	2
19		yes	yes	yes	yes	yes	no	3	1	3	1

```

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

```

```

# In[3]:

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

Out[14]:
   age  Medu  Fedu  ...  internet_yes  romantic_no
0    18     4     4  ...              0             1
1    17     1     1  ...              1             1
2    15     1     1  ...              1             1
3    15     4     2  ...              1             0
4    16     3     3  ...              0             1

[5 rows x 57 columns]

```

```

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

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

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

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

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

```

```
# shuffle rows
apel = apel.sample(frac=1)
# split training and testing data
apel_train = apel[:500]
apel_test = apel[500:]

apel_train_att = apel_train.drop(['pass'], axis=1)
apel_train_pass = apel_train['pass']

apel_test_att = apel_test.drop(['pass'], axis=1)
apel_test_pass = apel_test['pass']

apel_att = apel.drop(['pass'], axis=1)
apel_pass = apel['pass']

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

```
In [15]: apel = apel.sample(frac=1)
...: # split training and testing data
...: apel_train = apel[:500]
...: apel_test = apel[500:]
...:
...: apel_train_att = apel_train.drop(['pass'], axis=1)
...: apel_train_pass = apel_train['pass']
...:
...: apel_test_att = apel_test.drop(['pass'], axis=1)
...: apel_test_pass = apel_test['pass']
...:
...: apel_att = apel.drop(['pass'], axis=1)
...: apel_pass = apel['pass']
...:
...: # number of passing students in whole dataset:
...: import numpy as np
...: print("Passing: %d out of %d (%.2f%%)" %
(np.sum(apel_pass), len(apel_pass), 100*float(np.sum(apel_pass)) /
len(apel_pass)))
Passing: 166 out of 395 (42.03%)
```

5. # fit a decision tree

```
from sklearn import tree
```

```
t = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
```

```
t = t.fit(d_train_att, d_train_pass)
```

```
# In[5]:
# fit a decision tree
from sklearn import tree
nanas = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
nanas = nanas.fit(apel_train_att, apel_train_pass)
```

```
In [16]: from sklearn import tree
...: nanas = tree.DecisionTreeClassifier(criterion="entropy",
max_depth=5)
...: nanas = nanas.fit(apel_train_att, apel_train_pass)

In [17]:
```

```
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
```

```

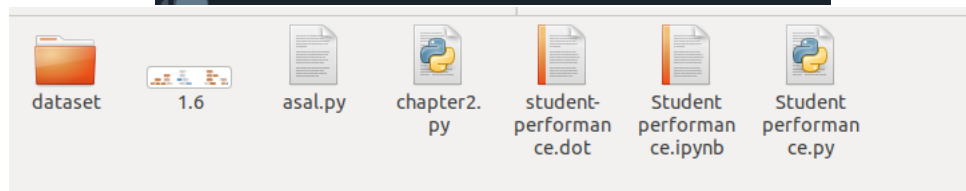
graph TD
    Root["failures <= 0.5  
samples = 309.0%  
value = [0.59, 0.42]  
class = fail"]
    Root -- True --> Node1["subscore no <= 0.5  
samples = 79.0%  
value = [0.519, 0.484]  
class = fail"]
    Root -- False --> Node2["failures <= 2.5  
samples = 21.0%  
value = [0.807, 0.193]  
class = fail"]
    
    Node1 -- True --> Node3["familiar <= 1.5  
samples = 28.0%  
value = [0.474, 0.526]  
class = pass"]
    Node1 -- False --> Node4["mob services <= 0.5  
samples = 0.0%  
value = [0.625, 0.375]  
class = fail"]
    
    Node2 -- True --> Node5["failures <= 3.5  
samples = 17.0%  
value = [0.703, 0.297]  
class = fail"]
    Node2 -- False --> Node6["failures <= 1.5  
samples = 0.0%  
value = [0.882, 0.118]  
class = fail"]
    
    Node3 -- True --> Node7["mob services <= 0.5  
samples = 0.0%  
value = [0.485, 0.515]  
class = pass"]
    Node3 -- False --> Node8["mob health <= 0.5  
samples = 62.7%  
value = [0.529, 0.471]  
class = fail"]
    
    Node4 -- True --> Node9["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node4 -- False --> Node10["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node5 -- True --> Node11["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    Node5 -- False --> Node12["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    
    Node6 -- True --> Node13["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    Node6 -- False --> Node14["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    
    Node7 -- True --> Node15["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    Node7 -- False --> Node16["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    
    Node8 -- True --> Node17["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node8 -- False --> Node18["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node9 -- True --> Node19["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node9 -- False --> Node20["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node10 -- True --> Node21["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node10 -- False --> Node22["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node11 -- True --> Node23["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    Node11 -- False --> Node24["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    
    Node12 -- True --> Node25["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    Node12 -- False --> Node26["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    
    Node13 -- True --> Node27["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    Node13 -- False --> Node28["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    
    Node14 -- True --> Node29["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    Node14 -- False --> Node30["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    
    Node15 -- True --> Node31["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    Node15 -- False --> Node32["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    
    Node16 -- True --> Node33["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    Node16 -- False --> Node34["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    
    Node17 -- True --> Node35["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node17 -- False --> Node36["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node18 -- True --> Node37["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node18 -- False --> Node38["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node19 -- True --> Node39["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node19 -- False --> Node40["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node20 -- True --> Node41["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node20 -- False --> Node42["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node21 -- True --> Node43["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node21 -- False --> Node44["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node22 -- True --> Node45["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node22 -- False --> Node46["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    
    Node23 -- True --> Node47["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    Node23 -- False --> Node48["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    
    Node24 -- True --> Node49["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    Node24 -- False --> Node50["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    
    Node25 -- True --> Node51["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    Node25 -- False --> Node52["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    
    Node26 -- True --> Node53["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    Node26 -- False --> Node54["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    
    Node27 -- True --> Node55["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    Node27 -- False --> Node56["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    
    Node28 -- True --> Node57["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    Node28 -- False --> Node58["absences <= 4.0  
samples = 1.0%  
value = [0.5, 0.5]  
class = fail"]
    
    Node29 -- True --> Node59["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    Node29 -- False --> Node60["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    
    Node30 -- True --> Node61["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    Node30 -- False --> Node62["job teacher <= 6.5  
samples = 7.7%  
value = [0.847, 0.153]  
class = fail"]
    
    Node31 -- True --> Node63["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    Node31 -- False --> Node64["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    
    Node32 -- True --> Node65["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    Node32 -- False --> Node66["gout <= 1.5  
samples = 14.7%  
value = [0.526, 0.474]  
class = pass"]
    
    Node33 -- True --> Node67["familiar <= 0.5  
samples = 2.8%  
value = [0.645, 0.355]  
class = fail"]
    Node33 -- False -->
```

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)
```

```
# In[7]:  
  
# save tree  
tree.export_graphviz(nanas, out_file="student-performance.dot", label="all", impurity=False, proportion=True,  
    feature_names=list(apel_train_att), class_names=["fail", "pass"],  
    filled=True, rounded=True)
```

```
In [20]: tree.export_graphviz(nanas, out_file="student-  
performance.dot", label="all", impurity=False, proportion=True,  
...:    feature_names=list(apel_train_att),  
class_names=["fail", "pass"],  
...:    filled=True, rounded=True)  
  
In [21]:
```



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

```
# In[8]:  
  
nanas.score(apel_test_att, apel_test_pass)
```

```
In [30]:  
...: tree.export_graphviz(nanas,  
out_file="student-performance.dot", label="all",  
impurity=False, proportion=True,  
...:  
feature_names=list(apel_train_att),  
class_names=["fail", "pass"],  
...:             filled=True,  
rounded=True)  
  
In [31]: nanas.score(apel_test_att, apel_test_pass)  
Out[31]: 0.6040268456375839
```



```

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))

```

```

# In[9]:
from sklearn.model_selection import cross_val_score
jeruk = cross_val_score(nanas, apel_att, apel_pass, cv=5)
# show average score and +/- two standard deviations away (covering 95% of scores)
print("Accuracy: %0.2f (+/- %0.2f)" % (jeruk.mean(), jeruk.std() * 2))

In [41]: from sklearn.model_selection import
cross_val_score
...: jeruk = cross_val_score(nanas, apel_att,
apel_pass, cv=5)
...: # show average score and +/- two standard
deviations away (covering 95% of scores)
...: print("Accuracy: %0.2f (+/- %0.2f)" %
(jeruk.mean(), jeruk.std() * 2))

Output from spyder call 'get_cwd':
Accuracy: 0.54 (+/- 0.05)

```

```

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)
    )

```

```

# In[10]:
for max_depth in range(1, 20):
    nanas = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
    jeruk = cross_val_score(nanas, apel_att, apel_pass, cv=5)
    print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" % (max_depth, jeruk.mean(), jeruk.std() * 2))

```

Output from spyder call 'get_cwd':

```

Max depth: 1, Accuracy: 0.58 (+/- 0.01)
Max depth: 2, Accuracy: 0.60 (+/- 0.05)
Max depth: 3, Accuracy: 0.55 (+/- 0.10)
Max depth: 4, Accuracy: 0.54 (+/- 0.09)
Max depth: 5, Accuracy: 0.53 (+/- 0.04)
Max depth: 6, Accuracy: 0.55 (+/- 0.07)
Max depth: 7, Accuracy: 0.52 (+/- 0.12)
Max depth: 8, Accuracy: 0.55 (+/- 0.12)
Max depth: 9, Accuracy: 0.55 (+/- 0.05)
Max depth: 10, Accuracy: 0.54 (+/- 0.06)
Max depth: 11, Accuracy: 0.52 (+/- 0.07)
Max depth: 12, Accuracy: 0.56 (+/- 0.08)
Max depth: 13, Accuracy: 0.55 (+/- 0.08)
Max depth: 14, Accuracy: 0.54 (+/- 0.04)
Max depth: 15, Accuracy: 0.53 (+/- 0.08)
Max depth: 16, Accuracy: 0.56 (+/- 0.03)
Max depth: 17, Accuracy: 0.55 (+/- 0.03)
Max depth: 18, Accuracy: 0.56 (+/- 0.03)
Max depth: 19, Accuracy: 0.53 (+/- 0.05)

```

```

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

```

depth_acc = np.empty((19,3), float)#membuat array baru
i = 0#membuat variable berisi 0
for max_depth in range(1, 20):#perulangan untuk masukkan data
    nanas = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)#membuat decis
    jeruk = cross_val_score(nanas, apel_att, apel_pass, cv=5)#mendefinisikan nagoya yang isinya
    depth_acc[i,0] = max_depth#menasukkan data max_depth ke array depth_acc
    depth_acc[i,1] = jeruk.mean()#menasukkan data rata rata dari jeruk ke array depth_acc
    depth_acc[i,2] = jeruk.std() * 2#menasukkan data akar 2 dari jeruk ke array depth_acc
    i += 1

depth_acc

```

```

Out[45]:
array([[1.00000000e+00, 5.79746835e-01, 1.01265823e-02],
       [2.00000000e+00, 6.00000000e-01, 5.21297729e-02],
       [3.00000000e+00, 5.54430380e-01, 1.00503459e-01],
       [4.00000000e+00, 5.44303797e-01, 8.31983632e-02],
       [5.00000000e+00, 5.26582278e-01, 3.43409113e-02],
       [6.00000000e+00, 5.56962025e-01, 5.77304013e-02],
       [7.00000000e+00, 5.18987342e-01, 1.07408625e-01],
       [8.00000000e+00, 5.31645570e-01, 1.08595497e-01],
       [9.00000000e+00, 5.39240506e-01, 5.90476141e-02],
       [1.00000000e+01, 5.67088608e-01, 8.82814976e-02],
       [1.10000000e+01, 5.51898734e-01, 4.11343717e-02],
       [1.20000000e+01, 5.67088608e-01, 5.63824239e-02],
       [1.30000000e+01, 5.49367089e-01, 4.96099188e-02],
       [1.40000000e+01, 5.44303797e-01, 6.60172395e-02],
       [1.50000000e+01, 5.44303797e-01, 6.20123986e-02],
       [1.60000000e+01, 5.46835443e-01, 1.89451007e-02],
       [1.70000000e+01, 5.62025316e-01, 4.69550304e-02],
       [1.80000000e+01, 5.54430380e-01, 6.48417644e-02],
       [1.90000000e+01, 5.46835443e-01, 5.16356406e-02]])

```

```
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#mengimport fungsi plot dari library notplotlib sebagai plt
fig, ax = plt.subplots()#membuat plot baru
ax.errorbar(depth_acc[:,0], depth_acc[:,1], yerr=depth_acc[:,2])#mengisikan data plot
plt.show()# menampilkan plot
```

```
In [49]: import matplotlib.pyplot as plt#mengimport
fungsi plot dari library notplotlib sebagai plt
...: fig, ax = plt.subplots()#membuat plot baru
...: ax.errorbar(depth_acc[:,0], depth_acc[:,1],
yerr=depth_acc[:,2])#mengisikan data plot
...: plt.show()# menampilkan plot

In [50]:
```

2.3 Penanganan Error

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

1. skrinshot error

```
100*float(np.sum(apel_pass)) / len(apel_pass)))
Passing: 166 out of 395 (42.03%)

In [16]: from sklearn import tree
...: nanas = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: nanas = nanas.fit(apel_train_att, apel_train_pass)

In [17]: import graphviz
...: dot_data = tree.export_graphviz(nanas, out_file=None, label="all",
...: in_purity=False, proportion=True,
...: feature_names=list(apel_train_att),
...: class_names=["fail", "pass"],
...: filled=True, rounded=True)
...: graph = graphviz.Source(dot_data)
...: graph
Traceback (most recent call last):

  Input In [17] in <cell line: 1>
    import graphviz
ModuleNotFoundError: No module named 'graphviz'
```

```
return func(*args, **kwargs)

File ~/anaconda3/envs/spyder/lib/python3.9/site-packages/graphviz/piping.py:114 in
_pipe_legacy
    return self._pipe_future(format,

File ~/anaconda3/envs/spyder/lib/python3.9/site-packages/graphviz/piping.py:139 in
_pipe_future
    return self._pipe_lines_string(*args, encoding=encoding, **kwargs)

File ~/anaconda3/envs/spyder/lib/python3.9/site-packages/graphviz/backend/piping.py:196
in pipe_lines_string
    proc = execute.run_check(cmd, capture_output=True, quiet=quiet, **kwargs)

File ~/anaconda3/envs/spyder/lib/python3.9/site-packages/graphviz/backend/execute.py:88
in run_check
    raise ExecutableNotFound(cmd) from e

ExecutableNotFound: failed to execute PosixPath('dot'), make sure the Graphviz executables
are on your systems' PATH

Out[18]: <graphviz.sources.Source at 0x7f31174dd910>
```

2. Tuliskan kode eror dan jenis errornya

ModulNotFoundError: no modul named 'graphviz'

ExecutableNotFound: failed to execute posixPath('dot'), make sure the Graphviz executables are in your system PATH

Jenis erornya adalah tidak adanya modul didalamnya dan masih kurang ketika di instal modulnya

Solusinya dengan melakukan instal modul.

3. Solusi pemecahan masalah error tersebut

```
Activities Terminal
Jum 14:58
yunita@yunita: ~

File Edit View Search Terminal Tabs Help

yunita@yunita: ~
The following packages have unmet dependencies:
libgraphviz-dev : Depends: zlib1g-dev but it is not going to be installed
E: Unable to correct problems, you have held broken packages.
(spyder) yunita@yunita:~$ pip install graphviz
Collecting graphviz
  Downloading graphviz-0.19.1-py3-none-any.whl (46 kB)
    | 46 kB 1.4 MB/s
Installing collected packages: graphviz
Successfully installed graphviz-0.19.1
(spyder) yunita@yunita:~$ sudo apt-get install graphviz
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcdt5 libcgraph6 libgvc6 libgvpr2 libpathplan4
Suggested packages:
  graphviz-doc
The following NEW packages will be installed:
  graphviz libcdt5 libcgraph6 libgvc6 libgvpr2 libpathplan4
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 1.533 kB of archives.
After this operation, 12,4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://cz.archive.ubuntu.com/ubuntu xenial/main amd64 libcdt5 amd64 2.38.0-12ubuntu2 [23,4 kB]
Get:2 http://cz.archive.ubuntu.com/ubuntu xenial/main amd64 libcgraph6 amd64 2.38.0-12ubuntu2 [43,6 kB]
Get:3 http://cz.archive.ubuntu.com/ubuntu xenial/main amd64 libpathplan4 amd64 2.38.0-12ubuntu2 [26,5 kB]
Get:4 http://cz.archive.ubuntu.com/ubuntu xenial/main amd64 libgvc6 amd64 2.38.0-12ubuntu2 [592 kB]
Get:5 http://cz.archive.ubuntu.com/ubuntu xenial/main amd64 libgvpr2 amd64 2.38.0-12ubuntu2 [170 kB]
Get:6 http://cz.archive.ubuntu.com/ubuntu xenial/main amd64 graphviz amd64 2.38.0-12ubuntu2 [677 kB]
Fetched 1.533 kB in 9s (168 kB/s)
Selecting previously unselected package libcdt5.
(Reading database ... 258276 files and directories currently installed.)
Preparing to unpack .../0-libcdt5_2.38.0-12ubuntu2_amd64.deb ...
Unpacking libcdt5 (2.38.0-12ubuntu2) ...
Selecting previously unselected package libcgraph6.
Preparing to unpack .../1-libcgraph6_2.38.0-12ubuntu2_amd64.deb ...
Unpacking libcgraph6 (2.38.0-12ubuntu2) ...
Preparing to unpack .../4-libgvpr2_2.38.0-12ubuntu2_amd64.deb ...
Unpacking libgvpr2 (2.38.0-12ubuntu2) ...
Selecting previously unselected package graphviz.
Preparing to unpack .../5-graphviz_2.38.0-12ubuntu2_amd64.deb ...
Unpacking graphviz (2.38.0-12ubuntu2) ...
Setting up libpathplan4 (2.38.0-12ubuntu2) ...
Setting up libcdt5 (2.38.0-12ubuntu2) ...
Setting up libcgraph6 (2.38.0-12ubuntu2) ...
Setting up libgvc6 (2.38.0-12ubuntu2) ...
Setting up libgvpr2 (2.38.0-12ubuntu2) ...
Setting up graphviz (2.38.0-12ubuntu2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
(spyder) yunita@yunita:~$ pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.5.1-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.whl (11.2 MB)
    | 11.2 MB 2.4 MB/s
Requirement already satisfied: numpy>=1.17 in ./anaconda3/envs/spyder/lib/python3.9/site-packages (from matplotlib) (1.22.3)
Collecting cycler>=0.10
  Using cached cycler-0.11.0-py3-none-any.whl (6.4 kB)
Requirement already satisfied: packaging>=20.0 in ./anaconda3/envs/spyder/lib/python3.9/site-packages (from matplotlib) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in ./anaconda3/envs/spyder/lib/python3.9/site-packages (from matplotlib) (3.0.4)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.4.0-cp39-cp39-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.6 MB)
    | 1.6 MB 2.2 MB/s
Requirement already satisfied: python-dateutil>=2.7 in ./anaconda3/envs/spyder/lib/python3.9/site-packages (from matplotlib) (2.8.2)
Collecting pillow>=6.2.0
  Downloading Pillow-9.0.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.3 MB)
    | 4.3 MB 2.6 MB/s
Collecting fonttools>=4.22.0
  Downloading fonttools-4.31.2-py3-none-any.whl (899 kB)
    | 899 kB 2.3 MB/s
Requirement already satisfied: six>=1.5 in ./anaconda3/envs/spyder/lib/python3.9/site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Installing collected packages: pillow, kiwisolver, fonttools, cycler, matplotlib
Successfully installed cycler-0.11.0 fonttools-4.31.2 kiwisolver-1.4.0 matplotlib-3.5.1 pillow-9.0.1
(spyder) yunita@yunita:~$
```

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 akurasinya dengan perintah listing 3.13

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

Listing 3.13: Score perolehan dari klasifikasi

3.1.2 Confusion Matrix

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

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

Listing 3.14: Membuat Confusion Matrix

Kemudian kita plot dengan perintah

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

```

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

```

Listing 3.15: Plotting Confusion Matrix

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

```

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

```

Listing 3.16: Membaca file classes.txt

Lalu kita plot

```

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

```

Listing 3.17: Plot hasil perubahan label

3.1.3 Mencoba dengan metode Decission Tree dan SVM

Kita coba menggunakan Decission tree

```

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

```

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

Kita coba menggunakan SVM

```

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

```

Listing 3.19: Mencoba klasifikasi dengan SVM dengan dataset yang sama

3.1.4 Pengecekan Cross Validation

Pengecekan Cross Validation untuk random forest

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

Listing 3.20: Hasil Cross Validation random forest

untuk decision tree

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

Listing 3.21: Hasil Cross Validation Decision Tree

untuk SVM

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

Listing 3.22: Hasil Cross Validation SVM

3.1.5 Pengamatan komponen informasi

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

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

Listing 3.23: Melakukan Pengamatan komponen informasi

Dan kita bisa melakukan plot informasi ini dengan kode

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

Listing 3.24: Plot Komponen informasi agar bisa dibaca

3.2 Soal Teori

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

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

3.3 Praktek Program

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

1. buat aplikasi sederhana menggunakan pandas dan jelaskan arti setiap baris kode yang dibuat(harus beda dengan teman sekelas)
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 Fores pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
5. jalankan program confusion matrix pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
6. jalankan program klasifikasi SVM dan Decission Tree pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
7. jalankan program cross validaiton pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.
8. jalankan program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran 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. Langkah pertama kita baca terlebih dahulu dengan perintah 4.1.

```

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

```

Listing 4.1: Membaca data file txt

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

```

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

```

Listing 4.2: Instansiasi Vektorizer

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

```

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

```

Listing 4.3: Vektorisasi data dari atribut CONTENT

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

```

1 daptarkata=vectorizer.get_feature_names()

```

Listing 4.4: Mendapatkan Daftar Kata

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

```

1 dshuf = d.sample(frac=1)

```

Listing 4.5: Mengocok Data Frame

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

```

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

```

Listing 4.6: Memisahkan data frame

Kita lakukan training perintah listing 4.7.

```

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

```

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

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

```

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

```

Listing 4.8: Transform tanpa fit dari data testing

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

```

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

```

Listing 4.9: Pengambilan label dari data testing dan training

4.1.2 Klasifikasi dengan Random Forest

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

```

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

```

Listing 4.10: Instansiasi kelas Random Forest

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

```

1 clf.fit(d_train_att, d_train_label)

```

Listing 4.11: Fitting random forest dengan dataset training

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

```

1 clf.predict(d_test_att)

```

Listing 4.12: Melihat Hasil prediksi

Untuk besaran skornya dengan perintah listing 4.13

```

1 clf.score(d_test_att, d_test_label)

```

Listing 4.13: Score perolehan dari klasifikasi

4.1.3 Confusion Matrix

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

```

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

```

Listing 4.14: Membuat Confusion Matrix

4.1.4 Pengecekan Cross Validation

Pengecekan Cross Validation untuk random forest dengan perintah 4.15.

```

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

```

Listing 4.15: Hasil Cross Validation random forest

4.2 Soal Teori

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

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

4.3 Praktek Program

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

1. buat aplikasi sederhana menggunakan pandas, buat data dummy format csv sebanyak 500 baris dan melakukan load ke dataframe panda.jelaskan arti setiap baris kode yang dibuat(harus beda dengan teman sekelas)
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 Decission 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 Decission 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 menda-
patkan 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 di lakukan 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=1NoV357ZvemI>

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.