

Modul Praktikum Kecerdasan Buatan



Rolly Maulana Awangga
0410118609

Applied Bachelor of Informatics Engineering
Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering
Politeknik Pos Indonesia

Bandung 2019

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

Acknowledgements

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

Abstract

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

Contents

1	Mengenal Kecerdasan Buatan dan Scikit-Learn	1
1.1	Teori	1
1.2	Instalasi	2
1.3	Penanganan Error	2
1.4	Andri Fajar S/1164065	2
1.4.1	TEORI	2
1.4.2	Instalasi	4
1.4.3	Mencoba Learning and predicting	4
1.4.4	Mencoba Model Persistance	5
1.4.5	Mencoba Conventions	9
1.5	Penanganan Error	12
1.6	Yusniar Nur Syarif Sidiq/1164089	13
1.6.1	Teori	13
1.6.2	Instalasi	15
1.6.3	Learning And Predicting	16
1.6.4	Model Persistence	18
1.6.5	Conventions	20
1.6.6	Penanganan Error	21
1.7	Imron Sumadireja / 1164076	22
1.7.1	Teori	22
1.7.2	Instalasi	25
1.7.2.1	Proses Instalasi Anaconda dan Library Scikit	25
1.7.3	Mencoba Loading Dataset	27
1.7.4	Learning and Predicting	27
1.7.5	Model Persistence	29
1.7.6	Conventions	31
1.7.7	Penanganan Error	38

2	Related Works	40
2.1	Same Topics	40
2.1.1	Topic 1	40
2.1.2	Topic 2	40
2.2	Same Method	40
2.2.1	Method 1	40
2.2.2	Method 2	40
2.3	Yusniar Nur Syarif Sidiq/1164089	41
2.3.1	Binary Classification	41
2.3.2	Supervised Learning, Unsupervised Learning, Dan Classtering	42
2.3.3	Evaluasi Dan Akurasi	43
2.3.4	Confusion Matrix	44
2.3.5	Cara Kerja K-Fold Cross Validation	45
2.3.6	Decision Tree	46
2.3.7	Gain Dan Entropi	46
2.4	Scikit Learn	47
2.5	Penangan Errorr	55
2.6	Andri Fajar Sunandhar/1164065	56
2.6.1	binary classification dilengkapi ilustrasi gambar	56
2.6.2	supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar	57
2.6.3	evaluasi dan akurasi dari buku dan disertai ilustrasi contoh den- gan gambar	60
2.6.4	bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix	60
2.6.5	bagaimana K-fold cross validation bekerja dengan gambar ilus- trasi	61
2.6.6	decision tree dengan gambar ilustrasi	61
2.6.7	Information Gain dan entropi dengan gambar ilustrasi	62
2.7	Imron Sumadireja / 1164076	63
2.7.1	Binary Classification	63
2.7.2	Supervised Learning, Unsupervised Learning, dan Classtering	63
2.7.3	Evaluasi dan Akurasi	64
2.7.4	Confusion Matrix	65
2.7.5	Cara kerja K-Fold Cross Validation	66
2.7.6	Decision Tree	67

2.7.7	Information Gain dan Entropi	67
2.8	Imron Sumadireja / 1164076	68
2.8.1	scikit-learn	68
2.8.2	Penanganan Error	73
2.9	scikit-learn	74
2.10	Penanganan Error	80
2.10.1	Error Graphviz	80
3	Methods	90
3.1	The data	90
3.2	Method 1	90
3.3	Method 2	90
3.4	Yusniar Nur Syarif Sidiq/1164089	90
3.5	Imron Sumadireja/1164076	93
3.5.1	Teori	93
3.6	Andri Fajar Sunandhar/1164065	97
3.6.1	Teori	97
4	Experiment and Result	102
4.1	Experiment	102
4.2	Result	102
5	Conclusion	103
5.1	Conclusion of Problems	103
5.2	Conclusion of Method	103
5.3	Conclusion of Experiment	103
5.4	Conclusion of Result	103
6	Discussion	104
7	Discussion	105
8	Discussion	106
9	Discussion	107
10	Discussion	108
11	Discussion	109

12 Discussion	110
13 Discussion	111
14 Discussion	112
A Form Penilaian Jurnal	113
B FAQ	116
Bibliography	118

List of Figures

1.1	conda install scikit-learn.	4
1.2	Melihat Version.	5
1.3	Install pip.	5
1.4	Hasil Kompile.	6
1.5	Hasil Kompile.	6
1.6	Hasil Kompile.	7
1.7	Membuka Python	7
1.8	Estimator Sklearn	8
1.9	Mendefinisikan Classifier	8
1.10	Memanggil Classifier	8
1.11	Memprediksi Nilai Baru	8
1.12	Hasil Classifier	8
1.13	Hasil Classifier	8
1.14	Pickle Python	9
1.15	Classifier Pickle	9
1.16	Joblib	9
1.17	Deklarasi Numpy	10
1.18	Contoh Casting	10
1.19	FitTransform	10
1.20	Regresi Yang Dilempar	11
1.21	Memperbaharui Parameter	11
1.22	MultiClass	12
1.23	MultiClass biner 2D	12
1.24	MultiLabel	13
1.25	Error Import	13
1.26	Instal Library Joblib	14
1.27	Import Library Joblib	14
1.28	conda install scikit-learn.	16

1.29	Melihat Version.	17
1.30	Install pip.	17
1.31	Hasil Kompile.	18
1.32	Dataset.	18
1.33	Install Joblib.	22
1.34	Hasil Joblib.	22
1.35	Download Aplikasi Anaconda	25
1.36	Proses Instalasi Aplikasi	26
1.37	Proses Instalasi Aplikasi	27
1.38	Proses Instalasi Aplikasi	28
1.39	Proses Instalasi Aplikasi	29
1.40	Proses Instalasi Aplikasi	30
1.41	Proses Instalasi Aplikasi	31
1.42	Proses Instalasi Aplikasi	32
1.43	Proses Instalasi Aplikasi	33
1.44	Proses Instalasi Aplikasi	34
1.45	Instalasi Library	34
1.46	Instalasi Library	35
1.47	Instalasi Library	35
1.48	Instalasi Library	36
1.49	Loading dataset	36
1.50	Error	38
1.51	Instalasi	39
1.52	Hasil	39
2.1	Binary Classification.	41
2.2	Supervised Learning.	42
2.3	Unsupervised Learning.	43
2.4	Clasterring.	44
2.5	Evaluasi Dan Akurasi.	45
2.6	K-Fold Cross Validation.	46
2.7	Decision Tree.	47
2.8	Gain Dan Entropi.	48
2.9	Output No 1.	48
2.10	Output No 2.	49
2.11	Output No 3.	49

2.12 Output No 4.	50
2.13 Output No 5.	51
2.14 Output No 6.	52
2.15 Output No 7.	52
2.16 Output No 8.	52
2.17 Output No 9.	53
2.18 Output No 10.	54
2.19 Output No 11.	55
2.20 Output No 12.	56
2.21 Graphviz Errorr.	57
2.22 Graphviz Install 1.	57
2.23 Graphviz Install 2.	58
2.24 Binary Classification	58
2.25 Supervised Learning	58
2.26 Unsupervised Learning	59
2.27 Cluster	60
2.28 Evaluasi dan Akurasi	60
2.29 K-fold cross validation	62
2.30 Decision Tree	62
2.31 Information gain	63
2.32 Binary Classification.	64
2.33 Supervised Learning.	65
2.34 Unsupervised Learning.	66
2.35 Clustering.	67
2.36 Evaluasi dan Akurasi.	68
2.37 K-Fold Cross Validation.	69
2.38 Decision Tree.	70
2.39 Information Gain dan Entropi.	71
2.40 Source Code.	71
2.41 Source Code.	72
2.42 Source Code.	72
2.43 Source Code.	73
2.44 Source Code.	73
2.45 Source Code.	74
2.46 Source Code.	74
2.47 Source Code.	75

2.48 Source Code.	75
2.49 Source Code.	76
2.50 Source Code.	76
2.51 Source Code.	77
2.52 Source Code.	77
2.53 Source Code.	78
2.54 Source Code.	78
2.55 Source Code.	79
2.56 Source Code.	79
2.57 Source Code.	80
2.58 Source Code.	80
2.59 Source Code.	81
2.60 Source Code.	82
2.61 Source Code.	82
2.62 Source Code.	83
2.63 Source Code.	83
2.64 Error.	84
2.65 Resolve.	84
2.66 Error.	84
2.67 Resolve.	84
2.68 Resolve.	85
2.69 Loading Dataset	85
2.70 Generate Binary Label	85
2.71 One-hot Encoding	86
2.72 Shuffle Rows	86
2.73 Fit Decision Tree	86
2.74 Fit Decision Tree	86
2.75 Fit Decision Tree	87
2.76 Score	87
2.77 Cross Val Score	87
2.78 Max Depth	87
2.79 Depth in Range	87
2.80 Matplotlib	88
2.81 Error Graphviz	88
2.82 install Graphviz	88
2.83 Solving Environment	88

2.84	Evaluasi Eror	89
3.1	Random Forest.	91
3.2	Confusion Matrix.	92
3.3	Voting.	93
3.4	Random Forest.	94
3.5	Confusion Matrix.	96
3.6	Voting.	96
3.7	Random Forest.	97
3.8	Kode membaca file.csv	98
3.9	Window Console	98
3.10	Variable Explorer	98
3.11	Dataset Cell	99
3.12	Pohon Keputusan	100
3.13	Data Testing	100
3.14	Voting.	101
A.1	Form nilai bagian 1.	114
A.2	form nilai bagian 2.	115

Chapter 1

Mengenal Kecerdasan Buatan dan Scikit-Learn

Buku umum yang digunakan adalah [4] dan untuk sebelum UTS menggunakan buku *Python Artificial Intelligence Projects for Beginners*[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. Buat Resume Definisi, Sejarah dan perkembangan Kecerdasan Buatan, dengan bahasa yang mudah dipahami dan dimengerti. Buatan sendiri bebas plagiat[hari ke 1](10)
2. Buat Resume mengenai definisi supervised learning, klasifikasi, regresi dan unsupervised learning. Data set, training set dan testing set.[hari ke 1](10)

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)
2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 1](10)
3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)
4. mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)
5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)

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)

1.4 Andri Fajar S/1164065

1.4.1 TEORI

1. Definisi, Sejarah, Dan Perkembangan Sejarah AI

Didefinisikan kecerdasan yang ditunjukkan oleh suatu entitas buatan. Umumnya dianggap komputer. Kecerdasan Buatan (Artificial Intelligence atau AI) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan dimasukkan ke

dalam mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Kecerdasan Buatan (Artificial Intelligence atau AI) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya di anggap komputer. Kecerdasan diciptakan dan dimasukkan melakukan pekerjaan seperti yang dapat dilakukan manusia.

Sejarah dan perkembangan kecerdasan buatan terjadi pada musim panas tahun 1956 tercatat adanya seminar mengenai AI di Darmouth College. Seminar pada waktu itu dihadiri oleh sejumlah pakar komputer dan membahas potensi komputer dalam meniru kepandaian manusia. Akan tetapi perkembangan yang sering terjadi semenjak diciptakannya LISP, yaitu bahasa kecerdasan buatan yang dibuat tahun 1960 oleh John McCarthy. Istilah pada kecerdasan buatan atau Artificial Intelligence diambil dari Marvin Minsky dari MIT. Dia menulis karya ilmiah berjudul Step towards Artificial Intelligence, The Institute of radio Engineers Proceedings 49, January 1961[?].

Supervised learning merupakan sebuah pendekatan dimana sudah terdapat data yang dilatih, dan terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengelompokkan suatu data ke data yang sudah ada. Sedangkan unsupervised learning tidak memiliki data latih, sehingga dari data yang ada, kita mengelompokkan data tersebut menjadi 2 bagian atau 3 bagian dan seterusnya.

Klasifikasi adalah salah satu topik utama dalam data mining atau machine learning. Klasifikasi yaitu suatu pengelompokan data dimana data yang digunakan tersebut mempunyai kelas label atau target.

Regresi adalah Supervised learning tidak hanya mempelajari classifier, tetapi juga mempelajari fungsi yang dapat memprediksi suatu nilai numerik. Contoh, ketika diberi foto seseorang, kita ingin memprediksi umur, tinggi, dan berat orang yang ada pada foto tersebut.

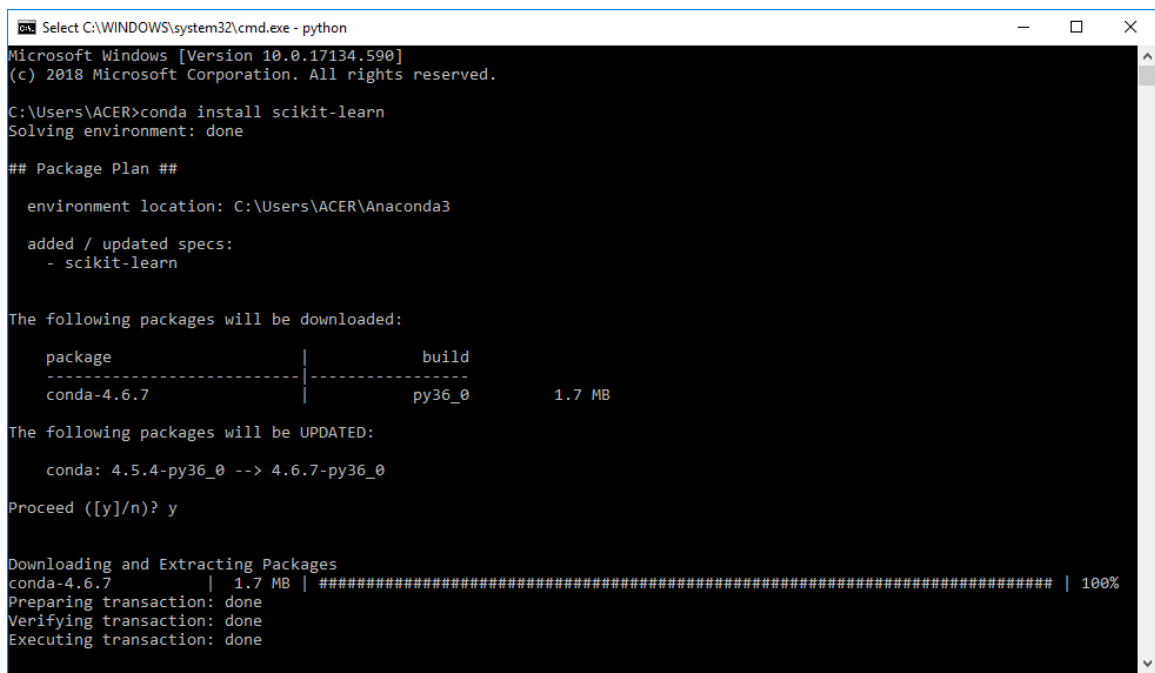
Data set adalah cabang aplikasi dari Artificial Intelligence/Kecerdasan Buatan yang fokus pada pengembangan sebuah sistem yang mampu belajar sendiri tanpa harus berulang kali di program oleh manusia.

2. Training set yaitu jika pasangan objek, dan kelas yang menunjuk pada objek tersebut adalah suatu contoh yang telah diberi label akan menghasilkan suatu algoritma pembelajaran.

Testing set digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar.

1.4.2 Instalasi

- Memberikan perintah conda install scikit-learn di cmd, lihat gambar 1.1
- Melihat versinya dengan memberikan perintah conda -version dan python -version, lihat gambar 1.2
- Install pip, lihat pada gambar 1.3
- Hasil Kompile, lihat gambar 1.4
- Import dataset kemudian load iris dan data dari digits, lihat gambar 1.5
- Melihat data digits



```
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ACER>conda install scikit-learn
Solving environment: done

## Package Plan ##

  environment location: C:\Users\ACER\Anaconda3

added / updated specs:
- scikit-learn

The following packages will be downloaded:

package | build | size
-----|-----|-----
conda-4.6.7 | py36_0 | 1.7 MB

The following packages will be UPDATED:

conda: 4.5.4-py36_0 --> 4.6.7-py36_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.6.7 | 1.7 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Figure 1.1: conda install scikit-learn.

1.4.3 Mencoba Learning and predicting

1. Buka CMD lalu ketikkan perintah Python.
2. "from sklearn import svm" artinya akan memanggil dan menggunakan estimator dari kelas sklearn.svm.SVC

```
C:\Users\ACER>conda --version
conda 4.6.7

C:\Users\ACER>python --version
Python 3.6.5
```

Figure 1.2: Melihat Version.

```
C:\Users\ACER>pip install -U scikit-learn
Collecting scikit-learn
  Downloading https://files.pythonhosted.org/packages/ee/c8/c89ebdc0d7dbba6e6fd222daabd257da3c28a967dd7c352d4272b2e1cef6
/scikit_learn-0.20.2-cp36-cp36m-win32.whl (4.3MB)
    100% |#####| 4.3MB 622kB/s
Collecting numpy>=1.8.2 (from scikit-learn)
  Downloading https://files.pythonhosted.org/packages/84/10/f1f99ba67aff4c3fb033571e87876ed0403114b13bc70cc125372b0c1dcb
/numpy-1.16.1-cp36-cp36m-win32.whl (10.0MB)
    100% |#####| 10.0MB 863kB/s
Collecting scipy>=0.13.3 (from scikit-learn)
  Downloading https://files.pythonhosted.org/packages/b7/b6/eedfa8b002ffae365c3f957154a9c29cb91a8e657808749c78758f0f8110
/scipy-1.2.1-cp36-cp36m-win32.whl (26.9MB)
    100% |#####| 27.0MB 89kB/s
Installing collected packages: numpy, scipy, scikit-learn
Successfully installed numpy-1.16.1 scikit-learn-0.20.2 scipy-1.2.1
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Figure 1.3: Install pip.

3. disini gamma didefinisikan secara manual
4. Estimator clf (for classifier) pertama kali dipasang pada model. Ini dilakukan dengan melewati training set ke metode fit. Untuk training set, akan menggunakan semua gambar dari set data yang ada, kecuali untuk gambar terakhir, yang dicadangan untuk prediksi. Pada skrip dibawah memilih training set dengan sintaks Python [: -1], yang menghasilkan array baru yang berisi semua kecuali item terakhir dari digits.data
5. Pada penggalan skrip dibawah, ini menunjukan prediksi nilai baru menggunakan gambar terakhir dari digits.data.

1.4.4 Mencoba Model Persistence

1. "from sklearn import svm" artinya akan mengimport sebuah Support Vector Machine(SVM) yang merupakan algoritma classification yang akan diambil dari Scikit-Learn.

```
C:\Users\ACER>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('andri')
andri
```

Figure 1.4: Hasil Kompile.

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

Figure 1.5: Hasil Kompile.

2. "from sklearn import datasets" artinya akan mengambil package datasets dari Scikit-Learn.
3. ketikan, `clf = svm.SVC(gamma='scale')` berfungsi untuk mendeklarasikan suatu value yang bernama `clf` yang berisi `gamma`.
4. Ketikan, `X, y = iris.data, iris.target`, artinya `X` sebagai data iris, dan `y` merupakan larik target.
5. Ketikan, `clf.fit(X, y)` berfungsi untuk melakukan pengujian classifier. hasilnya seperti ini

Dari gambar diatas dapat dijelaskan bahwa akan mengimport Pickle dari Python. Pickle digunakan untuk serialisasi dan de-serialisasi struktur objek Python. Objek apa pun dengan Python dapat di-Pickle sehingga dapat disimpan di disk. kemudian menyimpan data objek ke file CLF sebelumnya dengan menggunakan function `pickle.dumps(clf)`.

7. Setelah mengetikan fungsi fungsi diatas, selanjutnya ketikan "`clf2 = pickle.loads(s)`" yang artinya `pickle.loads` digunakan untuk memuat data pickle dari string byte. "S" dalam loads mengacu pada fakta bahwa dalam Python 2, data dimuat dari string.

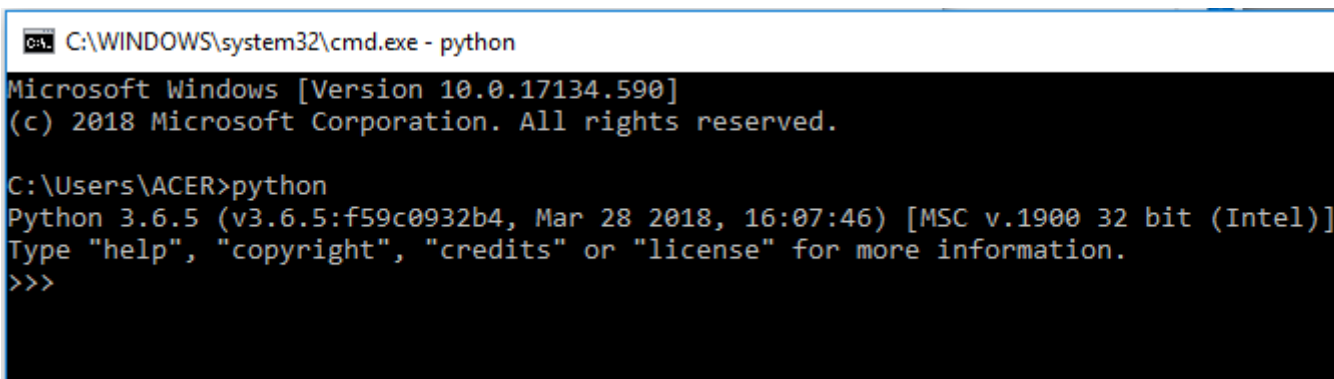
Pada gambar diatas dilakukan pengujian nilai baru dengan menggunakan "`clf2.predict(X[0:1])`" dengan target asumsinya (0,1) hasilnya berbentuk array.

```
>>> 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.]]
>>>
```

Figure 1.6: Hasil Kompile.

9. "from joblib import dump , load" yang artinya akan Merekonstruksi objek Python dari file yang sudah ada.

dump(clf, 'filename.joblib') akan merekontruksi file CLF yang tadi sudah dideklarasikan.
 clf = load('filename.joblib') untuk mereload model yang sudah di Pickle



```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ACER>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)]
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 1.7: Membuka Python

```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ACER>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)]
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import svm
>>>
```

Figure 1.8: Estimator Sklearn

```
>>> clf = svm.SVC(gamma=0.001, C=100.)
>>>
```

Figure 1.9: Mendefinisikan Classifier

```
>>> clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Figure 1.10: Memanggil Classifier

```
>>> clf.predict(digits.data[-1:])
array([8])
>>>
```

Figure 1.11: Memprediksi Nilai Baru

```
>>> clf.fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
>>>
```

Figure 1.12: Hasil Classifier

```
>>> import pickle
>>> s = pickle.dumps(clf)
```

6.

Figure 1.13: Hasil Classifier

```
>>> clf2 = pickle.loads(s)
```

Figure 1.14: Pickle Python

```
>>> clf2.predict(X[0:1])
array([0])
8. >>> y[0]
```

Figure 1.15: Classifier Pickle

1.4.5 Mencoba Conventions

1. Import numpy as np, digunakan untuk mengimport Numpy sebagai np.
From sklearn import randomprojection artinya modul yang mengimplementasikan cara sederhana dan efisien secara komputasi untuk mengurangi dimensi data dengan memperdagangkan sejumlah akurasi yang terkendali (sebagai varian tambahan) untuk waktu pemrosesan yang lebih cepat dan ukuran model yang lebih kecil.

Pada gambar diatas dapat dijelaskan bahwa :

rng = np.random.RandomState(0), digunakan untuk menginisialisasikan random number generator.

X = rng.rand(10, 2000) artinya akan merandom value antara 10 sampai 2000.

X = np.array(X, dtype='float32') Array numpy terdiri dari buffer memori "mentah" yang diartikan sebagai array melalui "views". Anda dapat menganggap semua array numpy sebagai tampilan. Mendeklarasikan X sebagai float32.

3. Dalam contoh ini, X adalah float32, yang dilemparkan ke float64 oleh fittransform (X).
4. Target regresi dilemparkan ke float64 dan target klasifikasi dipertahankan.
list(clf.predict(irisdata[:3])), akan memprediksi 3 data dari iris.
clf.fit irisdata, iristargetnames[iristarget] menguji classifier dengan ada targetnya yaitu irisnya sendiri.

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

Figure 1.16: Joblib

```
>>> import numpy as np
>>> from sklearn import random_projection
```

Figure 1.17: Deklarasi Numpy

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

Figure 1.18: Contoh Casting

`list(clf.predict(irisdata[:3]))`, setelah diuji maka akan muncul datanya seperti dibawah ini

Di sini, prediksi pertama () mengembalikan array integer, karena `iristarget` (array integer) yang digunakan sesuai. Prediksi kedua () mengembalikan array string, karena `iristargetnames` cocok.

5. Refitting dan Memperbaharui Parameter

`y = rngbinomial(1, 0.5, 100)` , random value dengan angka binomial atau suku dua untuk `y`

`clfsetparams(kernel='linear')fit(X, y)` mengubah kernel default menjadi linear
`clfsetparams(kernel='rbf', gamma='scale')fit(X, y)` Di sini, kernel default rbf pertama kali diubah menjadi linear melalui

`SVCsetparams()` setelah estimator dibuat, dan diubah kembali ke rbf untuk mereparasi estimator dan membuat prediksi kedua.

6. MultiClass VS MultiLabel Classifier

`from sklearn.multiclass import OneVsRestClassifier` , adalah ketika kita ingin melakukan klasifikasi multiclass atau multilabel dan baik untuk menggunakan `OneVsRestClassifier` per kelas. Untuk setiap classifier, kelas tersebut dipasang terhadap semua kelas lainnya. (Ini cukup jelas dan itu berarti bahwa masalah klasifikasi multiclass / multilabel dipecah menjadi beberapa masalah klasifikasi biner).

```
>>> transformer = random_projection.GaussianRandomProjection()
>>> X_new = transformer.fit_transform(X)
>>> X_new.dtype
dtype(<'float64'>)
```

Figure 1.19: FitTransform

```

>>> from sklearn import datasets
>>> from sklearn.svm import SVC
>>> iris = datasets.load_iris()
>>> clf = SVC(gamma='scale')
>>> clf.fit(iris.data, iris.target)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
>>> list(clf.predict(iris.data[:3]))
[0, 0, 0]
>>> clf.fit(iris.data, iris.target_names[iris.target])
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
>>> list(clf.predict(iris.data[:3]))
['setosa', 'setosa', 'setosa']

```

Figure 1.20: Regresi Yang Dilempar

```

>>> import numpy as np
>>> from sklearn.svm import SVC
>>> rng = np.random.RandomState(0)
>>> X = rng.rand(100, 10)
>>> y = rng.binomial(1, 0.5, 100)
>>> X_test = rng.rand(5, 10)
>>> clf = SVC()
>>> clf.set_params(kernel='linear').fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])
>>> clf.set_params(kernel='rbf', gamma='scale').fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])

```

Figure 1.21: Memperbaharui Parameter


```

>>> from sklearn.svm import SVC
>>> from sklearn.multiclass import OneVsRestClassifier
>>> from sklearn.preprocessing import LabelBinarizer
>>> X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
>>> y = [0, 0, 1, 1, 2]
>>> classif = OneVsRestClassifier(estimator=SVC(gamma='scale',
... random_state=0))
>>> classif.fit(X, y).predict(X)
array([0, 0, 1, 1, 2])

```

Figure 1.22: MultiClass

```

>>> y = LabelBinarizer().fit_transform(y)
>>> classif.fit(X, y).predict(X)
array([[1, 0, 0],
       [1, 0, 0],
       [0, 1, 0],
       [0, 0, 0],
       [0, 0, 0]])

```

Figure 1.23: MultiClass biner 2D

from sklearn.preprocessing import LabelBinarizer ,adalah kelas utilitas untuk membantu membuat matriks indikator label dari daftar label multi-kelas Dalam gambar dibawah, classifier cocok pada array 1d label multiclass dan oleh karena itu metode predict () memberikan prediksi multiclass yang sesuai.

7. Di sini, classifier cocok () pada representasi label biner 2d dari y, menggunakan LabelBinarizer. Dalam hal ini predict () mengembalikan array 2d yang mewakili prediksi multilabel yang sesuai.
8. from sklearn.preprocessing import MultiLabelBinarizer , artinya Transformasi antara iterable dari iterables dan format multilabel. Dalam hal ini, penggolongnya sesuai pada setiap instance yang diberi beberapa label. MultiLabelBinarizer digunakan untuk membuat binarize array 2d dari multilabel agar sesuai. Hasilnya, predict () mengembalikan array 2d dengan beberapa label yang diprediksi untuk setiap instance.

1.5 Penanganan Error

1. Berikut ini merupakan eror yang ditemui pada saat melakukan percobaan skrip.
2. Pada gambar eror diatas, kode erornya adalah "ImportError: No Module Named" artinya mengalami masalah saat mengimpor modul yang ditentukan.

```

>>> from sklearn.preprocessing import MultiLabelBinarizer
>>> y = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
>>> y = MultiLabelBinarizer().fit_transform(y)
>>> classif.fit(X, y).predict(X)
array([[1, 1, 0, 0, 0],
       [1, 0, 1, 0, 0],
       [0, 1, 0, 1, 0],
       [1, 0, 1, 0, 0],
       [1, 0, 1, 0, 0]])
>>>

```

Figure 1.24: MultiLabel

```

>>> from joblib import dump, load
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named joblib

```

Figure 1.25: Error Import

3. Solusinya bisa dilakukan seperti berikut :
 eror diatas terjadi dikarenakan Library Joblib belum terinstal pada PC. Maka dari itu sekarang kita harus menginstalnya dulu.
4. Buka CMD, kemudian ketikan "pip install joblib" tunggu sampai instalasi berhasil seperti gambar berikut.
5. Apabila sudah terinstall, dapat dilakukan lagi import library joblib, maka akan berhasil seperti dibawah berikut

1.6 Yusniar Nur Syarif Sidiq/1164089

1.6.1 Teori

1. Sejarah Perkembangan Dan Definisi AI

Kecerdasan buatan merupakan sebuah bidang dalam ilmu computer yang begitu penting di zaman ini dan masa yang akan datang guna mewujudkan sebuah sistem computer yang begitu cerdas. Kecerdasan buatan sudah berkembang begitu pesat dalam 20 tahun terakhir seiring dengan adanya kebutuhan perangkat yang cerdas pada bidang industry dan rumah tangga.

Artificial Intelligence atau biasa di singkat dengan AI berasal dari bahasa latin yang dimana intelligence berarti saya paham. AI dimulai dari kemunculan sebuah komputer pada tahun 1940-an, akan tetapi perkembangannya

pat memiliki kemampuan dan juga perilaku seperti manusia.[1].

2. Definisi Supervised Learning Dan Unsupervised Learning

Supervised Learning merupakan sebuah pendekatan yang dimana terdapat data dan variable yang telah ditargetkan sehingga pendekatan tersebut bertujuan untuk mengelompokkan sebuah data ke data yang sudah ada, beda dengan Unsupervised learning yang tidak mempunyai data, sehingga data yang ada harus di kelompokkan menjadi beberapa bagian.

3. Definisi Klasifikasi Dan Regresi

Klasifikasi merupakan sebuah kegiatan penggolongan atau pengelompokkan. Menurut kamus besar bahasa Indonesia yang dimana klasifikasi merupakan penyusunan sistem di dalam kelompok atau golongan berdasarkan kaidah atau standar yang telah ditetapkan. Regresi merupakan sebuah metode analisis statistic yang akan digunakan untuk melihat pengaruh variable.

4. Devinisi Dataset, Training Set, Dan Testing Set

Dataset merupakan sebuah objek yang akan mempresentasikan sebuah data dan relasinya di memory. Struktur pada dataset ini mirip dengan data yang ada di dalam database. Training set merupakan bagian dari dataset yang berperan dalam membuat prediksi atau algoritma sesuai tujuan masing – masing. Testing set merupakan bagian dari dataset yang akan di tes guna melihat keakuratan atau ketepatan datanya.

1.6.2 Instalasi

Untuk melakukan instalasi Anaconda ikuti tutorial berikut.

- Memberikan perintah

```
conda install scikit-learn
```

di cmd, lihat pada figure 1.28

- Melihat versinya dengan memberikan perintah

```
conda --version
```

dan

```
python --version
```

lihat pada figure 1.29

- Install pip, lihat pada figure 1.30
- Hasil Kompile, lihat pada figure 1.31

```
C:\Users\NS>conda install scikit-learn
Solving environment: done

## Package Plan ##

  environment location: C:\Users\NS\Anaconda3

  added / updated specs:
    - scikit-learn

The following packages will be UPDATED:

  conda: 4.5.4-py36_0 --> 4.6.7-py36_0

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Figure 1.28: conda install scikit-learn.

Dataset adalah objek seperti kamus yang menyimpan semua data dan berupa metadata tentang data. Data tersebut disimpan di .data anggota yang merupakan array. Misalnya dalam kasus dataset digit, memberikan akses ke fitur yang dapat digunakan untuk mengklarifikasikan sampel digit, lihat pada figure 1.32.

1.6.3 Learning And Predicting

- Pada codingan di bawah ini yaitu terjadinya pengambilan package datasets dari Scikit-Learn.

```
from sklearn import datasets
```

```
Command Prompt

Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\NS>conda --version
conda 4.5.4

C:\Users\NS>python --version
Python 3.6.5 :: Anaconda, Inc.
```

Figure 1.29: Melihat Version.

```
C:\Users\NS>pip install -U scikit-learn
Collecting scikit-learn
  Using cached https://files.pythonhosted.org/packages/ee/c8/c89ebdc0d7dbba6e6fd222daabd257da3c28a967dd7c352d4272b2e1cef6/scikit_learn-0.20.2-cp36-cp36m-win32.whl
Requirement not upgraded as not directly required: numpy>=1.8.2 in c:\users\ns\anaconda3\lib\site-packages (from scikit-learn) (1.14.3)
Requirement not upgraded as not directly required: scipy>=0.13.3 in c:\users\ns\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: scikit-learn
  Found existing installation: scikit-learn 0.19.1
    Uninstalling scikit-learn-0.19.1:
      Successfully uninstalled scikit-learn-0.19.1
Successfully installed scikit-learn-0.20.2
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Figure 1.30: Install pip.

- Pada codingan di bawah menjelaskan bahwa akan mengambil data iris dari package datasets.

```
iris = datasets.load_iris()
```

- Pada codingan di bawah menjelaskan bahwa akan mengambil data digits dari package datasets.

```
digits = datasets.load_digits()
```

- Pada codingan di bawah menjelaskan bahwa akan melakukan import sebuah Support Vector Machine (SVM) yang merupakan algoritma classification yang akan diambil dari Scikit-Learn.

```
from sklearn import svm
```

- Gamma tersebut akan didefinisikan secara manual, maka codingan akan terlihat seperti berikut

```
clf = svm.SVC(gamma=0.0001, C=100.)
```

```

C:\Users\NS>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Yusniar')
Yusniar
>>>

```

Figure 1.31: Hasil Kompile.

```

>>> 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.]]
>>>

```

Figure 1.32: Dataset.

- Untuk melihat angka prediksi dari data digits maka codingannya seperti berikut

```

clf.fit(digits.data[:-1], digits.target[:-1])
clf.predict(digits.data[-1:])

```

1.6.4 Model Persistence

- Pada codingan di bawah menjelaskan bahwa akan melakukan import sebuah Support Vector Machine (SVM) yang merupakan algoritma classification yang akan diambil dari Scikit-Learn.

```

from sklearn import svm

```

- Pada codingan di bawah menjelaskan bahwa akan melakukan import datasets yang akan diambil dari Scikit-Learn

```
from sklearn import datasets
```

- Codingan dibawah berfungsi untuk mendeklarasikan value pada clf yang berisi gamma

```
clf = svm.SVC(gamma='scale')
```

- Dimana codingan dibawah ini berfungsi untuk mengambil data iris dari datasets

```
iris = datasets.load_iris()
```

- Codingan dibawah menjelaskan bahwa nilai X sebagai data iris dan nilai y sebagai target iris

```
X, y = iris.data, iris.target
```

- Codingan dibawah berfungsi untuk melakukan pengujian classifier

```
clf.fit(X, y)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',  
    max_iter=-1, probability=False, random_state=None, shrinking=True,  
    tol=0.001, verbose=False)
```

- Pada codingan dibawah dijelaskan akan melakukan import pickle. Pickle digunakan sebagai serialisasi dan de-serialisasi struktur objek python. Pickle loads digunakan untuk memuat data pickle dari string byte.

```
import pickle  
s = pickle.dumps(clf)  
clf2 = pickle.loads(s)
```

- Codingan dibawah adalah pengujian dengan nilai baru dan target asumsinya adalah (0,1) dan hasilnya akan berbentuk array.

```
clf2.predict(X[0:1])  
array([0])  
y[0]  
0
```


1.6.5 Conventions

- Melakukan import numpy sebagai np

```
import numpy as np
```

- Melakukan import project secara random dari Sciki-Learn

```
from sklearn import random_projection
```

- Digunakan untuk menginisialisasikan random number generator

```
rng = np.random.RandomState(0)
```

- Melakukan random value antara 10 sampai 2000

```
X = rng.rand(10, 2000)
```

- Mendeklarasikan X sebagai float32

```
X = np.array(X, dtype='float32')
X.dtype
dtype('float32')
```

- Melemparkan nilai X yaitu float32 ke float64 dengan menggunakan transform

```
transformer = random_projection.GaussianRandomProjection()
X_new = transformer.fit_transform(X)
X_new.dtype
dtype('float64')
```

- Melakukan import datasets dari Sciki-Learn

```
from sklearn import datasets
```

- Melakukan import SVC dari sciki-learn yang ada pada svm

```
from sklearn.svm import SVC
```

- Mengambil datasets iris

```
iris = datasets.load_iris()
```

- Mendeklarasikan gamma dengan scale pada SVC dalam variabel clf

```
clf = SVC(gamma='scale')
```

- Membaca data iris dan target iris dari variabel clf

```
clf.fit(iris.data, iris.target)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

- Melakukan pengujian prediksi dengan asumsi (0,3)

```
list(clf.predict(iris.data[:3]))
[0, 0, 0]
```

- Membaca data iris dan nama target iris dari target iris pada variabel clf

```
clf.fit(iris.data, iris.target_names[iris.target])
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

- Melakukan pengujian prediksi dengan asumsi (0,3)

```
list(clf.predict(iris.data[:3]))
['setosa', 'setosa', 'setosa']
```

1.6.6 Penanganan Error

- Dimana Error Tersebut dapat di lihat pada figure ??
- Codingan yang eror tersebut adalah

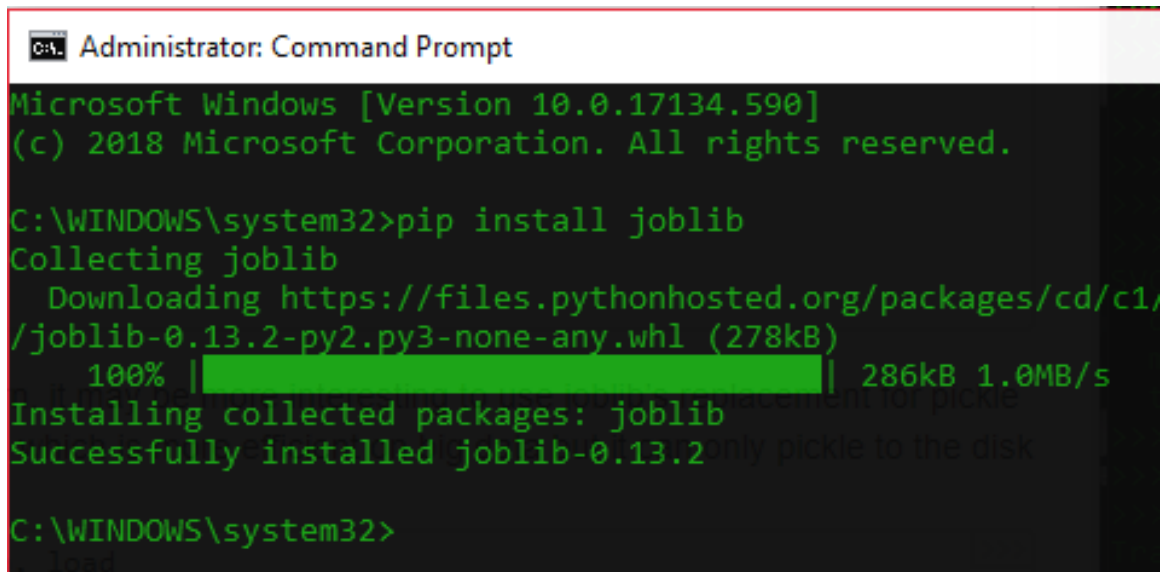
```
from joblib import dump, load
```

- Hal ini dikarenakan belum terinstallnya file joblib di dalam pc anda. Untuk menginstallnya cukup berikan perintah codingan dibawah, untuk lebih jelasnya liat pada figure 1.33

```
pip install joblib
```

- maka hasilnya akan nampak seperti pada figure 1.34

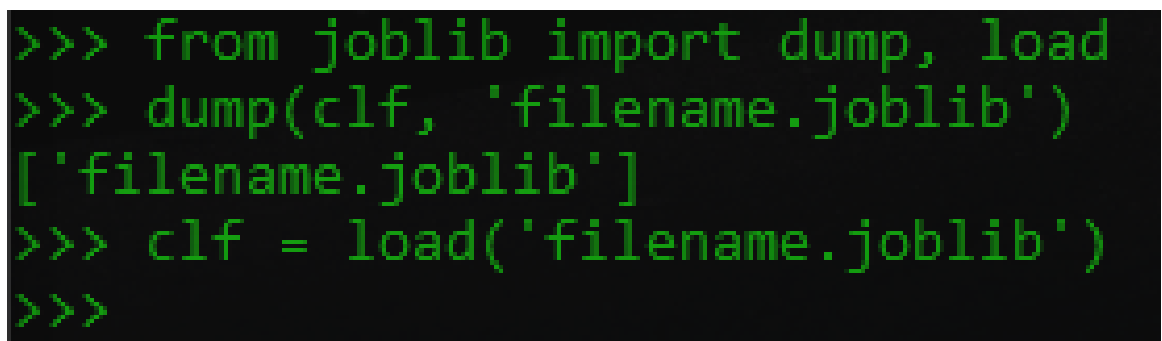
=====



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>pip install joblib
Collecting joblib
  Downloading https://files.pythonhosted.org/packages/cd/c1/
/joblib-0.13.2-py2.py3-none-any.whl (278kB)
    100% |████████████████████████████████████████| 286kB 1.0MB/s
Installing collected packages: joblib
Successfully installed joblib-0.13.2
```

Figure 1.33: Install Joblib.



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

Figure 1.34: Hasil Joblib.

1.7 Imron Sumadireja / 1164076

1.7.1 Teori

1. Pengertian

Kecerdasan Buatan Artificial Intelligence merupakan salah satu bagian dari ilmu komputer yang mempelajari cara membuat mesin komputer dapat melakukan pekerjaan sebaik bahkan lebih baik dari yang dilakukan oleh manusia. Agar mesin dapat bekerja layaknya manusia maka perlu diberi bekal pengetahuan, sehingga mempunyai kemampuan untuk menalar. Menurut para ahli kecerdasan buatan seperti berikut:

- H. A. Simon: Kecerdasan buatan Artificial Intelligence merupakan kawasan

penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

- Rich and Knight: Kecerdasan buatan Artificial Intelligence merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

2. Sejarah dan Perkembangan

Kata intelligence berasal dari bahasa latin intelligo yang memiliki arti saya paham. Arti dasar dari intelligence merupakan kemampuan untuk memahami dan melakukan aksi. Area Kecerdasan Buatan Artificial Intelligence, bermula pada saat kemunculan komputer sekitar tahun 1940-an, walaupun sejarah perkembangannya dapat dilacak sejak zaman Mesir kuno. Pada masa saat ini, perhatian difokuskan pada kemampuan komputer mengerjakan sesuatu yang dapat dilakukan oleh manusia. Dalam hal ini, komputer tersebut dapat meniru kemampuan kecerdasan dan perilaku manusia dengan akurasi yang cukup baik [5].

Pada akhir tahun 1955, Newell dan Simon mengembangkan The Logic Theorist, program AI pertama, program ini merepresentasikan masalah sebagai model pohon, lalu penyelesaiannya dengan memilih cabang yang akan menghasilkan kesimpulan yang paling benar. Pada tahun 1956 John McCarthy dari Massacuhetts Institute of Technology dianggap sebagai bapak AI, menyelenggarakan konferensi untuk menarik para ahli komputer bertemu, dengan nama kegiatan The Dartmouth Summer Research Project on Artificial Intelligence. Konferensi Dartmouth itu mempertemukan para pendiri AI, dan bertugas untuk meletakkan dasar bagi masa depan pengembangan dan penelitian AI. John McCarthy saat itu mengusulkan definisi AI adalah AI merupakan cabang dari ilmu komputer yang berfokus pada pengembangan komputer untuk dapat memiliki kemampuan dan berperilaku seperti manusia[2].

3. Supervised Learning dan Unsupervised Learning

Supervised Learning merupakan suatu pendekatan dimana sudah terdapat data yang dilatih, dan terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengelompokkan suatu data ke data yang sudah ada. Sebagai contoh, ketika Anda memiliki sejumlah buku yang sudah dibeli dengan

beberapa kategori. Misalnya, kategori buku akademik, dan buku novel. Selanjutnya Anda membeli sejumlah buku baru, maka Anda harus mengidentifikasi buku tersebut, dan memasukannya dalam kategori yang sudah ada.

Unsupervised Learning merupakan suatu pendekatan namun tidak memiliki data yang dilatih, sehingga dari data yang ada, kita dapat mengelompokkan data tersebut menjadi 2 bagian atau 3 bagian dan seterusnya. Sebagai contoh, Anda belum pernah membeli sejumlah buku, suatu hari Anda membeli sejumlah buku dan ingin membaginya kedalam beberapa kategori agar mudah dicari. Anda akan mengidentifikasi buku mana yang mirip. Dalam hal ini, kita memilih buku berdasarkan isinya.

4. Klasifikasi dan Regresi

Klasifikasi merupakan penempatan objek-objek ke salah satu dari beberapa kategori yang telah ditentukan sebelumnya. Klasifikasi banyak digunakan untuk memprediksi kelas pada suatu label atau atribut tertentu, yaitu dengan mengklasifikasi data membangun model berdasarkan training set dan nilai-nilai dalam mengklasifikasikan data yang baru. Regresi dibedakan menjadi 2, diantaranya regresi linear dan regresi nonlinear.

- Regresi Linear Regresi Linear merupakan bentuk hubungan di mana variabel bebas x maupun variabel tergantung y sebagai faktor yang berpangkat satu.
- Regresi Nonlinear Regresi Nonlinear merupakan bentuk hubungan atau fungsi di mana variabel x dan variabel tidak bebas y dapat berfungsi sebagai faktor atau variabel dengan pangkat tertentu.

5. Data set, Training set, dan Testing set

Untuk melakukan data set, training set, dan testing set diperlukan beberapa langkah, diantaranya:

- Membuat model atau mesin untuk memeriksa data,
- Membuat model atau mesin belajar dari kesalahannya,
- Membuat kesimpulan tentang seberapa baik kinerja model atau mesin tersebut.

(a) Data set

Data set ini mencakup sekumpulan contoh input yang modelnya akan cocok atau dilatih dengan menyesuaikan parameter.

(b) Training set

Training set diperlukan oleh model atau mesin agar dapat dilatih. Dengan menghitung kerugian tingkat kesalahan yang dilakukan model atau mesin menghasilkan pada set validasi pada titik tertentu, agar kita tahu seberapa akuratnya. Selanjutnya, model akan menyesuaikan parameternya berdasarkan hasil evaluasi yang sering pada training set ini.

(c) Testing set

Testing set sangat penting untuk menguji generalisasi model atau mesin. Dengan testing set ini, kita bisa mendapatkan akurasi kinerja model atau mesin.

1.7.2 Instalasi

1.7.2.1 Proses Instalasi Anaconda dan Library Scikit

1. Pertama kita unduh terlebih dahulu aplikasi anaconda, seperti gambar berikut

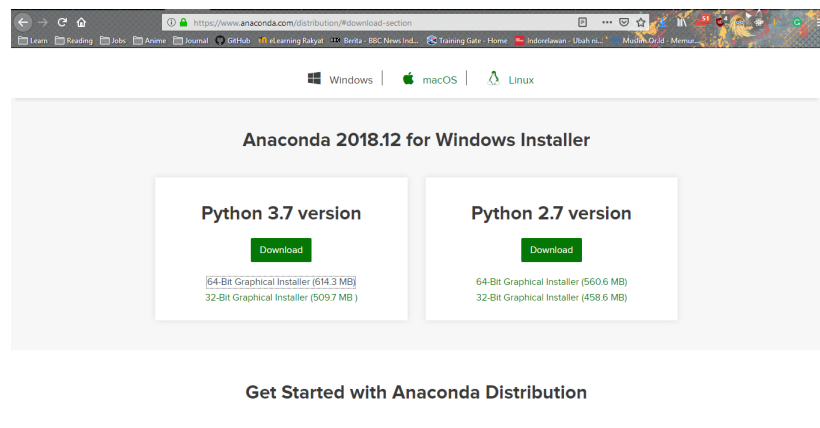


Figure 1.35: Download Aplikasi Anaconda

2. Setelah di unduh, selanjutnya buka aplikasi tersebut. Lalu klik next untuk melanjutkan.
3. Lalu klik I Agree untuk melanjutkan.

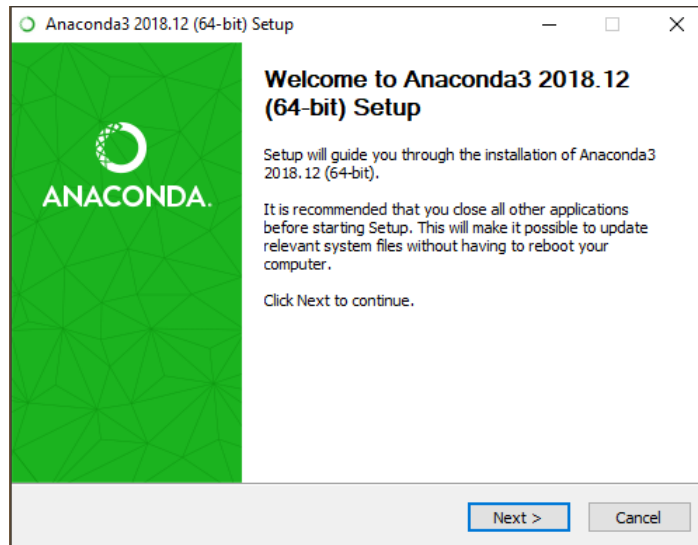


Figure 1.36: Proses Instalasi Aplikasi

4. Selanjutnya pilih Just me agar aplikasi tersebut hanya dapat digunakan oleh user yang login pada laptop tersebut.
5. Lalu tentukan direktori penyimpanan file tersebut
6. Selanjutnya akan muncul pop up box tentang advance installation options, ceklis keduanya.
7. Tunggu hingga proses install selesai
8. Setelah proses instalasi selesai, klik next
9. Pada bagian selanjutnya akan muncul box dengan memberikan pilihan untuk install VS Code, jika tidak klik skip.
10. Setelah selesai, klik finish
11. Setelah proses instalasi selesai, selanjutnya buka cmd dan ketikan seperti berikut.
12. Selanjutnya ketikan perintah berikut untuk mengunduh library scikit
13. Jika sudah berhasil selanjutnya, ketikan perintah seperti gambar berikut untuk melakukan cek versi conda dan python
14. Mencoba dan mengcompile source code, hasilnya seperti berikut

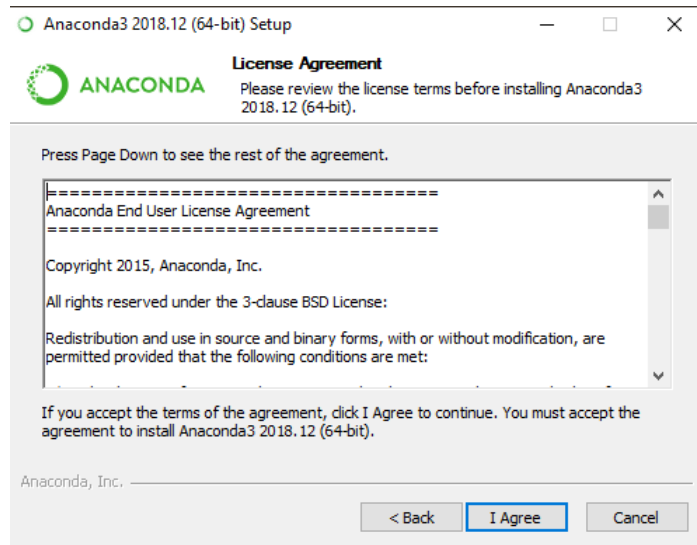


Figure 1.37: Proses Instalasi Aplikasi

1.7.3 Mencoba Loading Dataset

1. Berikut source code yang menjelaskan tentang loading dataset. Pada baris pertama code tersebut berfungsi untuk import library datasets dari sklearn. Baris kedua berfungsi untuk menampilkan data secara berurutan. Baris ketiga untuk menampilkan data tersebut berupa angka dan baris keempat untuk menampilkan data tersebut.

1.7.4 Learning and Predicting

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

- import datasets dari package sklearn
- loading dataset iris
- loading dataset digits
- menampilkan data dari loading dataset digits

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

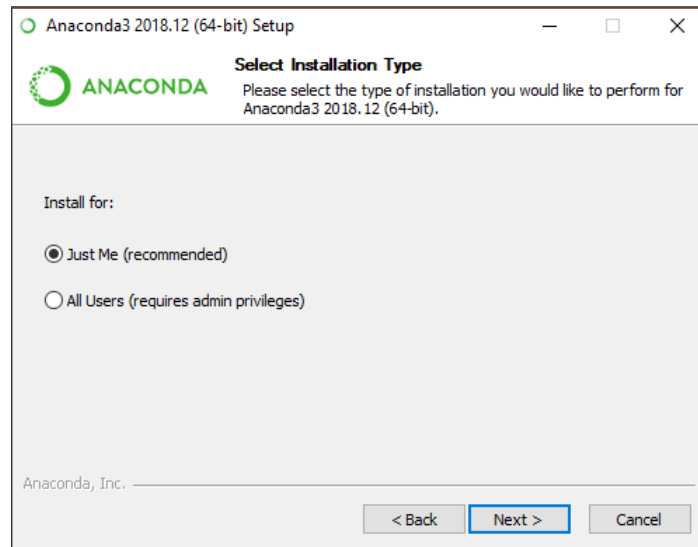



Figure 1.38: Proses Instalasi Aplikasi

- Baris tersebut menjelaskan bahwa dalam project ini kita menggunakan source dari sklearn dengan mengambil/import dari svm
- classifier svc dengan atribur gamma dan c
- classifier tersebut akan dijalankan dengan menggunakan metode fit

```
SVC(C=100.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

- hasilnya seperti di atas

```
>>> clf.predict(digits.data[-1:])
```

- classifier predict loading data digits

```
array([8])
```

- hasilnya seperti di atas

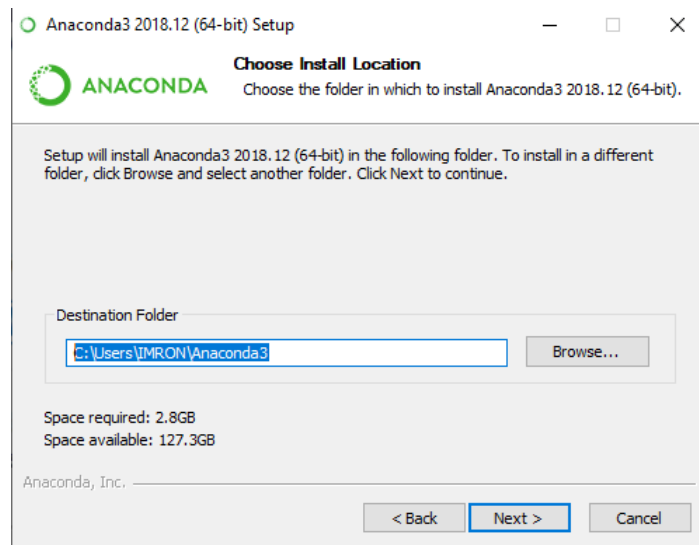


Figure 1.39: Proses Instalasi Aplikasi

1.7.5 Model Persistence

```
>>> from sklearn import svm
>>> from sklearn import datasets
>>> clf = svm.SVC(gamma='scale')
>>> iris = datasets.load_iris()
>>> X, y = iris.data, iris.target
>>> clf.fit(X, y)
```

- import svm dari package sklearn
- importt datasets dari package sklearn
- classifier svc dengan atribut gamma
- loading dataset iris
- parameter x dan y dengan key iris data dan iris target
- classifier akan dijalankan menggunakan metode fit

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

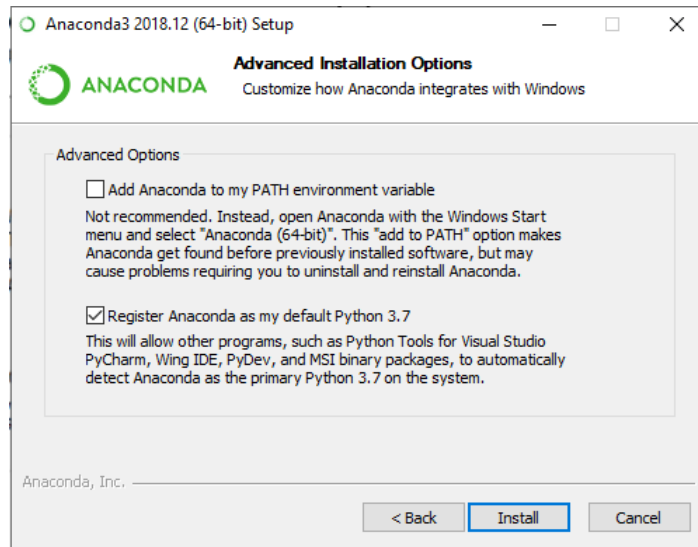


Figure 1.40: Proses Instalasi Aplikasi

- hasilnya seperti di atas

```
>>> import pickle
>>> s = pickle.dumps(clf)
>>> clf2 = pickle.loads(s)
>>> clf2.predict(X[0:1])
```

- import package pickle
- pickle akan melakukan dumps pada classifier
- classifier2 akan mengambil data pada classifier pertama
- classifier2 akan memprediksi hasilnya dengan menggunakan syntax python

```
array([0])
```

- hasilnya seperti diatas

```
>>> y[0]
```

- parameter y dengan atribut 0

```
0
```

- hasil seperti di atas

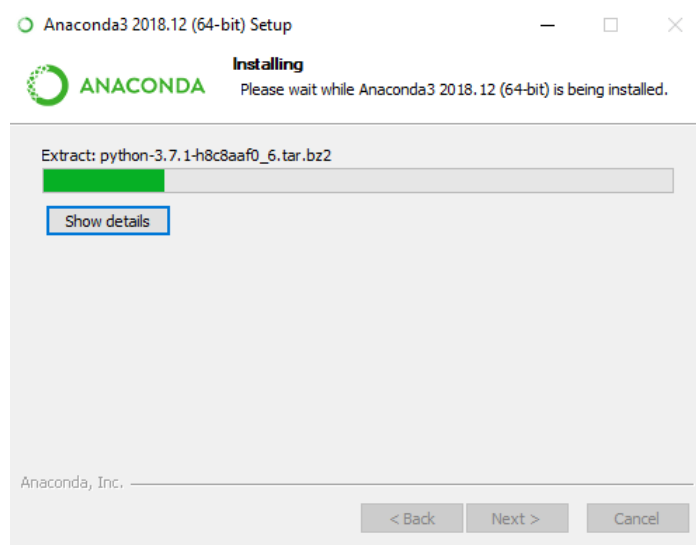


Figure 1.41: Proses Instalasi Aplikasi

1.7.6 Conventions

```
>>> 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
```

- iimport numpy dengan alias np
- import random projection pada package sklearn
- rng parameter dan akan melakukan proses random dalam menentukan hasil
- parameter x memiliki rand dengan nilai 10, 2000
- parameterr x dengan numpy array akan memunculkan kata float32 pada hasil terakhir

```
dtype('float32')
```

- hasilnya seperti diatas

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

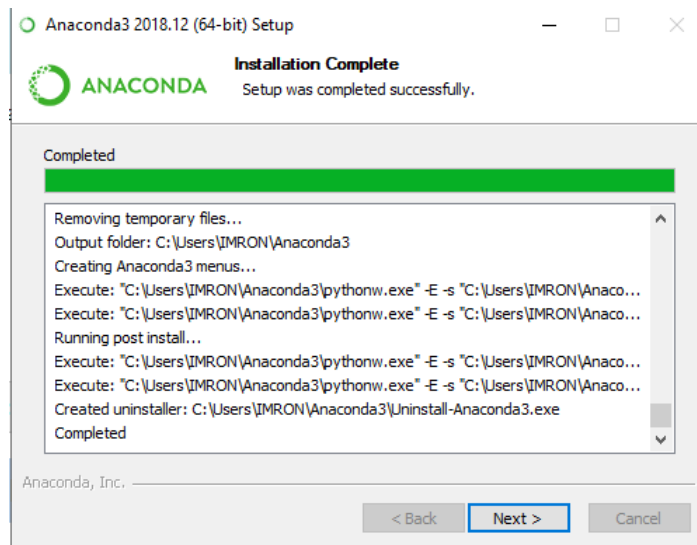


Figure 1.42: Proses Instalasi Aplikasi

- transformer parameter yang di gunakan untuk melakukan pencarian data dengan gaussianrandomprojection
- x new parameter dan akan dijalankan dengan menggunakan metode fit
- x dtype akan menampilkan hasilnya

```
dtype('float64')
```

- hasilnya seperti di atas

```
>>> from sklearn import datasets
>>> from sklearn.svm import SVC
>>> iris = datasets.load_iris()
>>> clf = SVC(gamma='scale')
>>> clf.fit(iris.data, iris.target)
```

- import datasets pada package sklearn
- import svc pada package sklearn
- loading dataset iris
- classifier dengan atribut gamma
- classifier dengan metode fit pada key iris dan target.

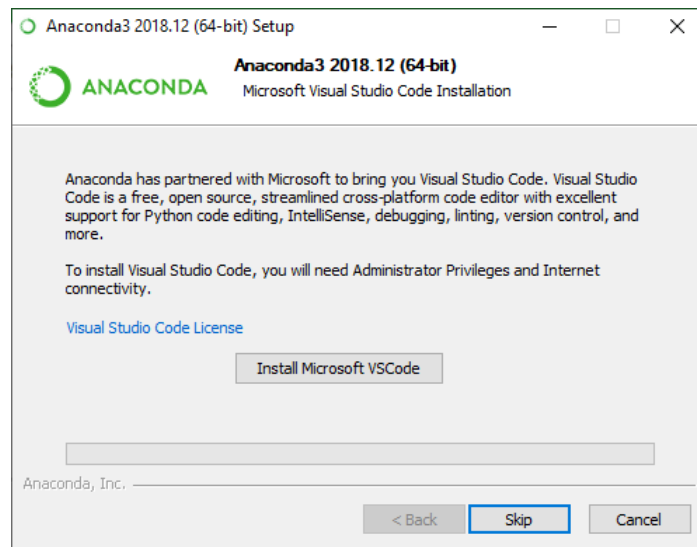


Figure 1.43: Proses Instalasi Aplikasi

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

- hasilnya seperti di atas

```
>>> list(clf.predict(iris.data[:3]))
```

- untuk list classifier predict pada loading dataset iris

```
[0, 0, 0]
```

- hasilnya seperti diatas

```
>>> clf.fit(iris.data, iris.target_names[iris.target])
```

- classifier dengan menggunakan metode fit dan key data dan target

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

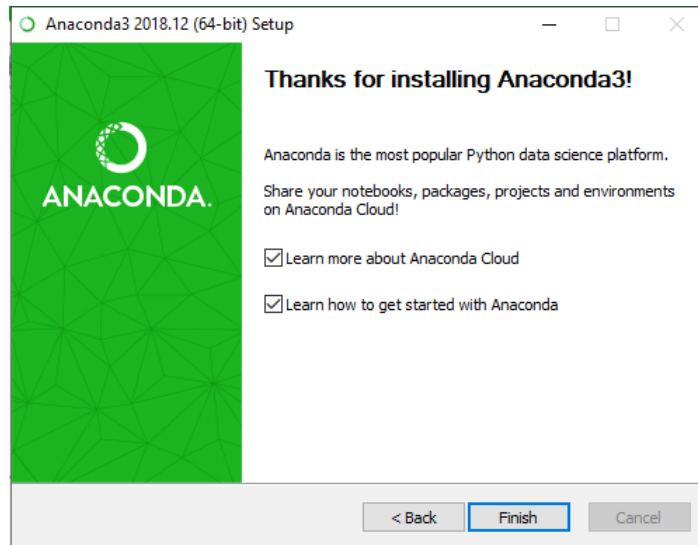


Figure 1.44: Proses Instalasi Aplikasi

```
Administrator: Command Prompt
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>pip install -U scikit-learn
Collecting scikit-learn
  Using cached https://files.pythonhosted.org/packages/ee/c8/c89ebdc0d7dbba6e6fd222daabd257da3c28a967dd7c352d4272b2e1cef6/scikit_learn-0.20.2-cp36-cp36m-win32.whl
Requirement not upgraded as not directly required: scipy>=0.13.3 in c:\programdata\anaconda3\lib\site-packages (from
m scikit-learn) (1.1.0)
Requirement not upgraded as not directly required: numpy>=1.8.2 in c:\programdata\anaconda3\lib\site-packages (from
scikit-learn) (1.14.3)
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: scikit-learn
  Found existing installation: scikit-learn 0.19.1
  Uninstalling scikit-learn-0.19.1:
    Successfully uninstalled scikit-learn-0.19.1
Successfully installed scikit-learn-0.20.2
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Figure 1.45: Instalasi Library

- hasilnya seperti diatas

```
>>> list(clf.predict(iris.data[:3]))
```

- list untuk classifier pada predict loading data iris

```
['setosa', 'setosa', 'setosa']
```

- hasilnya seperti diatas

```
>>> import numpy as np
>>> from sklearn.svm import SVC
>>> rng = np.random.RandomState(0)
>>> X = rng.rand(100, 10)
```

```
Administrator: Command Prompt
C:\WINDOWS\system32>conda install scikit-learn
Solving environment: done

## Package Plan ##

  environment location: C:\ProgramData\Anaconda3

added / updated specs:
- scikit-learn

The following packages will be downloaded:

package-----|-----build-----|-----size-----|
conda-4.6.7      |      py36_0      |      1.7 MB      |

The following packages will be UPDATED:

conda: 4.5.4-py36_0 --> 4.6.7-py36_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.6.7      | 1.7 MB | ##### | 1
00%
Preparing transaction: done
Verifying transaction: done
```

Figure 1.46: Instalasi Library

```
C:\WINDOWS\system32>python --version
Python 3.6.5 :: Anaconda, Inc.

C:\WINDOWS\system32>conda --version
conda 4.6.7
```

Figure 1.47: Instalasi Library

```
>>> y = rng.binomial(1, 0.5, 100)
>>> X_test = rng.rand(5, 10)
>>> clf = SVC()
>>> clf.set_params(kernel='linear').fit(X, y)
```

- import numpy alias np
- import svc dari package sklearn
- rng parameter untuk mencari data pada atribut randomstate
- X memiliki jangkauan rand 100, 10
- y memiliki binomial 5,10
- x memiliki rand 5,10
- classifier dengan atribut svc
- classifier parameter dengan atribut linear menggunakan metode fit


```

Administrator: Command Prompt - python
C:\WINDOWS\system32>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 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.]]
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
>>> digits.images[0]
array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.],
       [ 0.,  0., 13., 15., 10., 15.,  5.,  0.],
       [ 0.,  3., 15.,  2.,  0., 11.,  8.,  0.],
       [ 0.,  4., 12.,  0.,  0.,  8.,  8.,  0.],
       [ 0.,  5.,  8.,  0.,  0.,  9.,  8.,  0.],
       [ 0.,  4., 11.,  0.,  1., 12.,  7.,  0.],
       [ 0.,  2., 14.,  5., 10., 12.,  0.,  0.],
       [ 0.,  0.,  6., 13., 10.,  0.,  0.,  0.]])
>>> from sklearn.preprocessing import MultiLabelBinarizer
>>> y = [[0,1],[0,2],[1,3],[0,2,3],[2,4]]
>>> y = MultiLabelBinarizer().fit_transform(y)
>>> clf.fit(X,y).predict(X)

```

Figure 1.48: Instalasi Library

```

C:\WINDOWS\system32>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 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.]]
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
>>>

```

Figure 1.49: Loading dataset

```

SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)

```

- hasilnya seperti di atas

```
>>> clf.predict(X_test)
```

- classifier untuk memprediksi nilai x

```
array([1, 0, 1, 1, 0])
```

- hasilnya seperti di atas

```
>>> clf.set_params(kernel='rbf', gamma='scale').fit(X, y)
```

- classifier parameters dengan atribut gamma menggunakan metode fit

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

- hasilnya seperti di atas

```
>>> clf.predict(X_test)
```

- classifier prediksi dari x

```
array([1, 0, 1, 1, 0])
```

- hasilnya seperti di atas

```
>>> from sklearn.svm import SVC
>>> from sklearn.multiclass import OneVsRestClassifier
>>> from sklearn.preprocessing import LabelBinarizer
>>> X = [[1,2],[2,4],[4,5],[3,2],[3,1]]
>>> y = [0,0,1,1,2]
>>> classif = OneVsRestClassifier(estimator=SVC(gamma='scale', random_state=0))
>>> classif.fit(X, y).predict(X)
array([0, 0, 1, 1, 2])
>>> y = LabelBinarizer().fit_transform(y)
>>> classif.fit(X, y).predict(X)
```

- import svc dari package sklearn
- import OneVsRestClassifier dari package sklearn
- import LabelBinarizer dari package sklearn
- atribut x
- atribut y
- classifier dengan atribut OneVsRestClassifier dan estimator svc

```
array([[1, 0, 0],
       [1, 0, 0],
       [0, 1, 0],
       [0, 0, 0],
       [0, 0, 0]])
```

- hasilnya seperti di atas

```
>>> from sklearn.preprocessing import MultiLabelBinarizer
>>> y = [[0,1],[0,2],[1,3],[0,2,3],[2,4]]
>>> y = MultiLabelBinarizer().fit_transform(y)
>>> clf.fit(X, y).predict(X)
```

- import MultiLabelBinarizer dari package sklearn
- atribut y
- atribut y akan dijamakan dengan metode fit pada tampilan MultiLabelBinarizer
- classifier dengan metode fit pada x dan y untuk memprediksi

```
array([[1, 1, 0, 0, 0],
       [1, 0, 1, 0, 0],
       [0, 1, 0, 1, 0],
       [1, 0, 1, 0, 0],
       [1, 0, 1, 0, 0]])
```

- hasilnya seperti di atas

1.7.7 Penanganan Error

Dari percobaan yang telah dilakukan terdapat error yang di dapatkan pada bagian joblib model persistence

```
>>> from joblib import dump, load
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'joblib'
>>> dump(clf, 'filename.joblib')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'dump' is not defined
>>>
```

Figure 1.50: Error

```
C:\WINDOWS\system32>pip install joblib
Collecting joblib
  Downloading https://files.pythonhosted.org/packages/cd/c1/50a758e8247561e58cb87305b1e90b171b8c767b15b12a1734001f41d356
/joblib-0.13.2-py2.py3-none-any.whl (278kB)
    100% |#####| 286kB 5.4MB/s
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: joblib
Successfully installed joblib-0.13.2
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\WINDOWS\system32>
```

Figure 1.51: Instalasi

- Error tersebut dikarenakan saya belum install package joblib sehingga error pun terjadi dengan source code sebagai berikut: `from joblib import dump, load`
- Untuk itu solusinya saya install terlebih dahulu joblib agar error pun tidak terjadi kembali.

```
>>> from joblib import dump, load
>>> dump(clf, 'filename.joblib')
['filename.joblib']
>>>
```

Figure 1.52: Hasil

Chapter 2

Related Works

Your related works, and your purpose and contribution which must be different as below.

2.1 Same Topics

Cite every latest journal with same topic

2.1.1 Topic 1

cite for first topic

2.1.2 Topic 2

if you have two topics you can include here to

2.2 Same Method

write and cite latest journal with same method

2.2.1 Method 1

cite and paraphrase method 1

2.2.2 Method 2

cite and paraphrase method 2 if you have more method please add new subsection.

2.3 Yusniar Nur Syarif Sidiq/1164089

2.3.1 Binary Classification

1. Binary Classification atau diartikan kedalam bahasa indonesia yaitu Klasifikasi Biner adalah tugas dalam mengkalrifikasikan elemen-elemen dari himpunan yang diberikan kedalam dua kelompok berdasarkan aturan klarifikasi. Pada ummnya klarifikasi biner akan jatuh ke dalam domain Supervised Learning dan dimana kasus khusus hanya memiliki dua kelas. Beberapa contoh yang meliputi Binary Classification adalah

- Deteksi Transaksi Penipuan Kartu Kredit
- Diagnosa medis
- Deteksi Spam

Untuk contoh Binary Classification dapat dilihat pada gambar 2.1

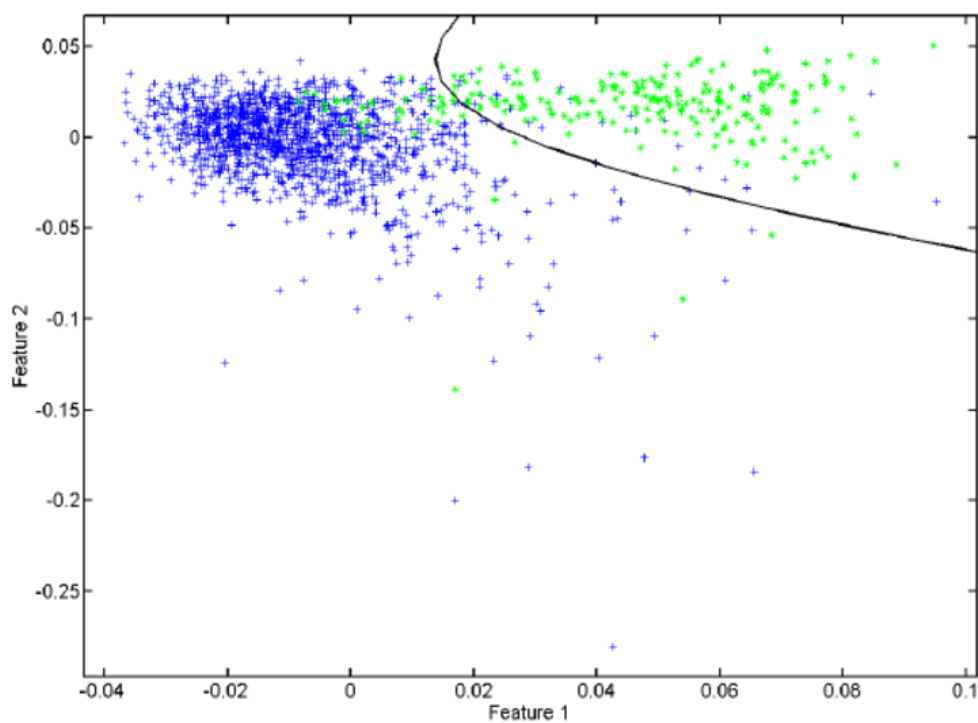


Figure 2.1: Binary Classification.

2.3.2 Supervised Learning, Unsupervised Learning, Dan Classtering

1. Supervised Learning merupakan sebuah pendekatan yang dimana sudah adanya data yang dilatih dan telah terdapat variabel yang telah ditargetkan sehingga bertujuan untuk mengelompokkan suatu data ke data yang sudah ada. Contoh dalam Supervised Learning yaitu ketika anda memiliki sejumlah buku yang telah dilabel dengan urutan kategori tertentu. Ketika anda akan membeli sebuah buku baru, maka harus diidentifikasi isi dari buku tersebut dan memasukkannya kedalam kategori tertentu. Ketika anda membeli sebuah buku tersebut maka anda telah menerapkan sebuah logika fuzzy. Ilustrasi Supervised Learning dapat dilihat pada gambar 2.2.

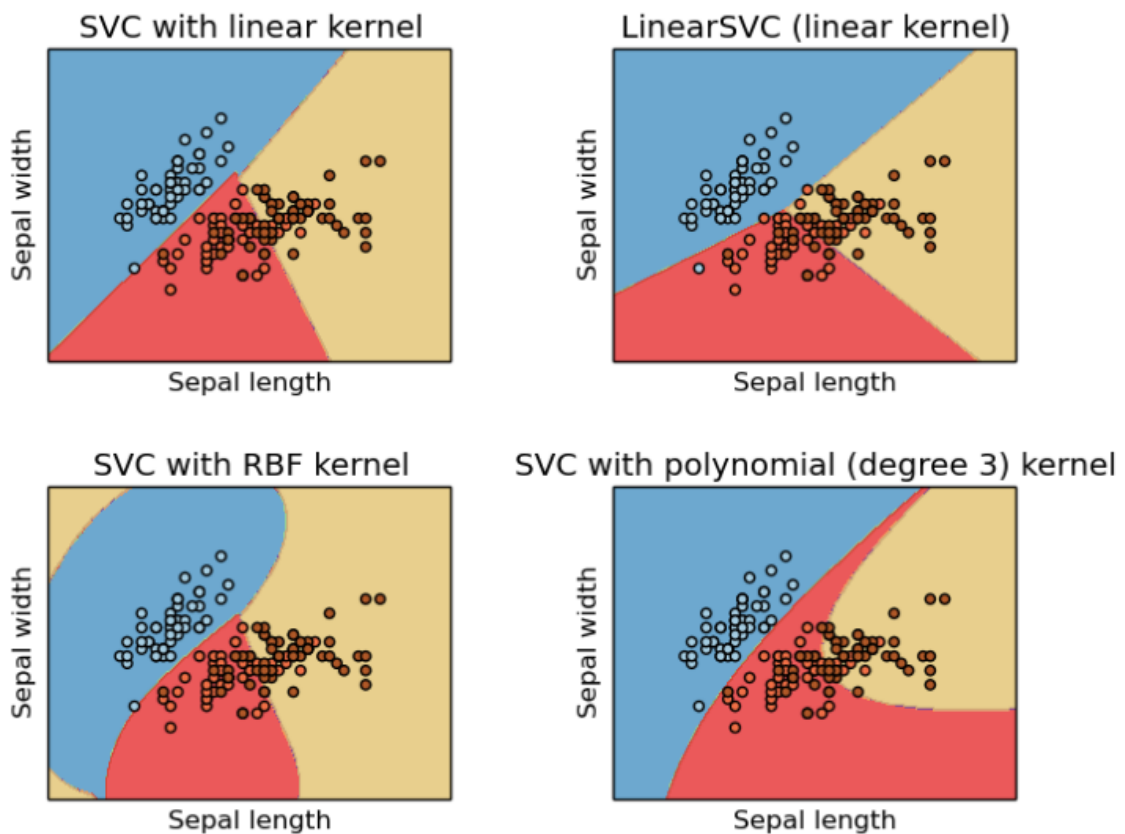


Figure 2.2: Supervised Learning.

2. Unsupervised Learning merupakan sebuah data yang belum ditentukan variabelnya jadi hanya berupa data saja. Dalam sebuah kasus Unsupervised Learning adalah aggap saja anda belum pernah membeli buku sama sekali dan pada

suatu hari anda telah membeli buku dengan sangat banyak dalam kategori yang berbeda. Sehingga buku tersebut belum di kategorikan dan hanya berupa data buku saja. Ilustrasi Unsupervised Learning dapat dilihat pada gambar 2.3.

Unsupervised Learning

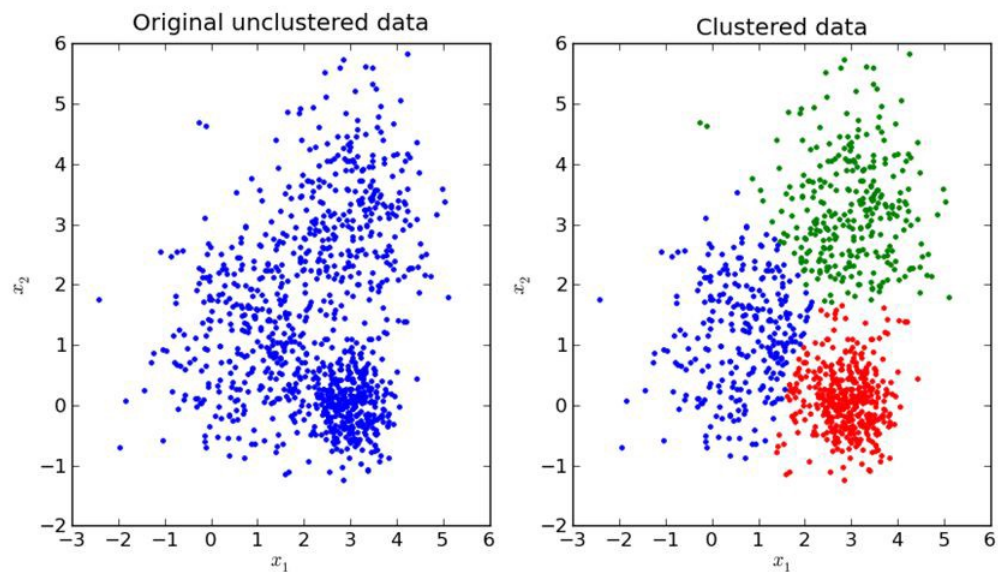


Figure 2.3: Unsupervised Learning.

3. Clustering merupakan sebuah proses untuk mengklasifikasikan sebuah data dalam satu parameter. Dalam kasus ini dapat dijelaskan ada beberapa orang yang memiliki kekuatan tubuh yang sehat dan kekuatan tubuh yang lemah. Parameter bagi orang yang memiliki tubuh yang kuat adalah orang yang terlihat bugar dan sehat maka dengan orang yang memiliki parameter adalah orang yang memiliki kekuatan tubuh yang kuat dan untuk kekuatan tubuh yang lemah adalah sebaliknya. Ilustrasi gambar dapat di lihat di gambar 2.4

2.3.3 Evaluasi Dan Akurasi

1. Evaluasi dan akurasi adalah bagaimana cara kita dapat mengevaluasi seberapa baik model melakukan pekerjaannya dengan cara mengukur akurasinya. Akurasi

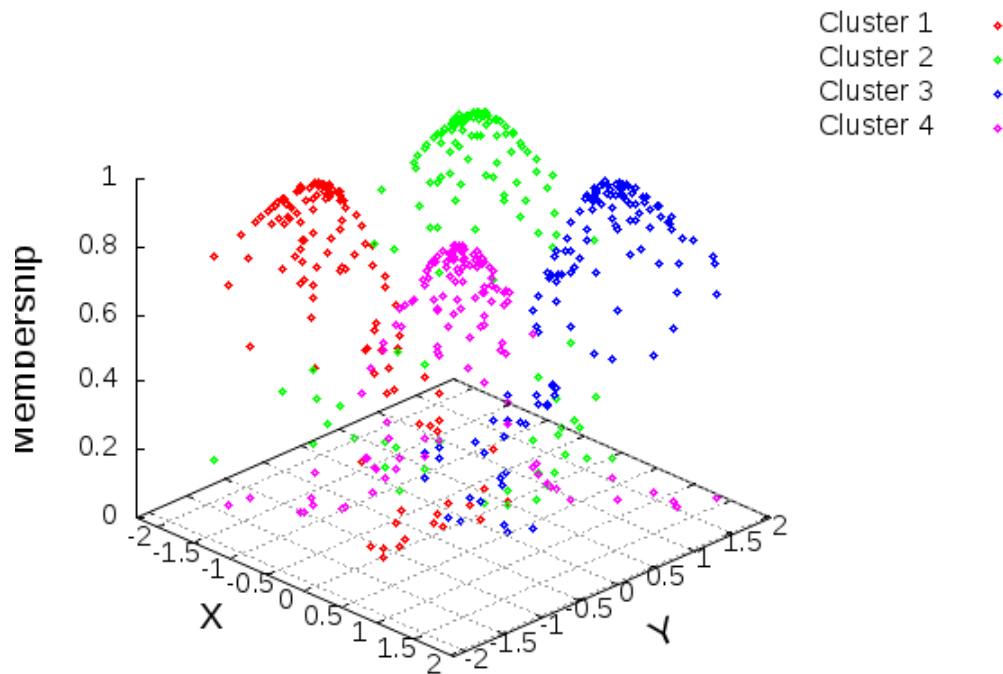


Figure 2.4: Clastering.

akan didefinisikan sebagai presentase kasus yang telah diklasifikasikan dengan benar. Kita dapat melakukan analisis kesalahan yang telah di buat oleh model. Dalam tabel tersebut baris true mangga dan true anggur menunjukkan kasus apakah itu objek mangga atau anggur. Kolom telah di prediksi dan dibuat oleh model. Ada 20 mangga yang di prediksi benar dan ada 5 anggur yang di prediksi salah. Ilustrasi dapat di lihat pada gambar 2.5

2.3.4 Confusion Matrix

1. Ada beberapa cara untuk membuat dan membaca confusion matrix antara lain

- Tentukan pokok permasalahan serta atributnya
- Buat Decision Tree
- Buat Data Testing
- Mencari nilai variabelnya misal a,b,c, dan d
- Mencari nilai recall, precision, accuracy, dan erorr rate

	<u>Pradicted Mangga</u>	<u>Pradicted Anggur</u>
<u>True Mangga</u>	20	5
<u>True Anggur</u>	3	22

Figure 2.5: Evaluasi Dan Akurasi.

Di bawah ini adalah contoh dari confusion matrix

$$\text{Recall} = 3 / (1 + 3) = 0,75$$

$$\text{Precision} = 3 / (1 + 3) = 0,75$$

$$\text{Accuracy} = (5 + 3) / (5 + 1 + 1 + 3) = 0,8$$

$$\text{Error Rate} = (1 + 1) / (5 + 1 + 1 + 3) = 0,2$$

2.3.5 Cara Kerja K-Fold Cross Validation

1. Untuk cara kerja K-Fold Cross Validation adalah sebagai berikut

- Total instance dibagi menjadi N bagian.
- Fold yang pertama adalah bagian pertama menjadi testing data dan sisanya menjadi training data.
- Hitung akurasi berdasarkan porsi data tersebut dengan menggunakan persamaan.
- Fold yang ke dua adalah bagian ke dua menjadi testing data dan sisanya training data.
- Hitung akurasi berdasarkan porsi data tersebut.
- Lakukan step secara berulang hingga habis mencapai fold ke-K.
- Terakhir hitung rata-rata akurasi K buah.

Untuk ilustrasi K-Fold Cross Validation data di lihat pada gambar 2.6

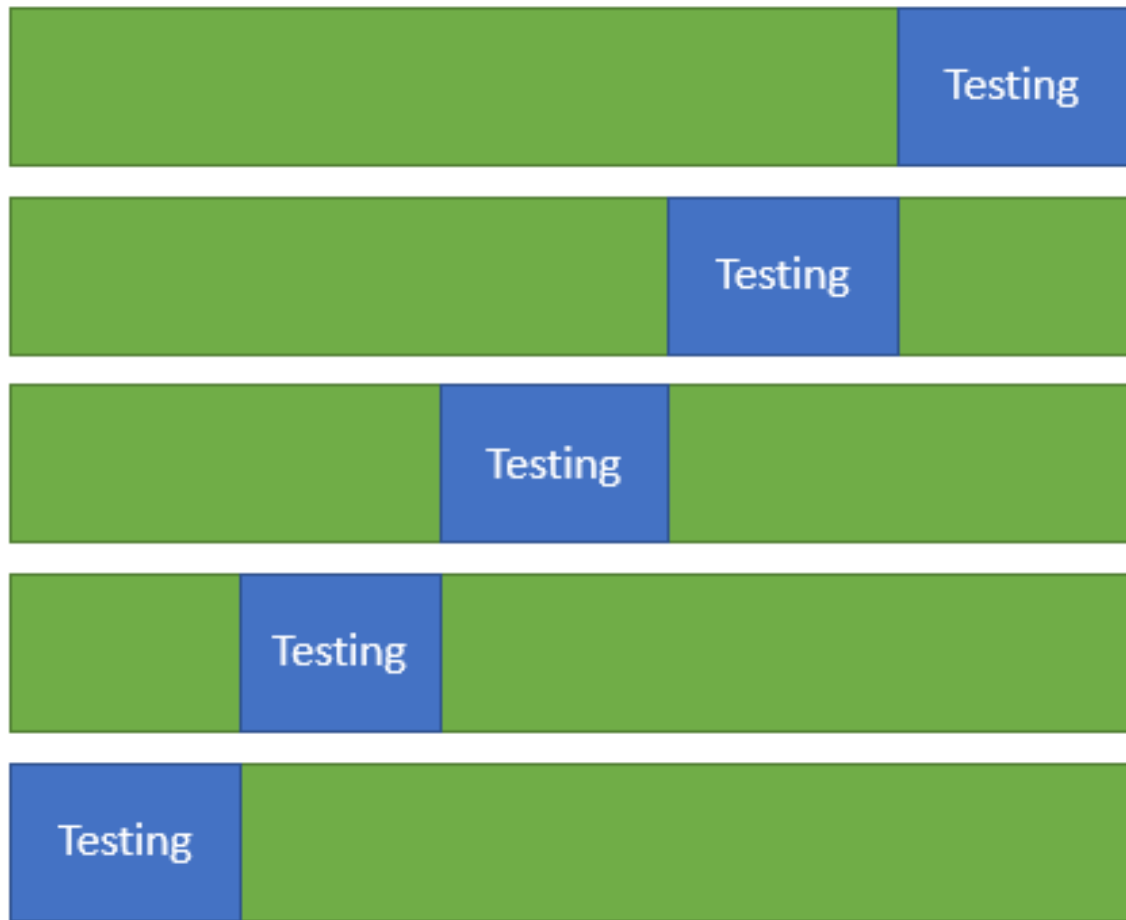


Figure 2.6: K-Fold Cross Validation.

2.3.6 Decision Tree

1. Decision Tree adalah sebuah metode pembelajaran yang digunakan untuk melakukan klarifikasi dan regresi. Decision Tree digunakan untuk membuat sebuah model yang dapat memprediksi sebuah nilai variabel target dengan cara mempelajari aturan keputusan dari fitur data. Contoh Decision Tree adalah untuk melakukan prediksi apakah Kuda termasuk hewan mamalia atau bukan, lihat pada gambar 2.7.

2.3.7 Gain Dan Entropi

1. Gain adalah pengurangan yang diharapkan dalam entropi. Dalam machine learning, gain dapat digunakan untuk menentukan sebuah urutan atribut atau memperkecil atribut yang telah dipilih. Urutan ini akan membentuk decision tree. atribut gain dipilih yang paling besar.

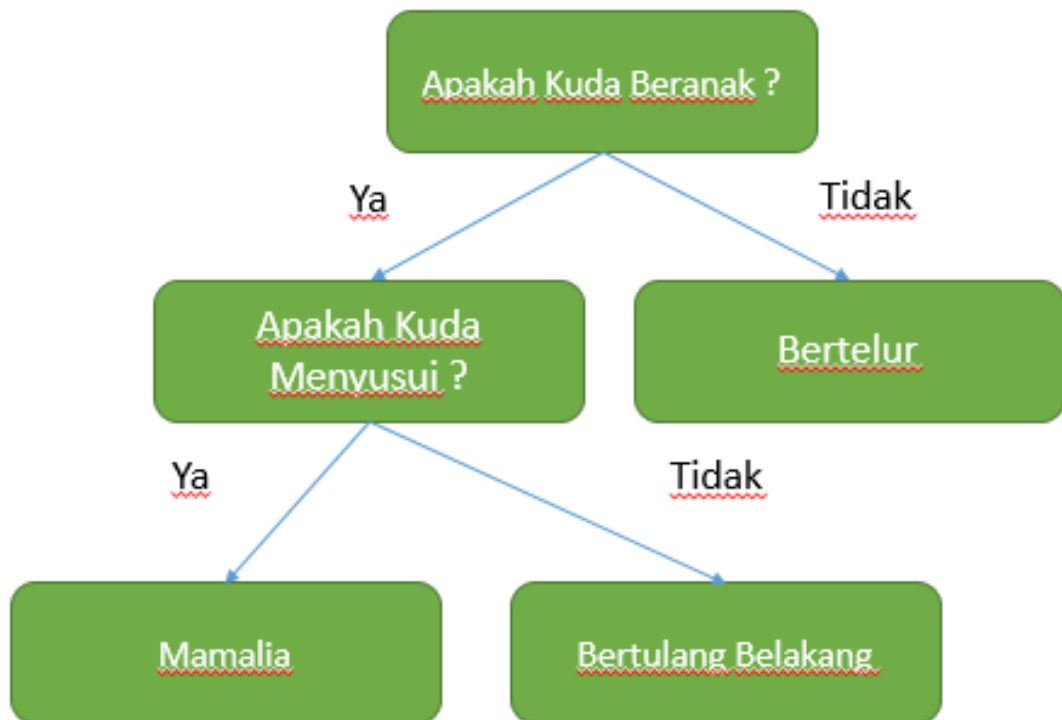


Figure 2.7: Decision Tree.

2. Entropi adalah ukuran ketidakpastian sebuah variabel acak sehingga dapat diartikan entropi adalah ukuran ketidakpastian dari sebuah atribut.

Ilustrasi dari gain dan entropi adalah bagaimana kita memprediksi jenis kelamin berdasarkan atributnya, perhatikan pada gambar 2.8

2.4 Scikit Learn

```

1. import pandas as salak
   sawo = salak.read_csv('D:/Perkuliahan/Semester 6/
   Kecerdasan Buatan/BUKU AI/
   Python-Artificial-Intelligence-Projects-for-Beginners/
   Chapter01/dataset/student-por.csv', sep=';')
   len(sawo)

```

Source Code tersebut digunakan untuk melakukan import pandas yang di rename dengan salak. Disini saya membuat variabel sawo yang dimana isinya fungsi dari read_csv lalu panggil file csv nya yaitu student-por.csv dan separatornya adalah ";". fungsi dari len sendiri yaitu hanya meng outputkan saja.

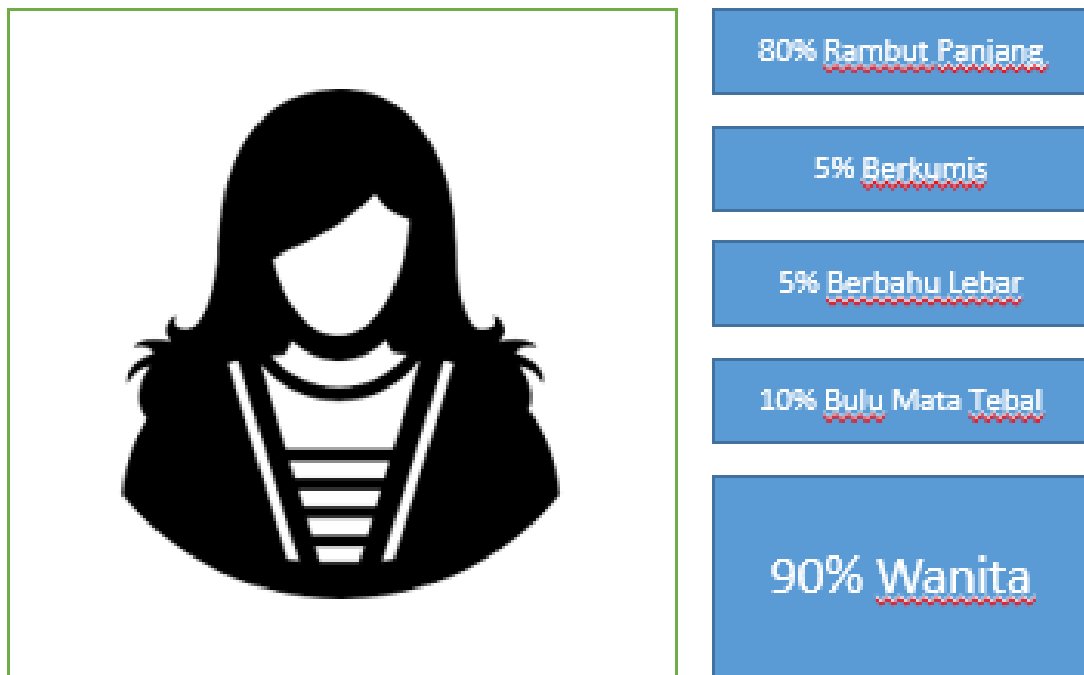


Figure 2.8: Gain Dan Entropi.

Hasil source code yang di running dalam spyder tersebut adalah gambar 2.9

```
In [14]: import pandas as salak
...: sawo = salak.read_csv('D:\Perkuliahan\Semester 6\Kecerdasan Buatan\BUKU AI
\Python-Artificial-Intelligence-Projects-for-Beginners\Chapter01\dataset\student-
por.csv', sep=';')
...: len(sawo)
Out[14]: 649
```

Figure 2.9: Output No 1.

```
2. sawo['pass'] = sawo.apply(lambda row: 1 if
(row['G1']+row['G2']+row['G3']) >= 35 else 0, axis=1)
sawo = sawo.drop(['G1', 'G2', 'G3'], axis=1)
sawo.head()
```

Source Code tersebut akan menambahkan satu field yang diberi nama "pas", dimana lamda tersebut merupakan decision yang berada di dalam decision atau if else. Dimana if else tersebut dinilai apabila lebih dari 35 maka dinyatakan lulus. untuk axis sendiri yaitu apabila lulus maka akan di deklarasikan dengan angka 1 dan apabila tidak lulus akan di deklarasikan dengan angka 0. Selanjutnya akan di running kembali. Hasil running dari spyder dapat dilihat pada

gambar 2.10.

```
Out[15]:
  school sex  age address famsize ... Dalc  Walc  health  absences  pass
0     GP   F   18      U    GT3 ...    1    1      3         4      0
1     GP   F   17      U    GT3 ...    1    1      3         2      0
2     GP   F   15      U    LE3 ...    2    3      3         6      1
3     GP   F   15      U    GT3 ...    1    1      5         0      1
4     GP   F   16      U    GT3 ...    1    2      5         0      1

[5 rows x 31 columns]
```

Figure 2.10: Output No 2.

```
3. sawo = salak.get_dummies(sawo, columns=
    ['sex', 'school', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob',
      'reason', 'guardian', 'schoolsup',
      'famsup', 'paid', 'activities',
      'nursery', 'higher', 'internet', 'romantic'])

sawo.head()
```

Source Code tersebut hanya menambahkan field baru akan tetapi dengan fungsi Binary Classification. Hasil running dari spyder adalah yang ditunjukkan pada gambar 2.11.

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

[5 rows x 57 columns]
```

Figure 2.11: Output No 3.

```
4. sawo = sawo.sample(frac=1)

sawo_train = sawo[:500]
sawo_test = sawo[500:]

sawo_train_att = sawo_train.drop(['pass'], axis=1)
```

```

sawo_train_pass = sawo_train['pass']

sawo_test_att = sawo_test.drop(['pass'], axis=1)
sawo_test_pass = sawo_test['pass']

sawo_att = sawo.drop(['pass'], axis=1)
sawo_pass = sawo['pass']

import numpy as kelapa
print("Passing: %d out of %d (%.2f%%)" % (kelapa.sum(sawo_pass),
    len(sawo_pass),
    100*float(kelapa.sum(sawo_pass)) / len(sawo_pass)))

```

Pada source code tersebut terdapat data train dan juga test yang dimana digunakan untuk membagi data training dan juga data test. Selanjutnya akan melakukan import numpy yang di rename dengan kelapa yang digunakan untuk mengembalikan suatu nilai passing dari keseluruhan datasets dengan cara melakukan print. Hasil running dalam spyder ditunjukkan pada gambar 2.12.

Name	Type	Size	Value
sawo	DataFrame	(649, 57)	Column names: age, Medu, Fedu, traveltime, studytime, failures, famrel ...
sawo_att	DataFrame	(649, 56)	Column names: age, Medu, Fedu, traveltime, studytime, failures, famrel ...
sawo_pass	Series	(649,)	Series object of pandas.core.series module
sawo_test	DataFrame	(149, 57)	Column names: age, Medu, Fedu, traveltime, studytime, failures, famrel ...
sawo_test_att	DataFrame	(149, 56)	Column names: age, Medu, Fedu, traveltime, studytime, failures, famrel ...
sawo_test_pass	Series	(149,)	Series object of pandas.core.series module
sawo_train	DataFrame	(500, 57)	Column names: age, Medu, Fedu, traveltime, studytime, failures, famrel ...
sawo_train_att	DataFrame	(500, 56)	Column names: age, Medu, Fedu, traveltime, studytime, failures, famrel ...

Figure 2.12: Output No 4.

```

5. from sklearn import tree
   anggur = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
   anggur = anggur.fit(sawo_train_att, sawo_train_pass)

```

Dimana akan di importkan modul bernama tree dari library scikitlearn dan kemudian akan di definisikan variabelnya dengan anggur menggunakan Deci-

sionClassifier. Pada variabel anggur terdapat fungsi Criterion yang dapat mengukur kualitas split. Untuk menjalankan DecisionTreeClassifier dibutuhkan sebuah perintah fit. Hasil running dalam spyder dapat dilihat pada gambar 2.13

```
In [18]: from sklearn import tree
...: anggur = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: anggur = anggur.fit(sawo_train_att, sawo_train_pass)
```

Figure 2.13: Output No 5.

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

Graphviz merupakan software visualisasi grafik yang open source. Dimana ada yang dinamakan fungsi Treexportgraphviz, dimana fungsi tersebut dapat menghasilkan representasi Graphviz yang di ambil dari Decision Tree dan ditulis kedalam outfile sehingga dapat memunculkan gambar diagram dengan grafik bercabang. Bila Source code tersebut di running dalam spyder maka hasilnya akan terlihat seperti pada gambar 2.14.

```
7. tree.export_graphviz(anggur, out_file=
"student-performance.dot",
label="all", impurity=False, proportion=True,
feature_names=list(sawo_train_att), class_names=
["fail", "pass"],
filled=True, rounded=True)
```

Treexportgraphviz, dimana fungsi tersebut dapat menghasilkan representasi Graphviz yang di ambil dari Decision Tree dan ditulis kedalam outfile. Dalam source code tersebut akan menyimpan classifier dan melakukan ekspor ke file student performance dan apabila salah akan mengembalikan sebuah nilai fail. Hasil running dalam spyder dapat dilihat pada gambar 2.15.

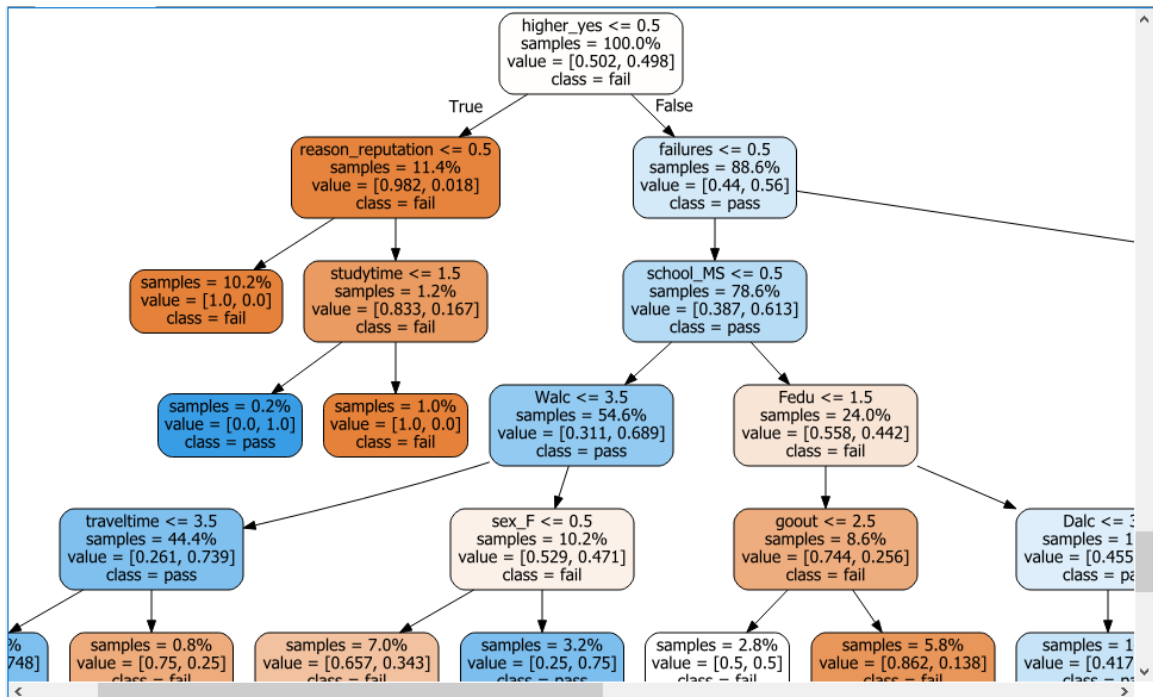


Figure 2.14: Output No 6.

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

Figure 2.15: Output No 7.

8. `anggur.score(sawo_test_att, sawo_test_pass)`

Dimana score adalah sebuah prediksi dan juga merupakan sebuah proses yang menghasilkan sebuah nilai berdasarkan model machine learning yang terlatih dan sudah diberikan beberapa data baru. Score yang telah dibuat dapat mewakili prediksi suatu nilai, dalam kasus ini variabel `anggur` akan melakukan prediksi nilai dari variabel `sawo` dari data testing `att` dan testing `pass`. Apabila source code tersebut di running dalam `spyder` maka hasilnya akan terlihat seperti pada gambar 2.16.

```
In [21]: anggur.score(sawo_test_att, sawo_test_pass)
Out[21]: 0.7248322147651006
```

Figure 2.16: Output No 8.

9. `from sklearn.model_selection import cross_val_score`

```
scores = cross_val_score(anggur, sawo_att, sawo_pass, cv=5)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```

Source code tersebut akan melakukan evaluasi terhadap nilai score dengan validasi silang. Dimana variabel Scores berisikan cross_val_score yaitu sebuah fungsi bantu pada estimator dan juga dataset. Dari hasil tersebut akan ditunjukkan score rata-rata dan kurang-lebih dua standar deviasi yang telah mencakup 95% score, data tersebut akan di print sehingga menunjukkan hasil seperti pada gambar 2.17.

```
In [22]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(anggur, sawo_att, sawo_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))
Accuracy: 0.67 (+/- 0.08)
```

Figure 2.17: Output No 9.

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

Dalam source code tersebut akan mendalami fungsi tree. Dimana semakin dalam tree maka akan semakin banyak perpecahan yang dimiliki dan dapat menangkap lebih banyak informasi. Dalam kasus ini variabel t akan melakukan pendevinisian file tree yang kemudian variabel scores akan melakukan evaluasi score dengan validasi silang. Hasil running source code tersebut di dalam spyder akan di perlihatkan pada gambar 2.18.

```
11. depth_acc = kelapa.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, sawo_att,
sawo_pass, cv=5)
        depth_acc[i,0] =
max_depth
```

```

IPython console
Console 1/A
...: t = tree.DecisionTreeClassifier(criterion='entropy', max_depth=max_depth)
...: scores = cross_val_score(t, sawo_att, sawo_pass, cv=5)
...: print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" % (max_depth, scores.mean(),
scores.std() * 2))
Max depth: 1, Accuracy: 0.62 (+/- 0.05)
Max depth: 2, Accuracy: 0.69 (+/- 0.01)
Max depth: 3, Accuracy: 0.69 (+/- 0.02)
Max depth: 4, Accuracy: 0.69 (+/- 0.06)
Max depth: 5, Accuracy: 0.68 (+/- 0.07)
Max depth: 6, Accuracy: 0.67 (+/- 0.07)
Max depth: 7, Accuracy: 0.69 (+/- 0.05)
Max depth: 8, Accuracy: 0.67 (+/- 0.08)
Max depth: 9, Accuracy: 0.68 (+/- 0.10)
Max depth: 10, Accuracy: 0.66 (+/- 0.09)
Max depth: 11, Accuracy: 0.65 (+/- 0.08)
Max depth: 12, Accuracy: 0.65 (+/- 0.10)
Max depth: 13, Accuracy: 0.64 (+/- 0.08)
Max depth: 14, Accuracy: 0.64 (+/- 0.08)
Max depth: 15, Accuracy: 0.64 (+/- 0.08)
Max depth: 16, Accuracy: 0.66 (+/- 0.10)
Max depth: 17, Accuracy: 0.66 (+/- 0.11)
Max depth: 18, Accuracy: 0.64 (+/- 0.09)
Max depth: 19, Accuracy: 0.64 (+/- 0.10)

In [24]: |

```

Figure 2.18: Output No 10.

```

depth_acc[i,1] =
scores.mean()
depth_acc[i,2] =
scores.std() * 2
i += 1

```

depth_acc

Depth acc akan membuat array kosong dengan mengembalikan array baru dengan bentuk dan tipe yang diberikan, tanpa menginisialisasi entri. Dimana 19 merupakan bentuk array kosong dan 3 adalah output data-typoe sedangkan float urutan kolom-utama dalam memori. Pada source code tersebut jika di running dalam spyder maka akan menunjukkan hasil seperti pada gambar 2.19.

```

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

```

Dimana source code tersebut akan melakukan import dari library matplotlib

```

....
....
....: depth_acc
Out[24]:
array([[1.00000000e+00, 6.16234552e-01, 4.62227393e-02],
       [2.00000000e+00, 6.87243396e-01, 1.23768569e-02],
       [3.00000000e+00, 6.94924142e-01, 2.23150024e-02],
       [4.00000000e+00, 6.82591687e-01, 6.18637635e-02],
       [5.00000000e+00, 6.71834018e-01, 7.60033236e-02],
       [6.00000000e+00, 6.74863966e-01, 6.76815174e-02],
       [7.00000000e+00, 6.82580672e-01, 3.83801373e-02],
       [8.00000000e+00, 6.68733789e-01, 7.48310958e-02],
       [9.00000000e+00, 6.74982680e-01, 1.01101809e-01],
       [1.00000000e+01, 6.54863966e-01, 7.64335218e-02],
       [1.10000000e+01, 6.54863966e-01, 9.17627318e-02],
       [1.20000000e+01, 6.39348528e-01, 1.07921469e-01],
       [1.30000000e+01, 6.51727776e-01, 8.93160325e-02],
       [1.40000000e+01, 6.51751811e-01, 1.11646902e-01],
       [1.50000000e+01, 6.47112028e-01, 7.34148210e-02],
       [1.60000000e+01, 6.39419902e-01, 9.05497305e-02],
       [1.70000000e+01, 6.40910841e-01, 7.46834954e-02],
       [1.80000000e+01, 6.42425269e-01, 7.07916986e-02],
       [1.90000000e+01, 6.34780665e-01, 8.24965986e-02]])

```

```

In [25]:

```

Figure 2.19: Output No 11.

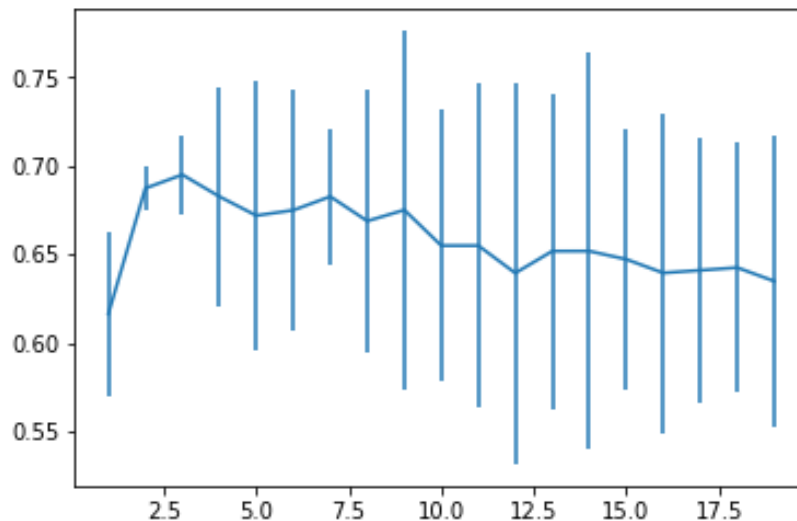
yang berupa pyplot dan di rename menjadi plt. fig dan juga ax akan menggunakan subplots guna membuat sebuah gambar dan satu set subplot. Fungsi ax.errorbar sendiri yaitu akan membuat error bar dan kemudian grafik akan ditampilkan menggunakan plt.show. Hasil source code tersebut jika di running dalam spyder ditunjukkan pada gambar 2.20.

2.5 Penangan Error

1. Untuk Screenshot yang error dapat dilihat pada gambar 2.21.
2. Untuk bagian errornya adalah sebagai berikut

```
ExecutableNotFound: failed to execute
```

```
In [25]: import matplotlib.pyplot as plt
...: fig, ax = plt.subplots()
...: ax.errorbar(depth_acc[:,0], depth_acc[:,1], yerr=depth_acc[:,2])
...: plt.show()
```



```
In [26]:
```

Figure 2.20: Output No 12.

['dot', '-Tsvg'], make sure the Graphviz executables are on your system' PATH

3. Untuk penanganannya dengan cara lakukan instalasi graphviz seperti yang diperlihatkan pada gambar 2.22 dan 2.23.

2.6 Andri Fajar Sunandhar/1164065

2.6.1 binary classification dilengkapi ilustrasi gambar

1. Binary classification yaitu berupa kelas positif dan kelas negatif. Klasifikasi biner adalah dikotomisasi yang diterapkan untuk tujuan praktis, dan dalam banyak masalah klasifikasi biner praktis, kedua kelompok tidak simetris - dari-pada akurasi keseluruhan, proporsi relatif dari berbagai jenis kesalahan yang menarik. Misalnya, dalam pengujian medis, false positive (mendeteksi penyakit ketika tidak ada) dianggap berbeda dari false negative (tidak mendeteksi penyakit ketika hadir).

```

File "C:\Users\NS\Anaconda3\lib\site-packages\graphviz\backend.py", line 206, in pipe
    out, _ = run(cmd, input=data, capture_output=True, check=True, quiet=quiet)

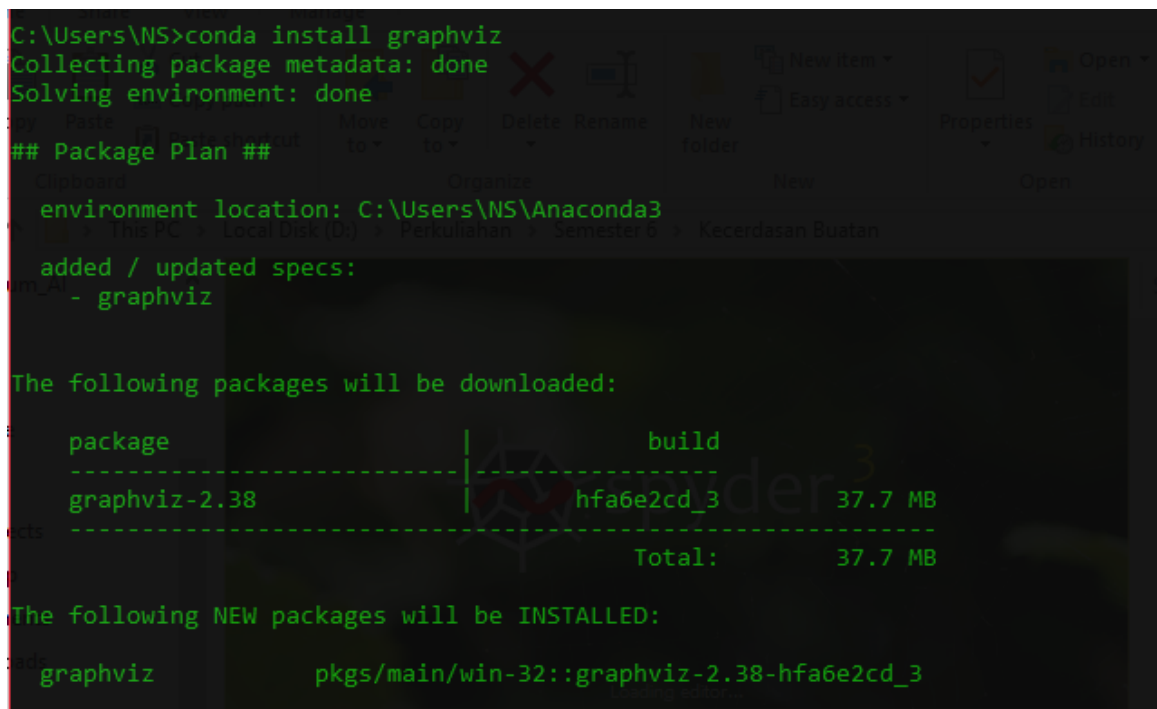
File "C:\Users\NS\Anaconda3\lib\site-packages\graphviz\backend.py", line 150, in run
    raise ExecutableNotFound(cmd)

ExecutableNotFound: failed to execute ['dot', '-Tsvg'], make sure the Graphviz
executables are on your systems' PATH

Out[9]: <graphviz.files.Source at 0x9709fd0>

```

Figure 2.21: Graphviz Errorr.



```

C:\Users\NS>conda install graphviz
Collecting package metadata: done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\NS\Anaconda3
  added / updated specs:
  - graphviz

The following packages will be downloaded:

  package | build | size
  -----|-----|-----
  graphviz-2.38 | hfa6e2cd_3 | 37.7 MB
  -----|-----|-----
  Total: | | 37.7 MB

The following NEW packages will be INSTALLED:

  graphviz | pkgs/main/win-32::graphviz-2.38-hfa6e2cd_3

```

Figure 2.22: Graphviz Install 1.

2.6.2 supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar

1. Supervised learning adalah tugas pembelajaran mesin untuk mempelajari suatu fungsi yang memetakan input ke output berdasarkan contoh pasangan input-output. Ini menyimpulkan fungsi dari data pelatihan berlabel yang terdiri dari serangkaian contoh pelatihan. Dalam pembelajaran yang diawasi, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output yang diinginkan (juga disebut sinyal pengawas). Algoritma pembelajaran yang diawasi menganalisis data pelatihan dan menghasilkan fungsi yang disimpulkan, yang dapat digunakan untuk memetakan contoh-contoh baru. Ske-

```

Proceed ([y]/n)? y
IMPORTANT NOTE: it seems that you are using Spyder with Anaconda/
graphviz-2.38 | 37.7 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
C:\Users\NS>

```

Figure 2.23: Graphviz Install 2.

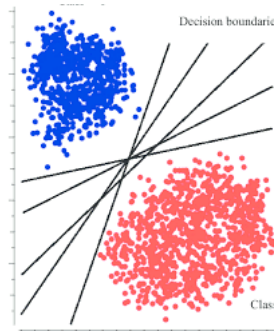


Figure 2.24: Binary Classification

nario optimal akan memungkinkan algoritma menentukan label kelas dengan benar untuk instance yang tidak terlihat. Ini membutuhkan algoritma pembelajaran untuk menggeneralisasi dari data pelatihan untuk situasi yang tidak terlihat dengan cara yang "masuk akal" (lihat bias induktif). Tugas paralel dalam psikologi manusia dan hewan sering disebut sebagai pembelajaran konsep. Contoh dibawah yaitu Supervised Learning dengan SVC.

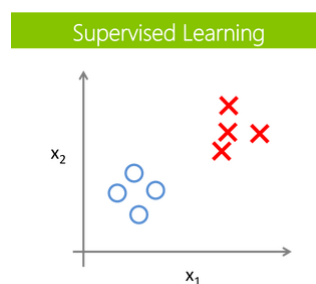


Figure 2.25: Supervised Learning

2. Unsupervised learning adalah istilah yang digunakan untuk pembelajaran bahasa Ibrani, yang terkait dengan pembelajaran tanpa guru, juga dikenal sebagai organisasi mandiri dan metode pemodelan kepadatan probabilitas input. Analisis cluster sebagai cabang pembelajaran mesin yang mengelompokkan data

yang belum diberi label, diklasifikasikan atau dikategorikan. Alih-alih menanggapi umpan balik, analisis kluster mengidentifikasi kesamaan dalam data dan bereaksi berdasarkan ada tidaknya kesamaan di setiap potongan data baru. Berikut merupakan contoh Unsupervised Learning dengan Gaussian mixture models.

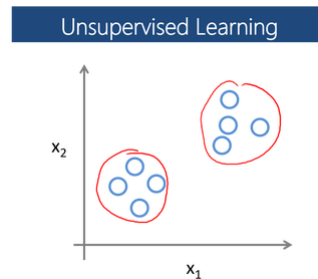


Figure 2.26: Unsupervised Learning

3. Cluster analysis or clustering adalah tugas pengelompokan sekumpulan objek sedemikian rupa sehingga objek dalam kelompok yang sama (disebut kluster) lebih mirip (dalam beberapa hal) satu sama lain daripada pada kelompok lain (kluster). Ini adalah tugas utama penambangan data eksplorasi, dan teknik umum untuk analisis data statistik, yang digunakan di banyak bidang, termasuk pembelajaran mesin, pengenalan pola, analisis gambar, pengambilan informasi, bioinformatika, kompresi data, dan grafik komputer. Analisis Cluster sendiri bukan merupakan salah satu algoritma spesifik, tetapi tugas umum yang harus dipecahkan. Ini dapat dicapai dengan berbagai algoritma yang berbeda secara signifikan dalam pemahaman mereka tentang apa yang merupakan sebuah cluster dan bagaimana cara menemukannya secara efisien. Gagasan populer mengenai cluster termasuk kelompok dengan jarak kecil antara anggota cluster, area padat ruang data, interval atau distribusi statistik tertentu. Clustering karena itu dapat dirumuskan sebagai masalah optimasi multi-objektif. Algoritma pengelompokan dan pengaturan parameter yang sesuai (termasuk parameter seperti fungsi jarak yang akan digunakan, ambang kepadatan atau jumlah cluster yang diharapkan) tergantung pada set data individual dan penggunaan hasil yang dimaksudkan. Analisis kluster bukan merupakan tugas otomatis, tetapi proses berulang penemuan pengetahuan atau optimasi multi-objektif interaktif yang melibatkan percobaan dan kegagalan. Seringkali diperlukan untuk memodifikasi praproses data dan parameter model hingga hasilnya mencapai properti yang diinginkan.

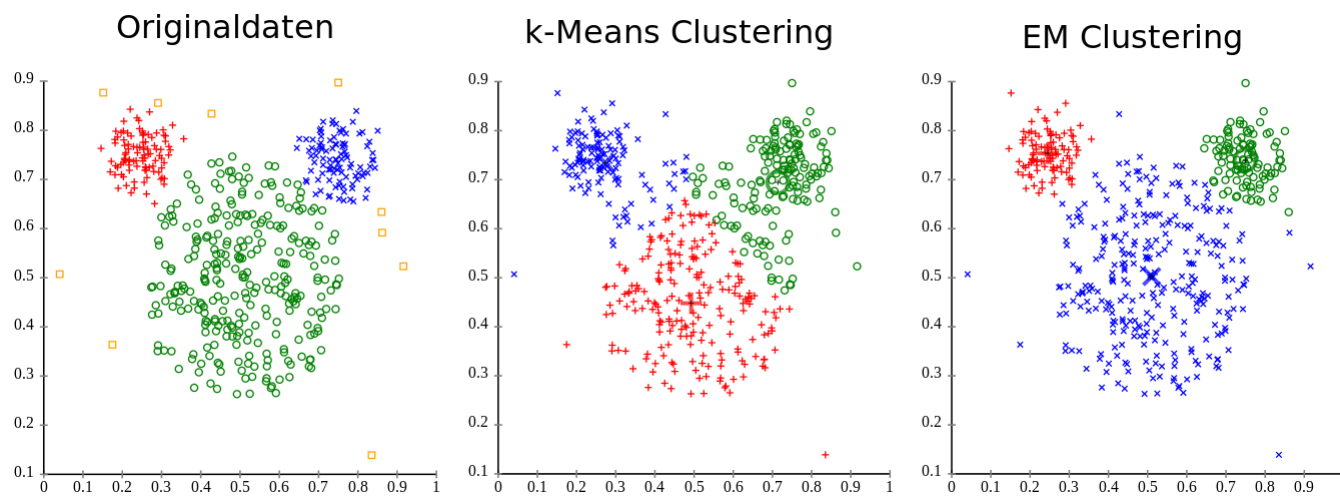


Figure 2.27: Cluster

2.6.3 evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar

1. Evaluasi adalah tentang bagaimana kita dapat mengevaluasi seberapa baik model bekerja dengan mengukur akurasi. Dan akurasi akan didefinisikan sebagai persentase kasus yang diklasifikasikan dengan benar. Kita dapat menganalisis kesalahan yang dibuat oleh model, atau tingkat kebingungannya, menggunakan matriks kebingungan. Matriks kebingungan mengacu pada kebingungan dalam model, tetapi matriks kebingungan ini bisa menjadi sedikit sulit untuk dipahami ketika mereka menjadi sangat besar.

	Predicted Motor	Predicted Mobil
True Motor	10	8
True Mobil	5	50

Figure 2.28: Evaluasi dan Akurasi

2.6.4 bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix

1. Cara membuat dan membaca confusion matrix :
 - 1) Tentukan pokok permasalahan dan atributnya, misal gaji dan listik.

- 2) Buat pohon keputusan
- 3) Lalu data testingnya
- 4) Lalu mencari nilai a, b, c, dan d. Semisal a = 5, b = 1, c = 1, dan d = 3.
- 5) Selanjutnya mencari nilai recall, precision, accuracy, serta dan error rate.

2. Berikut adalah contoh dari confusion matrix :

- Recall = $3/(1+3) = 0,75$
- Precision = $3/(1+3) = 0,75$
- Accuracy = $(5+3)/(5+1+1+3) = 0,8$
- Error Rate = $(1+1)/(5+1+1+3) = 0,2$

2.6.5 bagaimana K-fold cross validation bekerja dengan gambar ilustrasi

1. Cara kerja K-fold cross validation :

- 1) Total instance dibagi menjadi N bagian.
- 2) Fold yang pertama adalah bagian pertama menjadi data uji (testing data) dan sisanya menjadi training data.
- 3) Lalu hitung akurasi berdasarkan porsi data tersebut dengan menggunakan persamaan.
- 4) Fold yang ke dua adalah bagian ke dua menjadi data uji (testing data) dan sisanya training data.
- 5) Kemudian hitung akurasi berdasarkan porsi data tersebut.
- 6) Dan seterusnya hingga habis mencapai fold ke-K.
- 7) Terakhir hitung rata-rata akurasi K buah.

2.6.6 decision tree dengan gambar ilustrasi

1. Decision Tree adalah metode pembelajaran yang diawasi non-parametrik yang digunakan untuk klasifikasi dan regresi. Tujuannya adalah untuk membuat model yang memprediksi nilai variabel target dengan mempelajari aturan keputusan

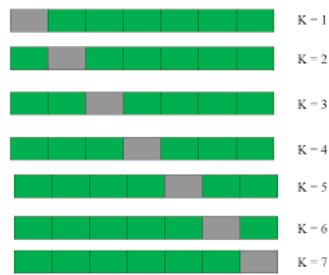


Figure 2.29: K-fold cross validation

sederhana yang disimpulkan dari fitur data.

Misalnya, dalam contoh di bawah ini, decision tree belajar dari data untuk memperkirakan kurva sinus dengan seperangkat aturan keputusan if-then-else. Semakin dalam pohon, semakin rumit aturan keputusan dan semakin bugar modelnya.

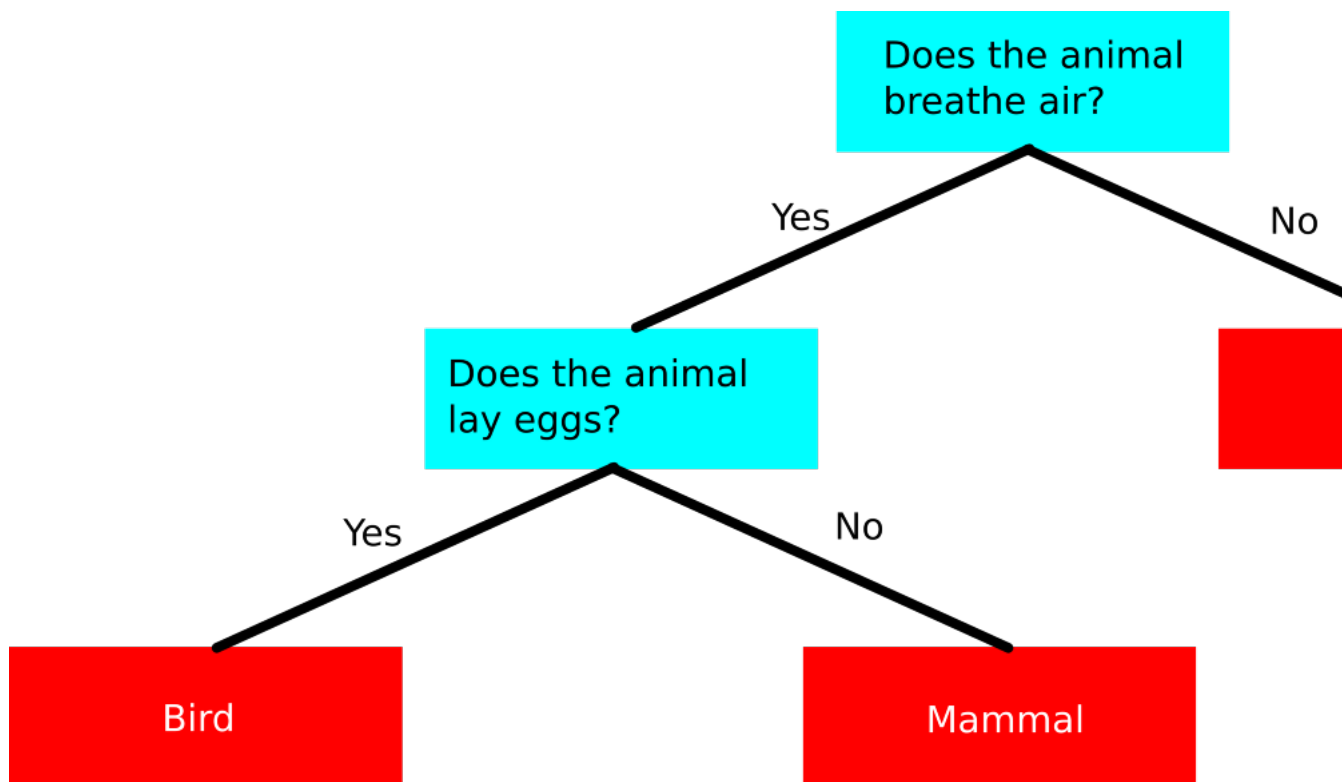


Figure 2.30: Decision Tree

2.6.7 Information Gain dan entropi dengan gambar ilustrasi

1. Information gain didasarkan pada penurunan entropi setelah dataset dibagi pada atribut. Membangun decision tree adalah semua tentang menemukan

atribut yang mengembalikan perolehan informasi tertinggi (mis., Cabang yang paling homogen).



Figure 2.31: Information gain

2. Entropi adalah ukuran keacakan dalam informasi yang sedang diproses. Semakin tinggi entropi, semakin sulit untuk menarik kesimpulan dari informasi itu. Membalik koin adalah contoh tindakan yang memberikan informasi yang acak. Untuk koin yang tidak memiliki afinitas untuk kepala atau ekor, hasil dari sejumlah lemparan sulit diprediksi. Mengapa? Karena tidak ada hubungan antara membalik dan hasilnya. Inilah inti dari entropi.

2.7 Imron Sumadireja / 1164076

2.7.1 Binary Classification

1. Binary classification merupakan suatu cara untuk mengklasifikasikan atau mengkategorikan objek set dengan atribut ke dalam ke dua kategori yang sudah ada atau biasa di sebut dengan supervised. Binary classification dapat diterapkan dengan tujuan praktis, dalam banyak masalah binary classification. Untuk contoh binary classification dapat dilihat pada gambar 2.32

2.7.2 Supervised Learning, Unsupervised Learning, dan Clustering

1. Supervised learning merupakan suatu pembelajaran bagi mesin untuk mempelajari suatu fungsi yang memetakan input ke output berdasarkan data yang telah diberikan dan terdapat variable yang telah ditargetkan sehingga tujuan dari pembelajaran ini mesin dapat memetakan output dengan baik. Sehingga proses training yang dilakukan pada mesin dapat berjalan sesuai dengan target

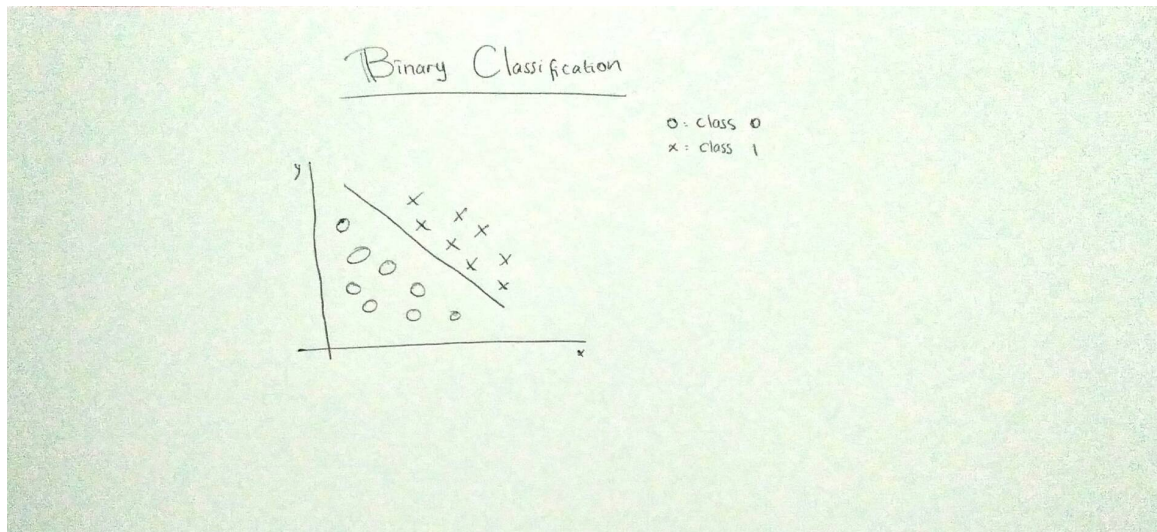


Figure 2.32: Binary Classification.

yang ditentukan dan hasil dari data training tersebut dapat digunakan untuk melakukan prediksi. Contoh supervised learning dapat dilihat pada gambar berikut 2.33

2. Unsupervised learning merupakan suatu pembelajaran bagi mesin, namun tidak memiliki data latih, atau data training. Unsupervised ini dapat mengklasifikasikan suatu objek secara langsung dengan atribut seadanya pada data tersebut. Sebagai contoh, jika kita ingin mengelompokkan sekumpulan orang hanya diperlukan dari data yang ada misalnya dari jenis kelamin, pakaian yang digunakan, dan lain sebagainya. Oleh karena itu unsupervised learning ini tidak memiliki data training. Contoh supervised learning dapat dilihat pada gambar berikut 2.34
3. Clustering adalah metode pengelompokan data ke dalam beberapa cluster atau kelompok agar data dalam satu cluster tersebut memiliki tingkat kemiripan yang maksimum dan data dengan kemiripan yang minimum. Clustering merupakan proses satu set data ke dalam himpunan bagian atau kelompok yang disebut dengan cluster. Contoh clustering dapat dilihat pada gambar berikut 2.36

2.7.3 Evaluasi dan Akurasi

1. Evaluasi adalah tentang bagaimana dapat mengevaluasi seberapa baik model bekerja dengan mengukur akurasinya. Dan akurasi akan didefinisikan sebagai

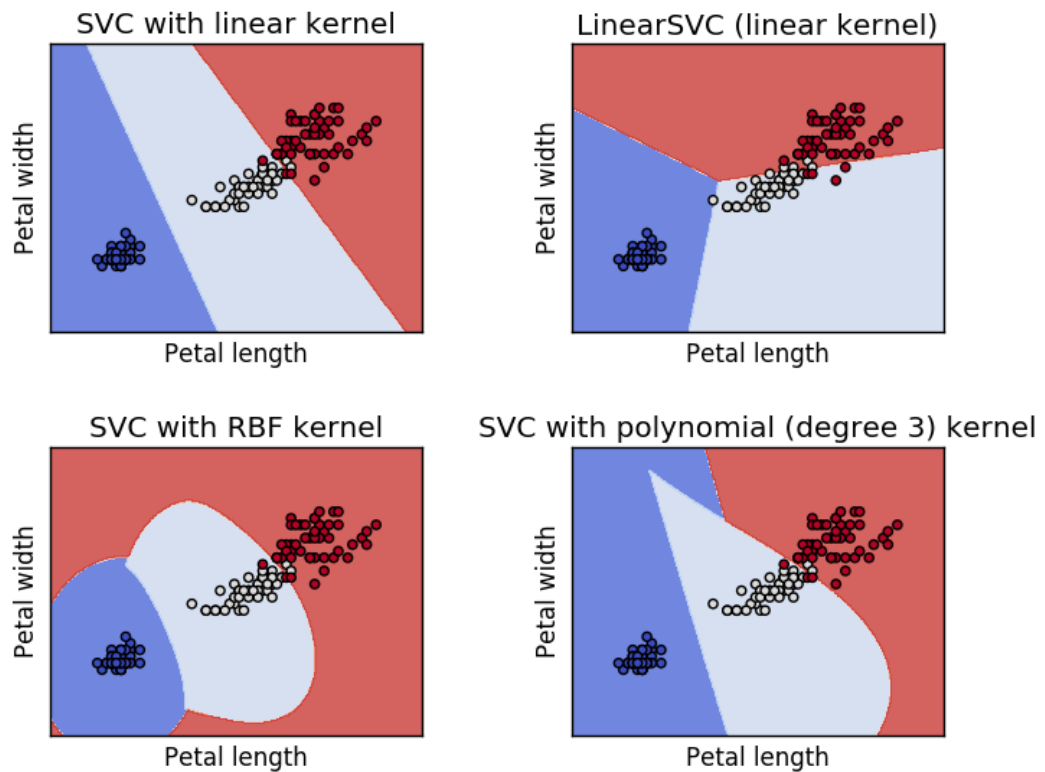


Figure 2.33: Supervised Learning.

persentase kasus yang diklasifikasikan dengan benar. Kita dapat menganalisis kesalahan yang dibuat oleh model, atau tingkat kebingungannya, menggunakan matriks kebingungan. Matriks kebingungan mengacu pada kebingungan mengacu pada kebingungan dalam model, tetapi matriks kebingungan ini bisa menjadi sedikit lebih sulit untuk dipahami ketika mereka menjadi sangat besar. Contohnya dapat dilihat pada gambar berikut ??

2.7.4 Confusion Matrix

1. Terdapat beberapa cara untuk membuat dan membaca confusion matrix diantaranya, sebagai berikut
 - Tentukan pokok permasalahan dan atributnya, misal pendapatan dan pengeluaran.
 - Buat decision tree
 - Buat data testingnya

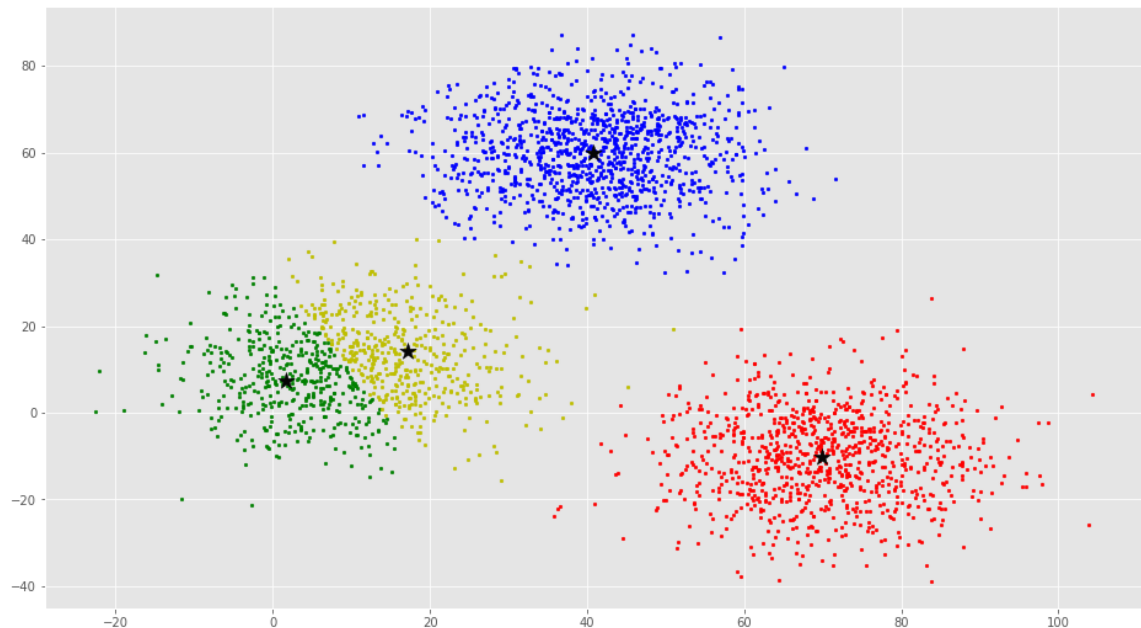


Figure 2.34: Unsupervised Learning.

- Lalu mencari nilai a , b , c dan d . Misal $a = 8$, $b = 2$, $c = 2$, dan $d = 6$.
- Selanjutnya mencari nilai recall, precision, accuracy, dan error rate.

Berikut contoh dari confusion matrix

$$\text{Recall} = 6/(2+6) = 1,33$$

$$\text{Precision} = 6/(2+6) = 1,33$$

$$\text{Accuracy} = (8+6)/(8+2+2+6) = 0,8$$

$$\text{Error rate} = (2+2)/(8+2+2+6) = 0,22$$

2.7.5 Cara kerja K-Fold Cross Validation

1. Untuk cara kerja K-Fold Cross Validation sebagai berikut

- Total instance dibagi menjadi N bagian
- Fold yang pertama adalah bagian pertama menjadi data uji (testing) dan sisanya menjadi training data
- Lalu hitung akurasi berdasarkan porsi data tersebut dengan menggunakan persamaan
- Fold yang kedua adalah bagian ke dua menjadi data uji (testing) dan sisanya menjadi training data

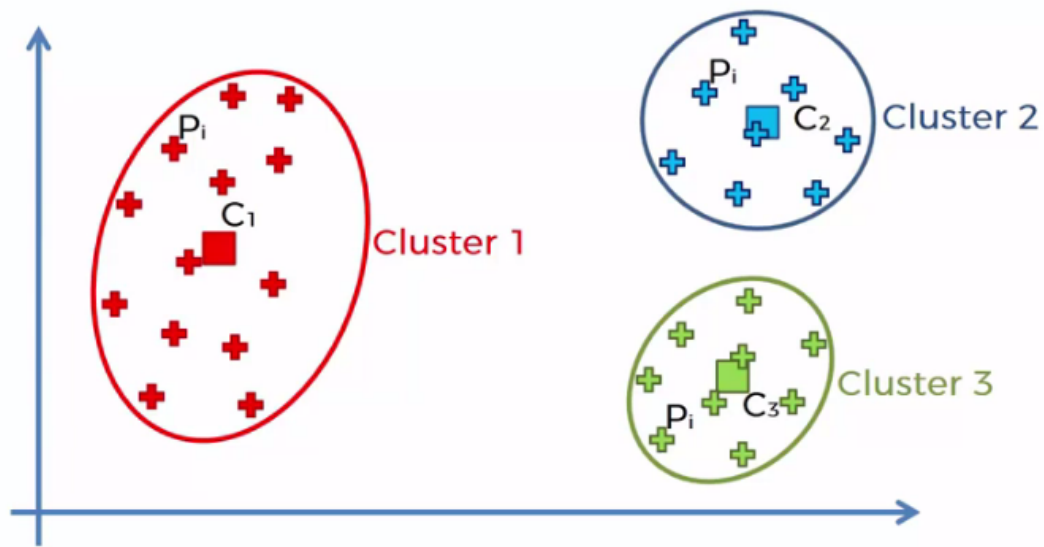


Figure 2.35: Clustering.

- Lalu hitung akurasi berdasarkan porsi data tersebut
- Dan selanjutnya hingga mencapai fold ke-4
- Terakhir hitung rata-rata akurasi K buah.

Ilustrasi dari K-Fold Cross Validation dapat dilihat pada gambar 2.37

2.7.6 Decision Tree

1. Decision tree adalah sebuah metode pembelajaran yang diawasi non-parametrik digunakan untuk klasifikasi dan regresi. Decision tree digunakan untuk membuat sebuah model yang dapat memprediksi variable dengan mempelajari aturan keputusan dengan ciri-ciri yang terdapat pada atribut tersebut. Sebagai contoh decision tree dapat melakukan prediksi apakah di bulan terdapat gravitasi atau bukan. Contohnya dapat dilihat pada gambar berikut 2.38

2.7.7 Information Gain dan Entropi

1. Information Gain adalah informasi atau kriteria dalam pembagian sebuah objek. Sebagai contoh misalnya information gain pada gambar laki-laki, atribut yang biasanya dimiliki pada gambar laki-laki diantaranya berambut pendek, berjakun, berjenggot, berkumis. Dalam beberapa hal terdapat perempuan yang memiliki rambut pendek, berkumis, dan berjenggot, namun dari parameter yang telah diidentifikasi bahwa gambar tersebut memiliki akurasi yang lebih tinggi

	Predict Alpukat	Predict Durian
True Alpukat	10	8
True Durian	2	15

Figure 2.36: Evaluasi dan Akurasi.

jadi dapat disimpulkan bahwa gambar tersebut adalah laki-laki. Untuk lebih jelasnya bisa dilihat dalam gambar 2.39

2. Entropi merupakan ukuran dari keacakan informasi, semakin tinggi entropi maka akan semakin sulit dalam menentukan suatu keputusan

2.8 Imron Sumadireja / 1164076

2.8.1 scikit-learn

Pada praktikum kali ini saya merubah beberapa variable yang terdapat pada source code dengan nama kota. Source code 1:

- Pada baris pertama dari source code tersebut menjelaskan bahwa kita akan import library pandas dengan merubah nama alias menjadi padalarang, seperti gambar berikut 2.40
- Pada baris kedua terdapat variable baru dengan nama dumai, dan akan membaca file dengan ekstensi .csv
- Berikut hasil yang di dapat dari source code berikut 2.41

Source code 2:

- Pada baris pertama menjelaskan bahwa kita akan menambahkan kolom lulus atau gagal. Data dari kolom tersebut akan berisi 1 dan 0. 1 Untuk mahasiswa

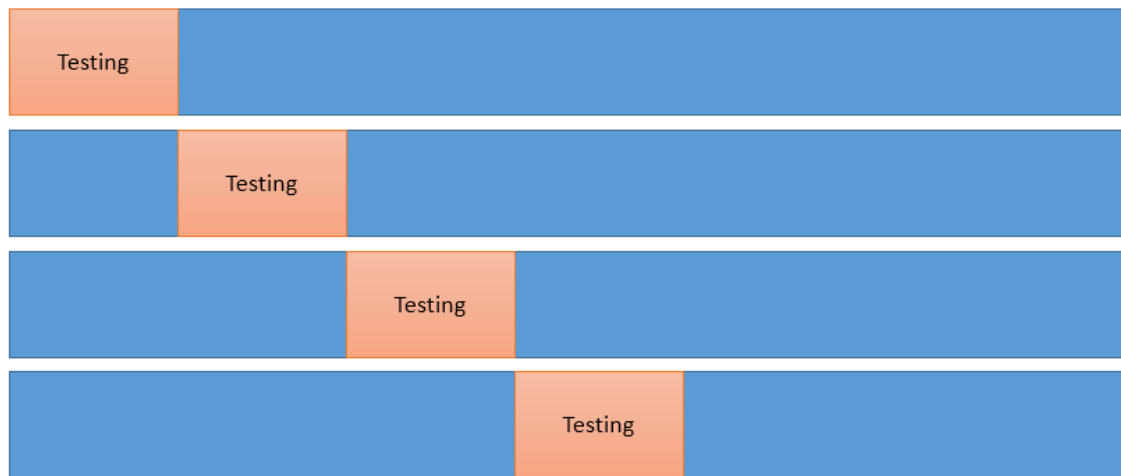


Figure 2.37: K-Fold Cross Validation.

yang dinyatakan lulus dan 0 untuk mahasiswa yang tidak lulus, seperti gambar berikut 2.42

- Pada baris kedua akan membuat data-data tersebut disusun secara berurutan sesuai atribut
- Pada baris ketiga berguna untuk menyinkronkan data yang terdapat pada source code pertama
- Berikut hasil yang di dapat dari source code berikut 2.43

Source code 3:

- Pada baris pertama menjelaskan bahwa dalam data tersebut akan memberikan tambahan kolom dengan atribut dengan isi 0 dan 1, seperti gambar berikut 2.44
- Pada baris kedua berguna untuk menyinkronkan data yang terdapat pada source code sebelumnya
- Berikut hasil yang di dapat dari source code berikut 2.45

Source code 4:

- Pada baris pertama menjelaskan bahwa variable `dumai` akan menjalankan fungsi `sample` dengan `frac 1`, seperti gambar berikut 2.46
- Pada baris kedua dan ketiga berguna untuk memberikan data training dan data testing dengan masing-masing nilai 500

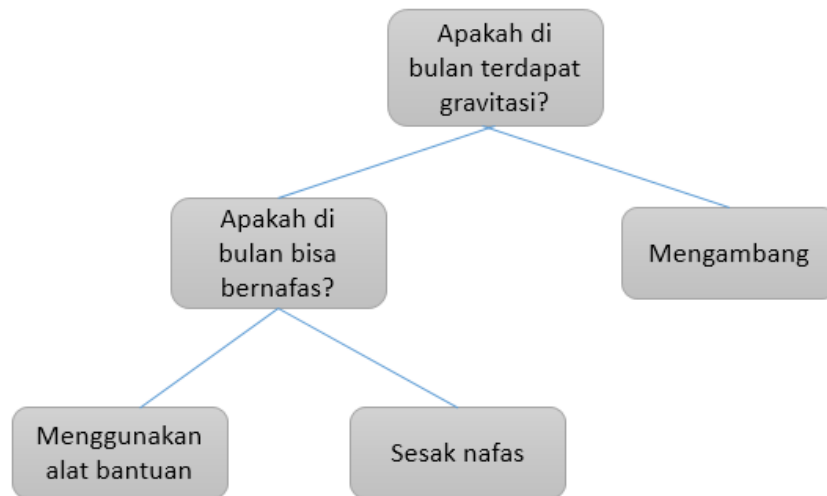


Figure 2.38: Decision Tree.

- Pada baris keempat dan kelima berguna untuk melatih data training
- Pada baris keenam dan ketujuh berguna untuk melatih data testing
- Pada baris kedelapan dan kesembilan berguna untuk membuat sebuah keputusan dari hasil data training dan data testing
- Pada baris kesepuluh berguna untuk import library numpy
- Pada baris kesebelas berguna untuk menampilkan hasil data suatu keputusan tersebut
- Berikut hasil yang di dapat dari source code berikut 2.47

Source code 5:

- Pada baris pertama berguna untuk import libray tree yang berguna untuk membuat keputusan dengan metode tree, seperti gambar berikut 2.48
- Pada baris kedua variable tangerang akan menjalankan fungsi tree decision
- Pada baris ketiga variable tangerang akan menjalankan fungsi tersebut menggunakan data training
- Berikut hasil yang di dapat dari source code berikut 2.49

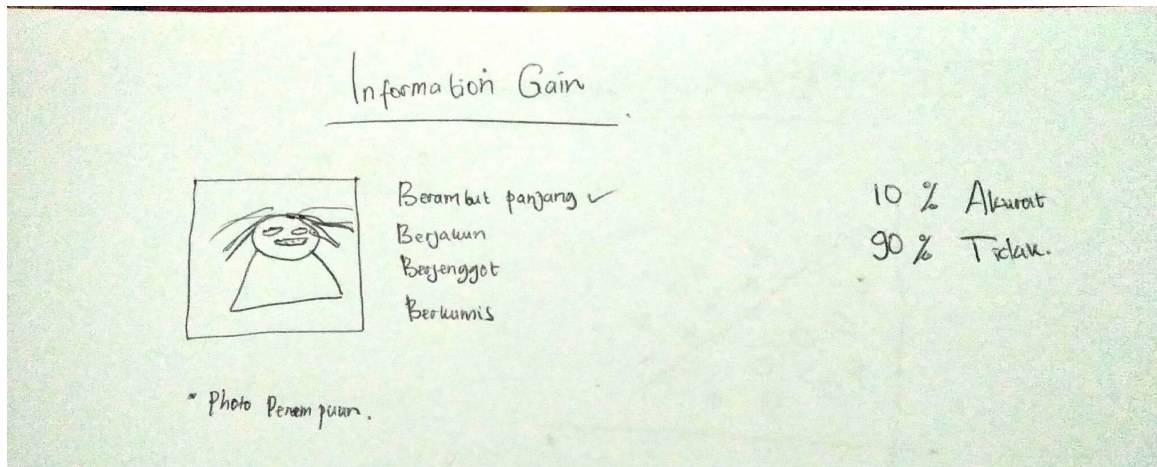


Figure 2.39: Information Gain dan Entropi.

```
# Load dataset (student Portuguese scores)
import pandas as padalarang
dumai = padalarang.read_csv('F:\Imron\Kuliah\Semester 6\Artificial Intelegence\praktikum\Python-Artificial-Intelligence-Projects-for-Beginners\Chapt
len(dumai)
```

Figure 2.40: Source Code.

Source code 6:

- Pada baris pertama berguna untuk import library graphviz, seperti pada gambar berikut 2.50
- Pada baris kedua berguna untuk membuat graphviz dari hasil data yang telah di latih pada source code sebelumnya
- Baris ketiga berguna untuk memanggil atribut dot untuk ditampilkan dalam bentuk graphic
- Berikut hasil yang di dapat dari source code berikut 2.51

Source code 7:

- Source code tersebut berguna untuk mengeksplor representasi visual dalam bentuk PDF atau format lainnya, seperti gambar berikut 2.52
- Berikut hasil yang di dapat dari source code berikut 2.53

Source code 8:

- Source code berikut 2.54 berguna untuk memeriksa skor tree dengan menggunakan set pengujian yang telah di buat sebelumnya

```
In [2]: import pandas as padalarang
...: dumai = padalarang.read_csv('F:\Imron\Kuliah\Semester 6\Artificial
Intelegence\praktikum\Python-Artificial-Intelligence-Projects-for-Beginners
\Chapter01\dataset\student-por.csv', sep=';')
...: len(dumai)
Out[2]: 649
```

Figure 2.41: Source Code.

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

Figure 2.42: Source Code.

- Berikut hasil yang di dapat dari source code berikut 2.55

Source code 9:

- Pada baris pertama berguna untuk import library cross val score, seperti gambar berikut 2.56
- Baris kedua data yang telah dibuat sebelumnya akan kembali digunakan untuk memastikan rata-rata tersebut
- Pada baris ketiga akan menampilkan hasil rata-rata dari data tersebut
- Berikut hasil yang di dapat dari source code berikut 2.57

Source code 10:

- Source code berikut ini berfungsi untuk melakukan pengecekan lebih dalam lagi untuk menentukan keputusan yang lebih akurat dibandingkan dengan metode sebelumnya. Pada source code tersebut melakukan validasi silang, seperti gambar berikut 2.58
- Berikut hasil yang di dapat dari source code berikut 2.59

Source code 11:

- Source code tersebut menjelaskan bahwa untuk mendapatkan hasil keputusan yang akurat diperlukan training yang lebih banyak lagi dalam kasus ini depth acc memiliki nilai 19 dan 3. Proses yang dilakukan sama dengan proses sebelumnya yakni dengan menggunakan decision tree dan dari data hasil data training dan data testing, seperti gambar berikut 2.60

```
In [3]: dumai['pass'] = dumai.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3'])
>= 35 else 0, axis=1)
...: dumai = dumai.drop(['G1', 'G2', 'G3'], axis=1)
...: dumai.head()
Out[3]:
  school sex  age address famsize ... Dalc  Walc  health  absences  pass
0     GP  F   18      U    GT3 ...    1    1      3         4      0
1     GP  F   17      U    GT3 ...    1    1      3         2      0
2     GP  F   15      U    LE3 ...    2    3      3         6      1
3     GP  F   15      U    GT3 ...    1    1      5         0      1
4     GP  F   16      U    GT3 ...    1    2      5         0      1

[5 rows x 31 columns]
```

Figure 2.43: Source Code.

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

Figure 2.44: Source Code.

- Berikut hasil yang di dapat dari source code berikut 2.61

Source code 12:

- Pada baris pertama berguna untuk import library matplotlib.pyplot, seperti gambar berikut 2.62
- Source code tersebut berguna untuk menampilkan diagram hasil keputusan pada pelatihan-pelatihan data sebelumnya
- Berikut hasil yang di dapat dari source code berikut 2.63

2.8.2 Penanganan Error

Dari percobaan yang telah dilakukan saya mengalami 2 kali error, berikut screenshot error serta penaganan yang saya dapat:

1. Screenshot error 2.64
2. Solusi dari permasalahan tersebut, kita tinggal memasukan direktori tempat file tersebut berada 2.65
3. Screenshot error 2.66

```

In [4]: dumai = padalarang.get_dummies(dumai, columns=['sex', 'school', 'address',
'famsize', 'Pstatus', 'Mjob', 'Fjob',
...:                                     'reason', 'guardian', 'schoolsup', 'famsup',
'paid', 'activities',
...:                                     'nursery', 'higher', 'internet', 'romantic'])
...: dumai.head()
Out[4]:
   age  Medu  Fedu  ...  internet_yes  romantic_no  romantic_yes
0   18     4     4  ...              0             1              0
1   17     1     1  ...              1             1              0
2   15     1     1  ...              1             1              0
3   15     4     2  ...              1             0              1
4   16     3     3  ...              0             1              0

[5 rows x 57 columns]

```

Figure 2.45: Source Code.

```

# shuffle rows
dumai = dumai.sample(frac=1)
# split training and testing data
dumai_train = dumai[:500]
dumai_test = dumai[500:]

dumai_train_att = dumai_train.drop(['pass'], axis=1)
dumai_train_pass = dumai_train['pass']

dumai_test_att = dumai_test.drop(['pass'], axis=1)
dumai_test_pass = dumai_test['pass']

dumai_att = dumai.drop(['pass'], axis=1)
dumai_pass = dumai['pass']

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

```

Figure 2.46: Source Code.

4. Solusi dari permasalahan tersebut, kita harus install library graphviz terlebih dahulu seperti gambar 2.67, terhubung saya sudah install maka gambarnya seperti itu.
5. Selanjutnya setelah install library graphviz selesai, kita masukkan path graphviz tersebut kedalam environment variables seperti gambar 2.68 agar dapat digunakan.
6. Setelah itu semua selesai, maka permasalahan pun sudah ditangani.

2.9 scikit-learn

HARI KEDUA ANDRI FAJAR SUNANDHAR 1164065

1. # load dataset (student Portuguese scores)

```

In [5]: dumai = dumai.sample(frac=1)
....: # split training and testing data
....: dumai_train = dumai[:500]
....: dumai_test = dumai[500:]
....:
....: dumai_train_att = dumai_train.drop(['pass'], axis=1)
....: dumai_train_pass = dumai_train['pass']
....:
....: dumai_test_att = dumai_test.drop(['pass'], axis=1)
....: dumai_test_pass = dumai_test['pass']
....:
....: dumai_att = dumai.drop(['pass'], axis=1)
....: dumai_pass = dumai['pass']
....:
....: # number of passing students in whole dataset:
....: import numpy as nabire
....: print("Passing: %d out of %d (%.2f%%)" % (nabire.sum(dumai_pass),
len(dumai_pass), 100*float(nabire.sum(dumai_pass)) / len(dumai_pass)))
Passing: 328 out of 649 (50.54%)

```

Figure 2.47: Source Code.

```

# fit a decision tree
from sklearn import tree
tangerang = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
tangerang = tangerang.fit(dumai_train_att, dumai_train_pass)

```

Figure 2.48: Source Code.

```

import pandas as apel
jeruk = apel.read_csv('E:\KAMPUS\Semester 6\Kecerdasan Buatan\modul\Python-Ar
len(jeruk)

```

Untuk mengimport atau memanggil module pandas sebagai apel. Kemudian mendefinisikan variabel "jeruk" yang akan memanggil dataset yang didapatkan dari data student-mat.csv

2. # generate binary label (pass/fail) based on G1+G2+G3 (test grades, each 0-20)

```

jeruk['pass'] = jeruk.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3'])
jeruk = jeruk.drop(['G1', 'G2', 'G3'], axis=1)
jeruk.head()

```

mendeklarasikan label pass/fail nya data berdasarkan G1+G2+G3. kemudian pada variabel jeruk dideklarasikan jika baris dengan G1+G2+G3 ditambahkan, dan hasilnya sama dengan 35 maka axisnya 1.


```
In [6]: from sklearn import tree
...: tangerang = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: tangerang = tangerang.fit(dumai_train_att, dumai_train_pass)
```

Figure 2.49: Source Code.

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

Figure 2.50: Source Code.

3. # use one-hot encoding on categorical columns

```
jeruk = apel.get_dummies(jeruk, columns=['sex', 'school', 'address', 'famsize',
                                         'reason', 'guardian', 'schoolsup', 'famsup',
                                         'nursery', 'higher', 'internet', 'romantic'])

jeruk.head()
```

One-hot encoding adalah proses di mana variabel kategorikal dikonversi menjadi bentuk yang dapat disediakan untuk algoritma .

4. # shuffle rows

```
jeruk = jeruk.sample(frac=1)
# split training and testing data
jeruk_train = jeruk[:500]
jeruk_test = jeruk[500:]

jeruk_train_att = jeruk_train.drop(['pass'], axis=1)
jeruk_train_pass = jeruk_train['pass']

jeruk_test_att = jeruk_test.drop(['pass'], axis=1)
jeruk_test_pass = jeruk_test['pass']

jeruk_att = jeruk.drop(['pass'], axis=1)
jeruk_pass = jeruk['pass']

# number of passing students in whole dataset:
import numpy as np
```

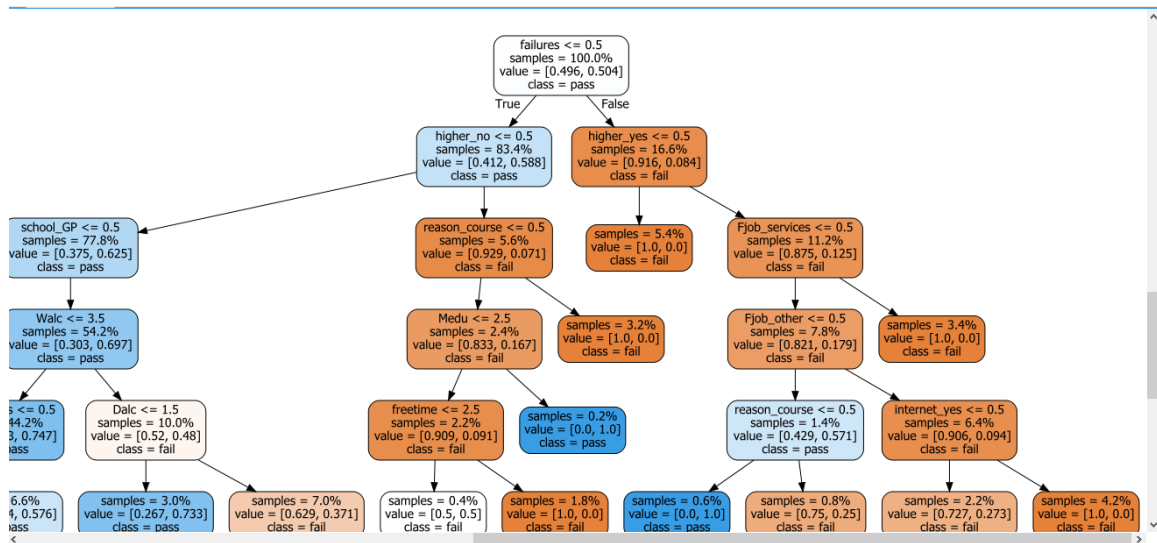


Figure 2.51: Source Code.

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

Figure 2.52: Source Code.

```
print("Passing: %d out of %d (%.2f%%)" % (np.sum(jeruk_pass), len(jeruk_pass)))
```

Pada bagian tersebut, terdapat train dan test yang digunakan untuk untuk membagi train, test dan kemudian membagi lagi train ke validasi dan test.

Kemudian akan mengimport module numpy sebagai np yang akan digunakan untuk mengembalikan nilai passing dari pelajar dari keseluruhan dataset dengan cara print.

```
5. # fit a decision tree
from sklearn import tree
semangka = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
semangka = semangka.fit(jeruk_train_att, jeruk_train_pass)
```

Dari librari scikitlearn import modul tree. Kemudian definisikan variabel semangka dengan menggunakan DecisionTreeClassifier. Kemudian pada variabel semangka terdapat Criterion, setelah itu agar DecisionTreeClassifier dapat dijalankan gunakan perintah fit. hasilnya seperti dibawah

```
6. # visualize tree
import graphviz
```

```
In [8]: tree.export_graphviz(tangerang, out_file="student-
performance.dot", label="all", impurity=False, proportion=True,
...:                        feature_names=list(dumai_train_att),
class_names=["fail", "pass"],
...:                        filled=True, rounded=True)
```

Figure 2.53: Source Code.

```
tangerang.score(dumai_test_att, dumai_test_pass)
```

Figure 2.54: Source Code.

```
dot_data = tree.export_graphviz(semangka, out_file=None, label="all", impurity=
...:                        feature_names=list(jeruk_train_att), class_names=
...:                        filled=True, rounded=True)
```

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

```
graph
```

Mengimport Graphviz Sehingga akan muncul gambardiagram grafik bercabang.

7. # save tree

```
tree.export_graphviz(semangka, out_file="student-performance.dot", label="all",
...:                        feature_names=list(jeruk_train_att), class_names=["fail",
...:                        filled=True, rounded=True)
```

tree.exportgraphviz merupakan fungsi yang menghasilkan representasi Graphviz dari decision tree.

8. semangka.score(jeruk_test_att, jeruk_test_pass)

Score juga disebut prediksi, Nilai atau skor yang dibuat dapat mewakili prediksi nilai masa depan, tetapi mereka juga mungkin mewakili kategori atau hasil yang mungkin. disini semangka akan memprediksi jeruk.

9. from sklearn.model_selection import cross_val_score

```
scores = cross_val_score(semangka, jeruk_att, jeruk_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]: tangerang.score(dumai_test_att, dumai_test_pass)
Out[9]: 0.6778523489932886
```

Figure 2.55: Source Code.

```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(tangerang, dumai_att, dumai_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))
```

Figure 2.56: Source Code.

Dari `sklearn.modelselection` akan mengimport `crossvalscore`. Kemudian akan menampilkan score rata rata dan kurang lebih dua standar deviasi yang mencakup 95 persen score.

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

Semangka akan mendefinisikan `tree.DecisionTreeClassifier` nya yang kemudian variabel `semangka` akan mengevaluasi score dengan validasi silang.

```
11. depth_acc = np.empty((19,3), float)
    i = 0
    for max_depth in range(1, 20):
        semangka = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
        scores = cross_val_score(semangka, jeruk_att, jeruk_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`

Dengan 19 sebagai bentuk array kosong, 3 sebagai output data-type dan float urutan kolom-utama (gaya Fortran) dalam memori. variabel `semangka` yang akan melakukan split score dan angka akan mengvalidasi score secara silang.

```
In [10]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(tangerang, dumai_att, dumai_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))
Accuracy: 0.70 (+/- 0.06)

In [11]:
```

Figure 2.57: Source Code.

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

Figure 2.58: Source Code.

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

Mengimpor librari dari matplotlib yaitu pyplot sebagai plt
fig dan ax menggunakan subplots untuk membuat gambar .
ax.errorbar akan membuat error bar

2.10 Penanganan Error

Hari Kedua Andri fajar Sunandhar 1164065

2.10.1 Error Graphviz

1. error yang didapatkan saat menjalankan Graphviz
2. Kode erornya adalah ModuleNotFoundError. Error ini terjadi karena module named Graphviz nya tidak ada.

```

In [11]: for max_depth in range(1, 20):
...:     tangerang =
tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...:     scores = cross_val_score(tangerang, dumai_att,
dumai_pass, cv=5)
...:     print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" %
(max_depth, scores.mean(), scores.std() * 2))
Max depth: 1, Accuracy: 0.64 (+/- 0.07)
Max depth: 2, Accuracy: 0.69 (+/- 0.08)
Max depth: 3, Accuracy: 0.70 (+/- 0.06)
Max depth: 4, Accuracy: 0.69 (+/- 0.04)
Max depth: 5, Accuracy: 0.70 (+/- 0.07)
Max depth: 6, Accuracy: 0.67 (+/- 0.06)
Max depth: 7, Accuracy: 0.67 (+/- 0.08)
Max depth: 8, Accuracy: 0.64 (+/- 0.06)
Max depth: 9, Accuracy: 0.67 (+/- 0.04)
Max depth: 10, Accuracy: 0.65 (+/- 0.05)
Max depth: 11, Accuracy: 0.65 (+/- 0.08)
Max depth: 12, Accuracy: 0.65 (+/- 0.08)
Max depth: 13, Accuracy: 0.62 (+/- 0.04)
Max depth: 14, Accuracy: 0.62 (+/- 0.05)
Max depth: 15, Accuracy: 0.63 (+/- 0.05)
Max depth: 16, Accuracy: 0.63 (+/- 0.03)
Max depth: 17, Accuracy: 0.63 (+/- 0.03)
Max depth: 18, Accuracy: 0.61 (+/- 0.04)
Max depth: 19, Accuracy: 0.63 (+/- 0.07)

```

Figure 2.59: Source Code.

3. Solusi yang bisa dilakukan untuk mengatasi eror tersebut adalah sebagai berikut :

- buka CMD kemudian perintah pip install graphviz
- masukan perintah conda install pip, untuk solving environment
- selanjutnya masukan perintah conda install python-graphviz , untuk menambahkan package python-graphviz pada conda

```

depth_acc = nabire.empty((19,3), float)
i = 0
for max_depth in range(1, 20):
    tangerang = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
    scores = cross_val_score(tangerang, dumai_att, dumai_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

```

Figure 2.60: Source Code.

```

In [12]: depth_acc = nabire.empty((19,3), float)
...: i = 0
...: for max_depth in range(1, 20):
...:     tangerang =
tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...:     scores = cross_val_score(tangerang, dumai_att,
dumai_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
Out[12]:
array([[ 1.,      0.63801062,  0.06582114],
       [ 2.,      0.68733808,  0.07826517],
       [ 3.,      0.69961108,  0.05769039],
       [ 4.,      0.69340899,  0.04174051],
       [ 5.,      0.69953971,  0.05912439],
       [ 6.,      0.67640243,  0.04914176],
       [ 7.,      0.66881636,  0.081656  ],
       [ 8.,      0.64724249,  0.06734589],
       [ 9.,      0.67043794,  0.06886266],
       [10.,      0.67506525,  0.07532491],
       [11.,      0.65191695,  0.08060474],
       [12.,      0.65817759,  0.09001119],
       [13.,      0.63043111,  0.03318451],
       [14.,      0.63043111,  0.03318451],
       [15.,      0.63043111,  0.03318451],
       [16.,      0.63043111,  0.03318451],
       [17.,      0.63043111,  0.03318451],
       [18.,      0.63043111,  0.03318451],
       [19.,      0.63043111,  0.03318451]])

```

Figure 2.61: Source Code.

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

Figure 2.62: Source Code.

```
In [13]: import matplotlib.pyplot as plt
...: fig, ax = plt.subplots()
...: ax.errorbar(depth_acc[:,0], depth_acc[:,1], yerr=depth_acc[:,
2])
...: plt.show()
```

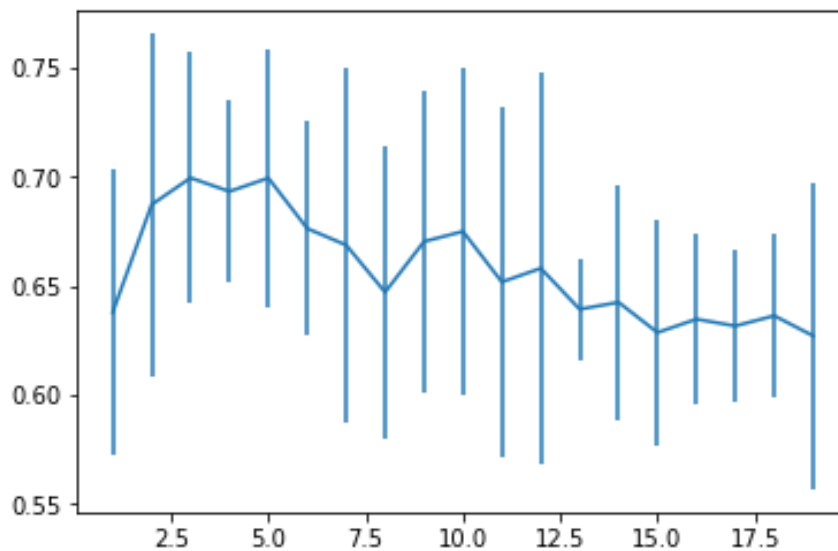


Figure 2.63: Source Code.


```
File C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers.py, line
1708, in __init__
    self._reader = parsers.TextReader(src, **kwargs)

File "pandas\_libs\parsers.pyx", line 384, in
pandas._libs.parsers.TextReader._cinit_

File "pandas\_libs\parsers.pyx", line 695, in
pandas._libs.parsers.TextReader._setup_parser_source

FileNotFoundError: File b'student-por.csv' does not exist

In [2]:
In [2]:
```

Figure 2.64: Error.

```
# Load dataset (Student Portuguese scores)
import pandas as padalarang
dumai = padalarang.read_csv('F:\Imron\Kuliah\Semester 6\Artificial Intelligence\praktikum\Python-Artificial-Intelligence-Projects-for-Beginners\Chapter01\
len(dumai)]
```

Figure 2.65: Resolve.

```
File "C:\Users\NS\Anaconda3\lib\site-packages\graphviz\backend.py", line 206, in pipe
    out, _ = run(cmd, input=data, capture_output=True, check=True, quiet=quiet)

File "C:\Users\NS\Anaconda3\lib\site-packages\graphviz\backend.py", line 150, in run
    raise ExecutableNotFound(cmd)

ExecutableNotFound: failed to execute ['dot', '-Tsvg'], make sure the Graphviz
executables are on your systems' PATH

Out[9]: <graphviz.files.Source at 0x9709fd0>
```

Figure 2.66: Error.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.316]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>conda install graphviz
Collecting package metadata: done
Solving environment: done

# All requested packages already installed.

C:\WINDOWS\system32>
```

Figure 2.67: Resolve.

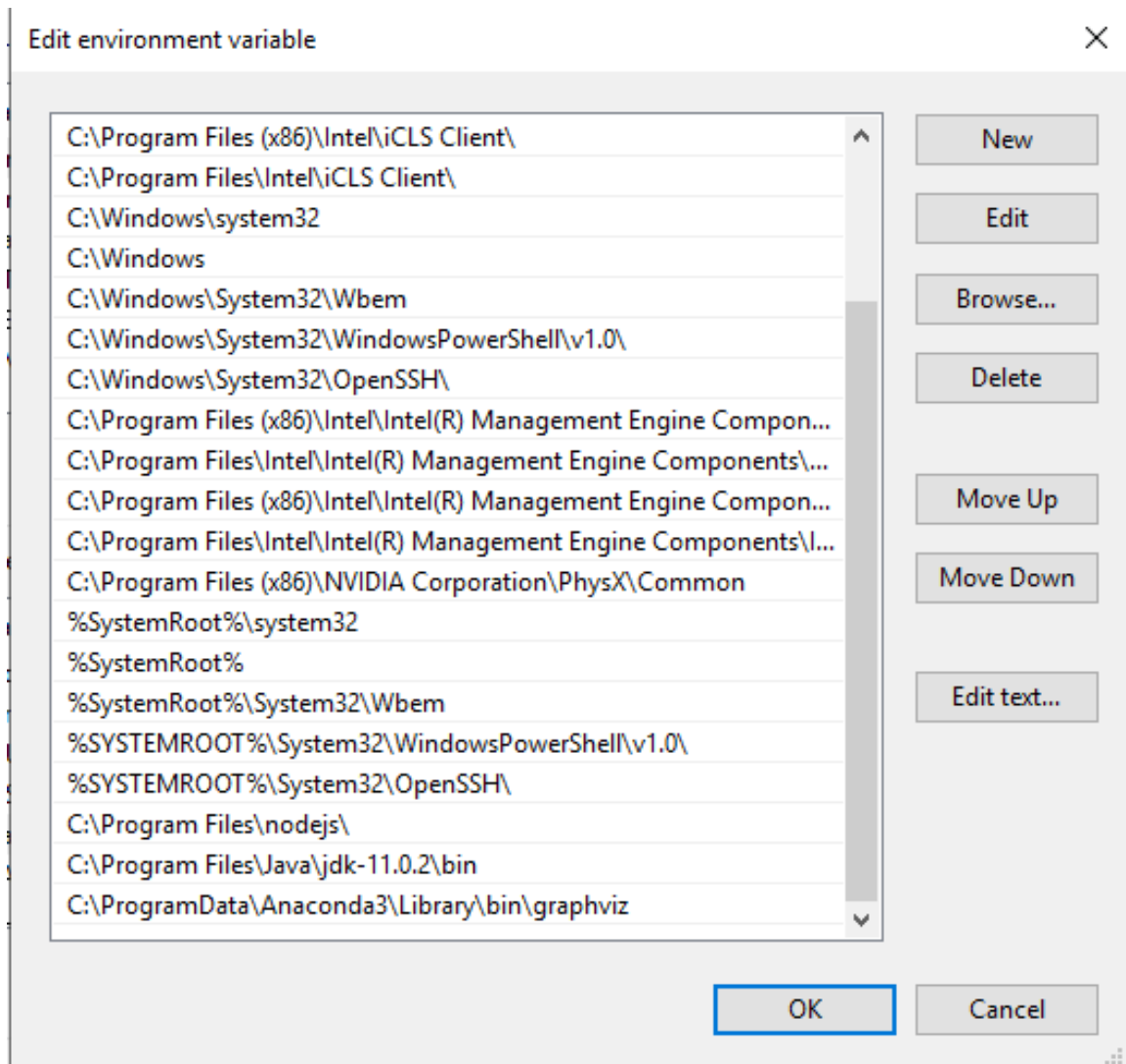


Figure 2.68: Resolve.

```
In [1]: import pandas as pd
...: d = pd.read_csv('E:\KAMPUS\Semester 6\Kecerdasan Buatan\modul
\Python-Artificial-Intelligence-Projects-for-Beginners\Chapter01\dataset
\student-mat.csv', sep=';')
...: len(d)
Out[1]: 395
```

Figure 2.69: Loading Dataset

```
In [8]: jeruk['pass'] = jeruk.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3']) >= 35
else 0, axis=1)
...: jeruk = jeruk.drop(['G1', 'G2', 'G3'], axis=1)
...: jeruk.head()
Out[8]:
  school sex  age address famsize ... Dalc  Walc  health absences pass
0    GP   F   18     U    GT3 ...    1    1    3         6      0
1    GP   F   17     U    GT3 ...    1    1    3         4      0
2    GP   F   15     U    LE3 ...    2    3    3        10      0
3    GP   F   15     U    GT3 ...    1    1    5         2      1
4    GP   F   16     U    GT3 ...    1    2    5         4      0

[5 rows x 31 columns]
```

Figure 2.70: Generate Binary Label

```
In [9]: jeruk = apel.get_dummies(jeruk, columns=['sex', 'school', 'address', 'famsize',
'Pstatus', 'Mjob', 'Fjob',
...:                                     'reason', 'guardian', 'schoolsup', 'famsup',
'paid', 'activities',
...:                                     'nursery', 'higher', 'internet', 'romantic'])
...: jeruk.head()
Out[9]:
   age  Medu  Fedu  ...  internet_yes  romantic_no  romantic_yes
0   18     4     4  ...              0             1              0
1   17     1     1  ...              1             1              0
2   15     1     1  ...              1             1              0
3   15     4     2  ...              1             0              1
4   16     3     3  ...              0             1              0

[5 rows x 57 columns]
```

Figure 2.71: One-hot Encoding

```
In [10]: jeruk = jeruk.sample(frac=1)
...: # split training and testing data
...: jeruk_train = jeruk[:500]
...: jeruk_test = jeruk[500:]
...:
...: jeruk_train_att = jeruk_train.drop(['pass'], axis=1)
...: jeruk_train_pass = jeruk_train['pass']
...:
...: jeruk_test_att = jeruk_test.drop(['pass'], axis=1)
...: jeruk_test_pass = jeruk_test['pass']
...:
...: jeruk_att = jeruk.drop(['pass'], axis=1)
...: jeruk_pass = jeruk['pass']
...:
...: # number of passing students in whole dataset:
...: import numpy as np
...: print("Passing: %d out of %d (%.2f%%)" % (np.sum(jeruk_pass), len(jeruk_pass),
100*float(np.sum(jeruk_pass)) / len(jeruk_pass)))
Passing: 166 out of 395 (42.03%)
```

Figure 2.72: Shuffle Rows

```
In [11]: from sklearn import tree
...: semangka = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: semangka = semangka.fit(jeruk_train_att, jeruk_train_pass)
```

Figure 2.73: Fit Decision Tree

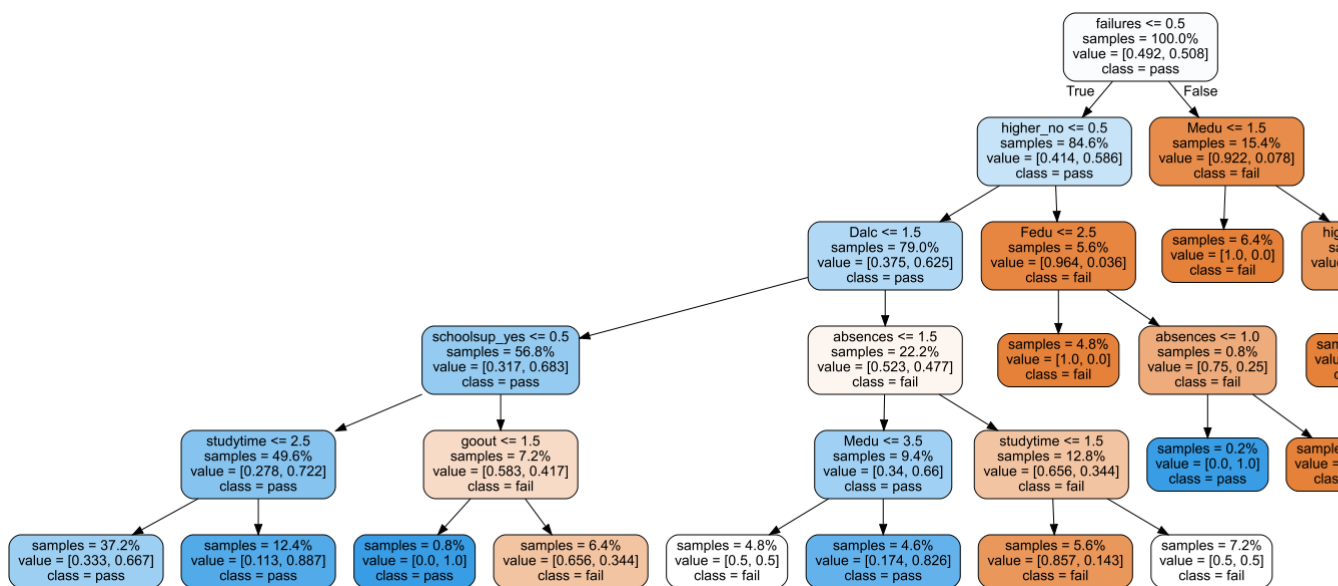


Figure 2.74: Fit Decision Tree

```
In [13]: tree.export_graphviz(semangka, out_file="student-
performance.dot", label="all", impurity=False, proportion=True,
...:     feature_names=list(jeruk_train_att),
class_names=["fail", "pass"],
...:     filled=True, rounded=True)
```

Figure 2.75: Fit Decision Tree

```
In [9]: semangka.score(jeruk_test_att, jeruk_test_pass)
Out[9]: 0.6845637583892618
```

Figure 2.76: Score

```
In [15]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(semangka, jeruk_att, jeruk_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))
Accuracy: 0.58 (+/- 0.04)
```

Figure 2.77: Cross Val Score

```
In [16]: for max_depth in range(1, 20):
...:     semangka = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...:     scores = cross_val_score(semangka, jeruk_att, jeruk_pass, cv=5)
...:     print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" % (max_depth, scores.mean(), scores.std() * 2))
Max depth: 1, Accuracy: 0.58 (+/- 0.01)
Max depth: 2, Accuracy: 0.58 (+/- 0.05)
Max depth: 3, Accuracy: 0.58 (+/- 0.02)
Max depth: 4, Accuracy: 0.59 (+/- 0.08)
Max depth: 5, Accuracy: 0.58 (+/- 0.03)
Max depth: 6, Accuracy: 0.58 (+/- 0.08)
Max depth: 7, Accuracy: 0.57 (+/- 0.10)
Max depth: 8, Accuracy: 0.60 (+/- 0.10)
Max depth: 9, Accuracy: 0.60 (+/- 0.09)
Max depth: 10, Accuracy: 0.62 (+/- 0.05)
Max depth: 11, Accuracy: 0.61 (+/- 0.08)
Max depth: 12, Accuracy: 0.63 (+/- 0.08)
Max depth: 13, Accuracy: 0.60 (+/- 0.12)
Max depth: 14, Accuracy: 0.59 (+/- 0.07)
Max depth: 15, Accuracy: 0.62 (+/- 0.07)
Max depth: 16, Accuracy: 0.59 (+/- 0.05)
Max depth: 17, Accuracy: 0.62 (+/- 0.05)
Max depth: 18, Accuracy: 0.62 (+/- 0.09)
Max depth: 19, Accuracy: 0.62 (+/- 0.07)
```

Figure 2.78: Max Depth

```
In [17]: depth_acc = np.empty((19,3), float)
...: i = 0
...: for max_depth in range(1, 20):
...:     semangka = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...:     scores = cross_val_score(semangka, jeruk_att, jeruk_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
Out[17]:
array([[1.00000000e+00, 5.79751704e-01, 6.30768599e-03],
[2.00000000e+00, 5.79817429e-01, 5.40865102e-02],
[3.00000000e+00, 5.82348264e-01, 2.23551177e-02],
[4.00000000e+00, 5.84752515e-01, 7.32490010e-02],
[5.00000000e+00, 5.84719247e-01, 4.05630974e-02],
[6.00000000e+00, 5.79462837e-01, 7.53158175e-02],
[7.00000000e+00, 5.71772152e-01, 9.89792479e-02],
[8.00000000e+00, 6.02088608e-01, 1.20171591e-01],
[9.00000000e+00, 6.02249270e-01, 9.20901855e-02],
[1.00000000e+01, 6.17666342e-01, 6.17102083e-02],
[1.10000000e+01, 6.02378286e-01, 5.13911736e-02],
[1.20000000e+01, 6.15198799e-01, 5.15762620e-02],
[1.30000000e+01, 6.05072217e-01, 5.81263071e-02],
[1.40000000e+01, 6.15038137e-01, 5.79996256e-02],
[1.50000000e+01, 5.97380721e-01, 6.43910242e-02],
[1.60000000e+01, 5.82190036e-01, 7.12462584e-02],
[1.70000000e+01, 5.97284972e-01, 8.33497031e-02],
[1.80000000e+01, 6.25166342e-01, 6.78027722e-02],
[1.90000000e+01, 5.97285784e-01, 1.01208800e-01]])
```

Figure 2.79: Depth in Range

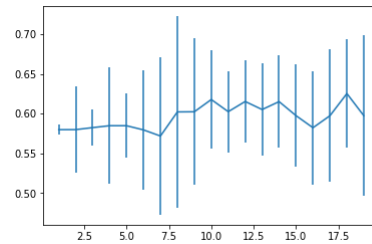


Figure 2.80: Matplotlib

```

...: 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
Traceback (most recent call last):
  File "<ipython-input-11-ef75b81d0d6a>", line 1, in <module>
    import graphviz
ModuleNotFoundError: No module named 'graphviz'

```

Figure 2.81: Error Graphviz

```

C:\Users\VACER>pip install graphviz
Collecting graphviz
  Downloading https://files.pythonhosted.org/packages/1f/e2/ef2581b5b866256
/graphviz-0.10.1-py2.py3-none-any.whl
Installing collected packages: graphviz
Successfully installed graphviz-0.10.1
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip'

```

Figure 2.82: install Graphviz

```

C:\Users\VACER>conda install pip
Collecting package metadata: done
Solving environment: done

# All requested packages already installed.

```

Figure 2.83: Solving Environment

```

C:\Users\ACER>conda install python-graphviz
Collecting package metadata: done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\ACER\Anaconda3

added / updated specs:
- python-graphviz

The following packages will be downloaded:

package | build | size
-----|-----|-----
graphviz-2.38 | hfa6e2cd_3 | 37.7 MB
python-graphviz-0.8.4 | py36_1 | 28 KB
-----|-----|-----
Total: | | 37.8 MB

The following NEW packages will be INSTALLED:

graphviz      pkgs/main/win-32::graphviz-2.38-hfa6e2cd_3
python-graphviz pkgs/main/win-32::python-graphviz-0.8.4-py36_1

Proceed ([y]/n)? y

```

Figure 2.84: Evaluasi Error

Chapter 3

Methods

3.1 The data

Please tell where is the data come from, a little brief of company can be put here.

3.2 Method 1

Definition, steps, algorithm or equation of method 1 and how to apply into your data

3.3 Method 2

Definition, steps, algorithm or equation of method 2 and how to apply into your data

3.4 Yusniar Nur Syarif Sidiq/1164089

1. Random Forest merupakan algoritma yang digunakan terhadapap klasifikasi data dalam jumlah yang besar. Klasifikasi pada random forest dilakukan dengan penggabungan dicision tree dengan melakuakn training terhadap sampel data yang dimiliki. Semakin banyak dicision tree maka data yang di dapat akan semakin akurat. Untuk gambar Random Forest dapat dilihat pada figure 3.1
2. Pertama download dataset terlebih dahulu lalu buka dengan menggunakan software spyder guna melihat isi dari dataset tersebut. Data tersebut memiliki extensi file bernama .txt dan didalamnya terdapat class dari field. Misalnya saja pada data jenis burung memiliki file index dan angka, dimana index berisi angka yang memiliki makna berupa jenis burung atau bahkan nama burung sedangkan field memiliki isi nilai berupa 0 dan 1 yang dimana sifatnya boolean

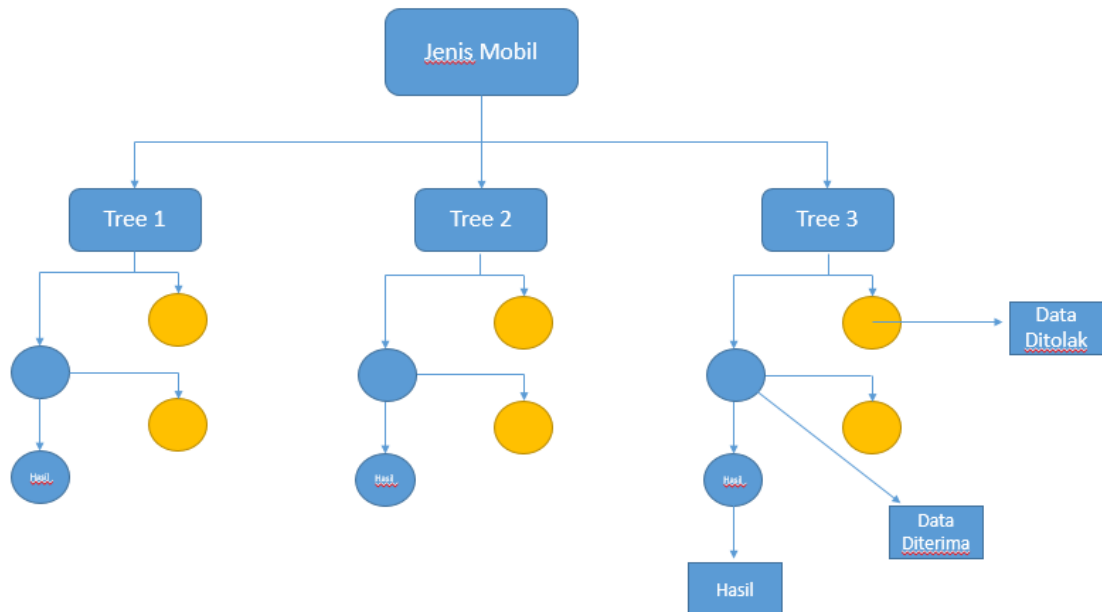


Figure 3.1: Random Forest.

atau Ya dan Tidak. Hal ini dikarenakan komputer hanya dapat membaca bilangan biner maka dari itu field yang di isikan berupa angka. Artinya angka 0 berarti tidak dan angka 1 berarti Ya.

3. Cross Validation adalah sebuah teknik validasi model yang digunakan untuk menilai bagaimana hasil analisis statistik akan digeneralisasi ke kumpulan data independen. Cross validation digunakan dengan tujuan prediksi, dan bila kita ingin memperkirakan seberapa akurat model prediksi yang dilakukan dalam sebuah praktek. Tujuan dari cross validation yaitu untuk mendefinisikan dataset guna menguji dalam fase pelatihan untuk membatasi masalah seperti overfitting dan underfitting serta mendapatkan wawasan tentang bagaimana model akan digeneralisasikan ke set data independen.
4. Dimana Score 44 % diperoleh dari hasil pengolahan dataset jenis burung. Dimana akan dilakukan proses pembagian data testing dan data training lalu diproses dan menghasilkan score sebanyak 44 % dimana menjelaskan bahwa score tersebut digunakan sebagai pembandingan dalam tingkat keakuratannya. Pada decision tree akan memperoleh data lebih kecil yaitu sebanyak 27 % hal ini dikarenakan data yang diolah menggunakan decision tree dibagi menjadi beberapa tree dan lalu disimpulkan untuk mendapatkan data yang akurat. Pada

SVM akan memperoleh score sebanyak 29 % hal ini dikarenakan data yang dimiliki masih bernilai netral sehingga tingkat keakuratannya masih belum jelas.

5. Untuk membaca confusion matriks dapat menggunakan source code sebagai berikut,

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

Dimana numpy akan mengurus semua data yang berhubungan dengan matrix. Pada source code tersebut digunakan dalam melakukan read pada dataset burung dengan menggunakan metode confusion matrix. Dalam confusion matrix memiliki 4 istilah yaitu True Positive yang merupakan data positif yang terditeksi benar, True Negatif yang merupakan data negatif akan tetapi terditeksi benar, False Positif merupakan data negatif namun terditeksi sebagai data positif, False Negatif merupakan data positif namun terditeksi sebagai data negatif. Adapun contoh hasil read dataset menggunakan confusion matrix dapat dilihat pada figure 3.2

		True Values	
		True	False
Prediction	True	TP Correct result	FP Unexpected result
	False	FN Missing result	TN Correct absence of result

Figure 3.2: Confusion Matrix.

6. Voting merupakan proses pemilihan dari tree yang dimana akan dimunculkan hasilnya dan disimpulkan menjadi informasi yang pasti. Untuk lebih jelasnya saya akan memberikan sebuah contoh bagaimana voting bekerja.

Dimana ditunjukkan pada figure 3.3 terdapat 3 tree. Dalam tree tersebut akan dilakukan proses voting. Saya akan memberikan contoh kasus, dimana akan

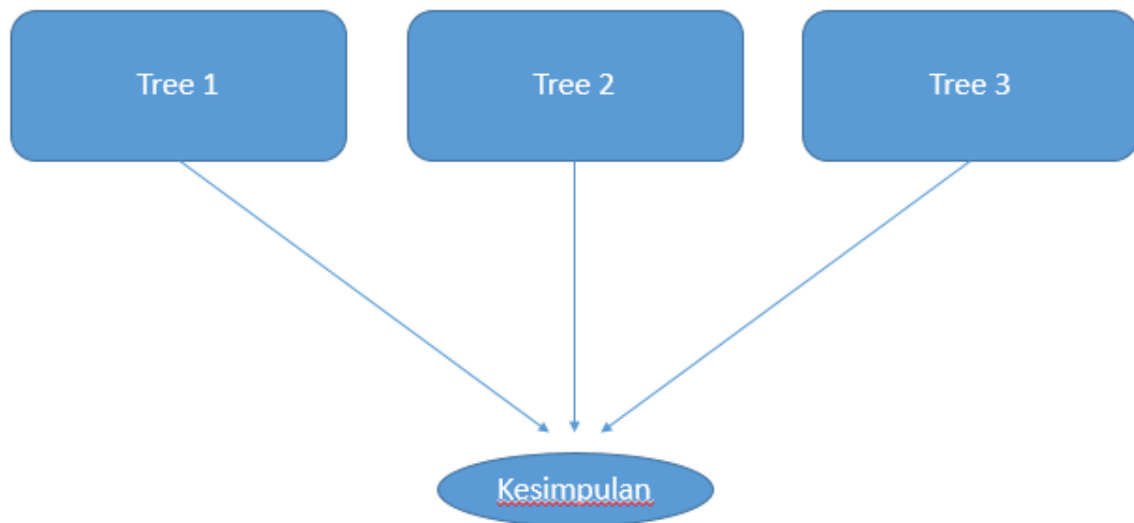


Figure 3.3: Voting.

diadakan voting untuk menentukan sebuah mobil. Dalam tree akan diberikan sejumlah data misalnya saja data tersebut berupa gambar, yang dimana data tersebut akan dipilih dengan cara voting. Hasil voting akhir dari setiap tree menunjukkan mobil jazz, yang berarti kesimpulan dari data yang telah diberikan menyatakan gambar tersebut adalah mobil jazz. Bagaimana apabila terjadi perbedaan data misalnya saja pada tree 1 dan 2 menyatakan mobil jazz sedangkan pada tree 3 menyatakan mobil yaris, maka kesimpulan yang di ambil adalah mobil jazz dikarenakan hasil voting terbanyak adalah mobil jazz.

3.5 Imron Sumadireja/1164076

3.5.1 Teori

1. Random Forest Beserta Ilustrasinya

Random Forest adalah salah satu algoritma yang digunakan pada klasifikasi data dalam jumlah yang besar. Klasifikasi random forest ini dilakukan melalui penggabungan decision tree dengan melakukan training pada sampel data yang dimiliki atau biasa disebut dengan supervised learning. Semakin banyak menggunakan decision tree maka akan mempengaruhi akurasi yang didapatkan menjadi lebih baik. Setiap decision tree memiliki atribut yang berbeda, serta decision tree tersebut spesifik terhadap atributnya yang merupakan bagian kecil

dari keseluruhan atribut pada data set. Contoh sederhanaanya bisa dilihat pada gambar berikut 3.4

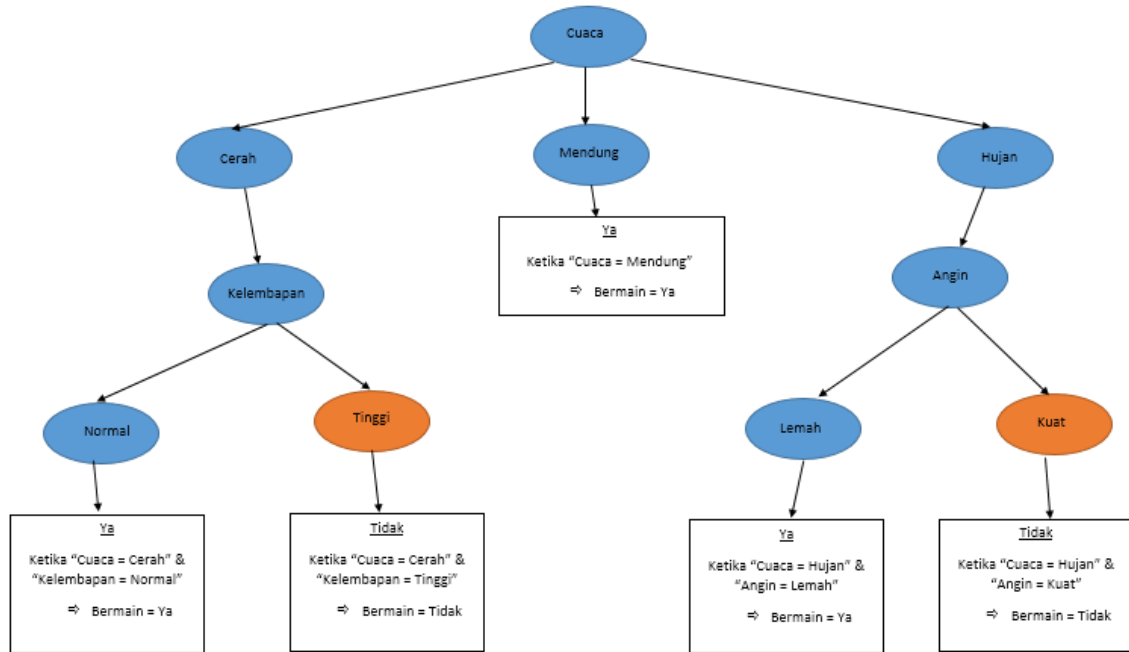


Figure 3.4: Random Forest.

2. Membaca Dataset, Makna Setiap File Serta Field Masing-Masing File

Pertama download terlebih dahulu datasetnya kemudian buka menggunakan spyder untuk mengetahui isi dari dataset tersebut. Untuk menjalankan code tersebut tinggal blok bagian yang akan di jalankan. Dataset tersebut di dalam-nya terdapat class dari field atau data. Sebagai contoh pada data burung terdapat field index dan angka, untuk index biasanya berupa angka, angka tersebut memiliki makna sebagai pengganti nama atau jenis burung. Sedangkan field berisi nilai 0 dan 1 maknanya untuk memberikan penilaian ya atau tidak pada setiap suatu data namun pada kasus ini field di ganti dari ya atau tidak menjadi 0 dan 1 karena komputer kesulitan membaca ya atau tidak dan hanya bisa membaca dengan 0 dan 1 saja.

3. Cross Validation

Cross validation adalah metode statistik yang dapat digunakan untuk mengevaluasi kinerja model atau algoritma dengan data dipisahkan menjadi dua subset

yaitu data testing dan data training. Selain itu cross validation digunakan untuk memperkirakan seberapa akurat sebuah model prediktif ketika dijalankan. Untuk melakukan proses cross validation ini dibutuhkan sebuah data. Cross validation mengambil data dari output yang telah di eksekusi oleh algoritma sebelumnya. Hasil tersebut akan dipisahkan menjadi dua subset berdasarkan ukuran dataset. Selanjutnya dataset tersebut akan di test secara bergantian hingga seluruh bagian terpenuhi.

4. Arti Score 44% Pada Random Forest, 27% Pada Decision Tree dan 29% Dari SVM

(a) Arti Score 44% Pada Random Forest

Score tersebut merupakan hasil prediksi dari data yang telah dieksekusi sebelumnya dengan algoritma random forest, score tersebut menandakan bahwa akurasi yang didapatkan tidak terlalu baik karena data yang diujinya cukup banyak. Tetapi itu jauh lebih baik daripada menebak secara acak.

(b) Arti Score 27% Pada Decision Tree

Score tersebut merupakan hasil prediksi dari data yang dieksekusi sebelumnya dengan algoritma decision tree, selain itu pada decision tree menggunakan library sklearn sebagai acuan untuk melakukan prediksi. Untuk decision tree ini hasil yang didapatkan ialah 27%. Hasil tersebut lebih buruk dibandingkan dengan menggunakan algoritma random forest.

(c) Arti Score 29% Pada SVM

Score tersebut merupakan hasil prediksi dari data yang dieksekusi sebelumnya dengan algoritma Support Vector Machine, score tersebut lebih baik daripada hasil yang di prediksi oleh decision tree namun score yang dimiliki oleh SVM tidak lebih baik dari hasil random forest.

5. Cara Membaca Confusion Matriks Beserta Ilustrasinya

Cara untuk membaca confusion matriks yakni dengan cara memasukan parameter nilai yang tersedia pada datasets. Data tersebut akan menghasilkan 0.5, 0.2 dan lain seterusnya sampai mendekati angka 1 atau akurasi yang sempurna. Pada confusion matriks terdapat 4 istilah sebagai representasi hasil proses klasifikasi, seperti gambar berikut 3.5

		True Class	
		Positive	Negative
Predicted Class	Positive	true positives count (TP)	false negatives count (FP)
	Negative	false positives count (FN)	true negatives count (TN)

Figure 3.5: Confusion Matrix.

6. Jelaskan Voting Pada Random Forest Beserta Ilustrasinya

Voting pada random forest berguna untuk mengambil nilai pada masing-masing tree yang akan digunakan untuk menentukan hasil final dengan akurasi yang lebih baik. Untuk ilustrasi sederhananya sebagai berikut 3.6

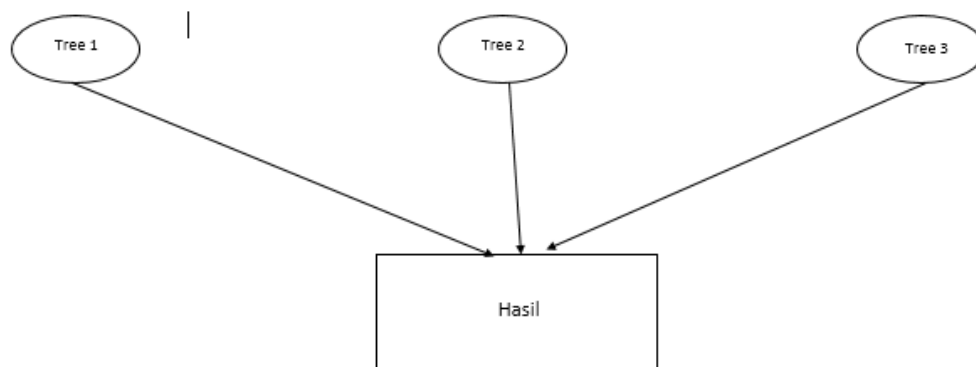


Figure 3.6: Voting.

3.6 Andri Fajar Sunandhar/1164065

3.6.1 Teori

1. Apa itu Random Forest Serta Gambar Ilustrasinya

Random Forest adalah suatu algoritma yang digunakan pada klasifikasi data dalam jumlah yang besar. Klasifikasi random forest dilakukan melalui penggabungan pohon dengan melakukan training pada sampel data yang dimiliki. Penggunaan tree yang semakin banyak akan mempengaruhi akurasi yang akan didapatkan menjadi lebih baik. Penentuan klasifikasi dengan random forest diambil berdasarkan hasil voting dari pohon yang terbentuk. Pemenang dari pohon yang terbentuk ditentukan dengan vote terbanyak. Pembangunan pohon pada random forest sampai dengan mencapai ukuran maksimum dari pohon data. Akan tetapi, pembangunan pohon Random Forest tidak dilakukan pemangkasan yang merupakan sebuah metode untuk mengurangi kompleksitas ruang. Contoh Ilustrasi sederhana Gambar Random Forest.

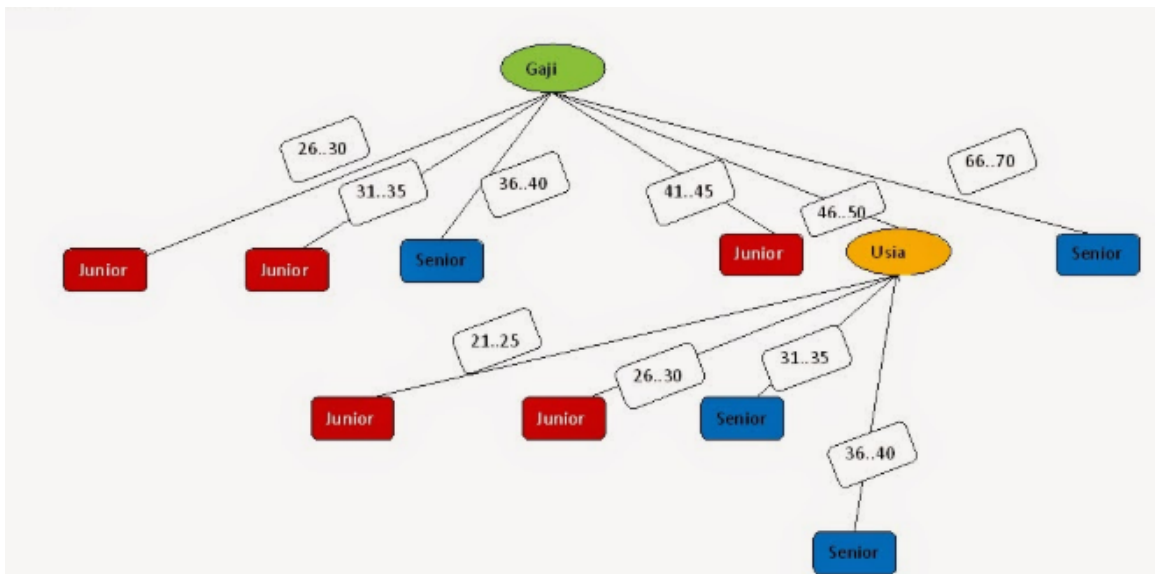


Figure 3.7: Random Forest.

2. Cara Membaca Dataset

- Buka Anaconda Navigator.
- Jalankan Spyder
- Import libraries yang dibutuhkan

```

16
17 dataset = pd.read_csv('Data.csv')
18 |

```

Figure 3.8: Kode membaca file.csv

- (d) Masukkan kode berikut untuk membaca file Data.csv.
- (e) Jalankan kode tersebut, maka di windiws console akan muncul pesan :

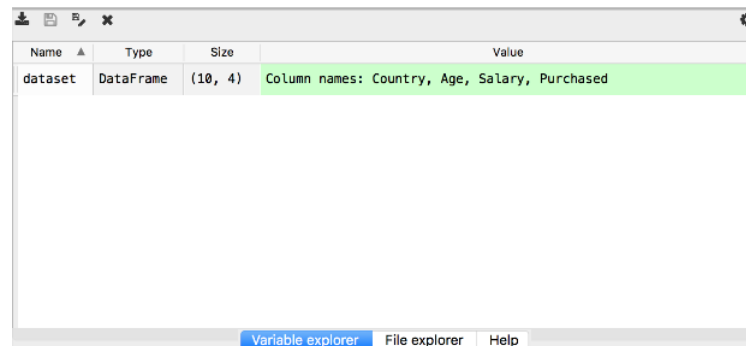
```

In [6]: dataset = pd.read_csv('Data.csv')
In [7]:

```

Figure 3.9: Window Console

- (f) Klik variable explorer, maka akan terlihat dataset yang baru ter-import.



Name ▲	Type	Size	Value
dataset	DataFrame	(10, 4)	Column names: Country, Age, Salary, Purchased

Figure 3.10: Variable Explorer

- (g) Kemudian double klik pada dataset cell, maka akan muncul pop-up windows seperti berikut:
- (h) Seperti yang terlihat pada gambar tersebut dataset ini memiliki Kolom Country, Age, dan Salary sebagai independent variable-nya dan kolom Purchased sebagai dependent variable-nya.

3. Cross Validation

Cross validation adalah metode statistik yang digunakan untuk memperkirakan keterampilan model pembelajaran mesin. Ini biasanya digunakan dalam pembelajaran mesin yang diterapkan untuk membandingkan dan memilih model untuk masalah pemodelan prediktif yang diberikan karena mudah dipahami, mudah

Index	Country	Age	Salary	Purchased
0	France	44	72000	No
1	Spain	27	48000	Yes
2	Germany	30	54000	No
3	Spain	38	61000	No
4	Germany	40	nan	Yes
5	France	35	58000	Yes
6	Spain	nan	52000	No
7	France	48	79000	Yes
8	Germany	50	83000	No
9	France	37	67000	Yes

Figure 3.11: Dataset Cell

diimplementasikan, dan menghasilkan estimasi keterampilan yang umumnya memiliki bias lebih rendah daripada metode lainnya.

4. Arti Score 44% Pada Random Forest, 27% Pada Decision Tree dan 29% Dari SVM

- (a) Arti Score 44%

Pada Random Forest, Score tersebut merupakan hasil dari akurasi.

- (b) Arti Score 27%

Pada decision tree adalah presentasi hasil dari perhitungan dataset.

- (c) Arti Score 29% Pada SVM

merupakan hasil pendekatan jaringan saraf. Jaringan saraf sendiri merupakan komponen jaringan utama dari sistem saraf. Sistem tersebut mengatur dan mengontrol fungsi tubuh dan aktivitas dan terdiri dari dua bagian: (SSP) yang terdiri dari otak dan sumsum tulang belakang, dan percabangan saraf perifer dari sistem saraf tepi (SST) yang terdapat dalam pengolahan dataset terkait.

- (a) Confusion Matrix Dan Ilustrasinya

- i. Perhitungan confusion matrix adalah sebagai berikut, akan saya beri contoh sederhana yaitu pengambilan keputusan untuk mendapatkan bantuan beasiswa. Saya menggunakan dua atribut, yaitu rekening listrik dan gaji. Ini adalah pohon keputusannya:

Kemudian data testingnya adalah

Yang pertama kita lakukan yaitu mencari 4 nilai yaitu a,b,c, dan d:

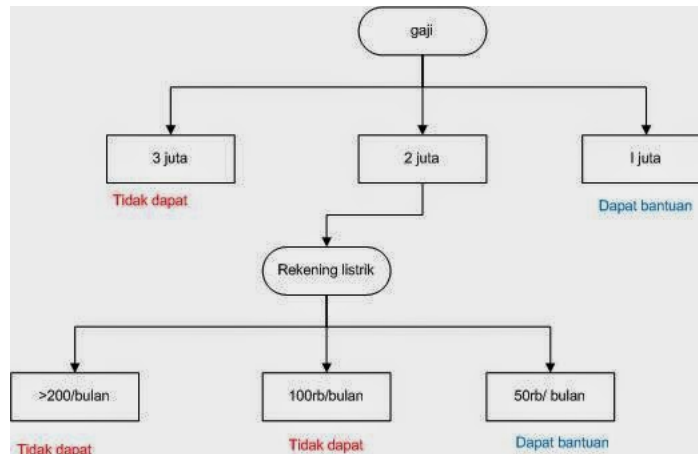


Figure 3.12: Pohon Keputusan

no	nama	gaji	rekening	hasil	kecocokan
1	Aji	3 juta	100rb/bulan	dapat bantu	t
2	Ali	1 juta	50rb/bulan	dapat bantu	y
3	Amar	2 juta	100rb/bulan	tidak dapat	y
4	Bastoni	1 juta	100rb/bulan	tidak dapat	y
5	Tolib	2 juta	50rb/bulan	dapat bantu	y
6	Sarip	3 juta	>200rb/bulan	tidak dapat	y
7	Tuwar	3 juta	100rb/bulan	tidak dapat	y
8	Rokip	2 juta	100rb/bulan	tidak dapat	y
9	Habib	1 juta	100rb/bulan	dapat bantu	y
10	Sohe	2 juta	50rb/bulan	tidak dapat	t

Figure 3.13: Data Testing

$$a = 5$$

$$b = 1$$

$$c = 1$$

$$d = 3$$

Kemudian kita dapat mencari nilai Recall, Precision, accuracy dan Error Rate

$$\text{Recall} = 3 / (1 + 3) = 0,75$$

$$\text{Precision} = 3 / (1 + 3) = 0,75$$

$$\text{Accuracy} = (5 + 3) / (5 + 1 + 1 + 3) = 0,8$$

$$\text{Error Rate} = (1 + 1) / (5 + 1 + 1 + 3) = 0,2$$

5. Jelaskan Voting Pada Random Forest Beserta Ilustrasinya

Voting merupakan metode yang paling umum digunakan dalam random forest. Ketika classifier membuat keputusan, Anda dapat memanfaatkan yang terbaik keputusan umum dan rata-rata yang didefinisikan ke dalam bentuk "voting".

Setelah pohon terbentuk, maka akan dilakukan voting pada setiap kelas dari data sampel. Kemudian, mengkombinasikan vote dari setiap kelas kemudian diambil vote yang paling banyak. Dengan menggunakan random forest pada klasifikasi data maka, akan menghasilkan vote yang paling baik. 3.14

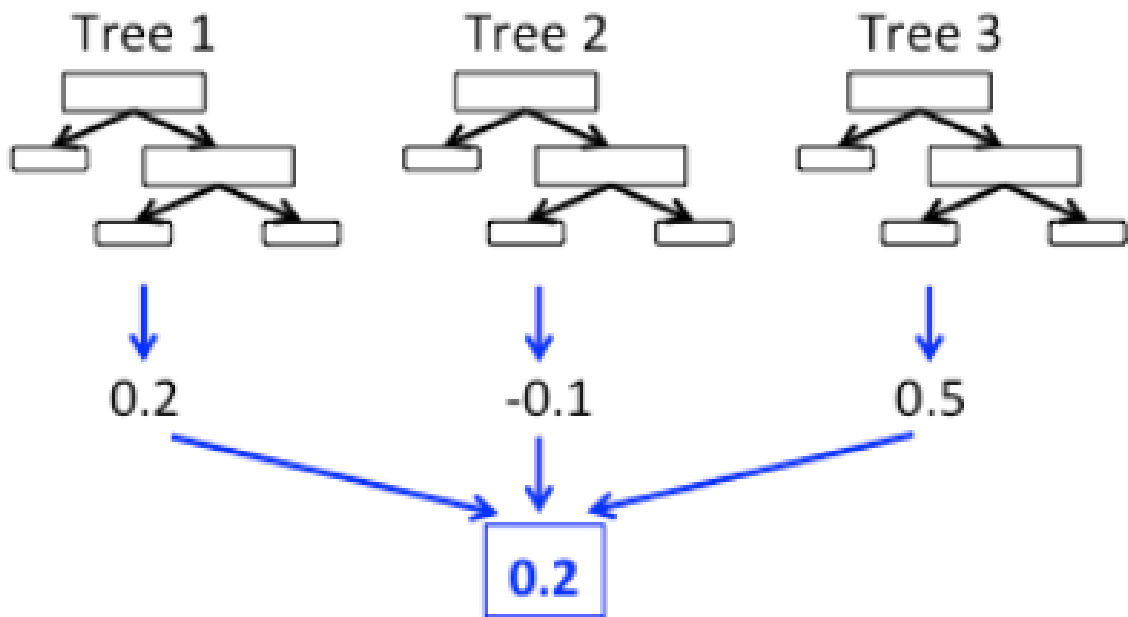


Figure 3.14: Voting.

Chapter 4

Experiment and Result

brief of experiment and result.

4.1 Experiment

Please tell how the experiment conducted from method.

4.2 Result

Please provide the result of experiment

Chapter 5

Conclusion

brief of conclusion

5.1 Conclusion of Problems

Tell about solving the problem

5.2 Conclusion of Method

Tell about solving using method

5.3 Conclusion of Experiment

Tell about solving in the experiment

5.4 Conclusion of Result

tell about result for purpose of this research.

Chapter 6

Discussion

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

Chapter 7

Discussion

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

Chapter 8

Discussion

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

Chapter 9

Discussion

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

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] Abdillah Baraja. Kecerdasan buatan tinjauan historikal. *Speed-Sentra Penelitian Engineering dan Edukasi*, 1(1), 2008.
- [2] Youssef Bassil. Expert pc troubleshooter with fuzzy-logic and self-learning support. *arXiv preprint arXiv:1204.0181*, 2012.
- [3] 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.
- [4] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.
- [5] Kevin Warwick. *Artificial intelligence: the basics*. Routledge, 2013.