

Tugas

Kecerdasan Buatan



Alvian Daniel Sinaga
1184077

Applied Bachelor of Informatics Engineering
Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering
Politeknik Pos Indonesia

Bandung 2021

‘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
1.1.1	Definisi Kecerdasan Buatan	1
1.1.2	Sejarah dan Perkembangan Kecerdasan Buatan	2
1.1.3	Supervised Learning	5
1.1.4	Klasifikasi	6
1.1.5	Regresi	7
1.1.6	Unsupervised Learning	7
1.1.7	Data Set	9
1.1.8	Training Set	9
1.1.9	Testing Set	9
1.2	Praktek	9
1.3	Penanganan Error	15
1.3.1	Bukti Tidak Plagiat	17
	Bibliography	18

List of Figures

1.1	Supervised Learning.	6
1.2	Unsupervised Learning.	8
1.3	Instalasi Library Scikit-Learn.	9
1.4	Variabel Explorer Library Scikit-Learn.	12
1.5	Hasil Loading an Example Dataset.	12
1.6	Hasil Learning and Predicting.	13
1.7	Hasil Model Persistence.	14
1.8	Hasil Conventions.	15
1.9	Name Error.	15
1.10	Import Error.	16
1.11	Syntax Error.	16
1.12	Bukti Tidak Plagiat.	17

Chapter 1

Mengenai Kecerdasan Buatan dan Scikit-Learn

1.1 Teori

1.1.1 Definisi Kecerdasan Buatan

Kecerdasan buatan atau artificial intelligence (AI) menurut beberapa pakar adalah sebagai berikut:

1. Schalkoff (1990): AI adalah bidang studi yang mencoba meniru dan men-erangkan perilaku cerdas yang dilakukan oleh manusia dalam bentuk proses komputasi.
2. Rich dan Knight (1991): AI adalah studi tentang cara membuat komputer melakukan sesuatu yang sampai saat ini orang dapat melakukannya lebih baik.
3. Luger dan Stubblefield (1993): AI adalah cabang ilmu komputer yang berhubungan dengan otomasi perilaku yang cerdas.

Jadi dapat disimpulkan kecerdasan buatan atau artificial intelligence (AI) merupakan salah satu bagian ilmu komputer yang membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Seperti yang kita tahu pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi sekedar digunakan sebagai alat hitung namun diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.

1.1.2 Sejarah dan Perkembangan Kecerdasan Buatan

Istilah AI pertama kali dikemukakan pada tahun 1956 di konferensi Dartmouth. Sejak saat itu AI terus dikembangkan sebab berbagai penelitian mengenai teori-teori dan prinsip-prinsipnya juga terus berkembang. Meskipun istilah AI baru muncul tahun 1956, tetapi teori-teori mengarah ke AI sudah muncul sejak tahun 1941. Berikut ini tahapan-tahapan sejarah perkembangan AI:

1. Era Komputer Elektronik (1941)

Pada tahun 1941 telah ditemukan alat penyimpanan dan pemrosesan informasi. Penemuan tersebut dinamakan komputer elektronik yang dikembangkan di USA dan Jerman. Komputer pertama ini memerlukan ruangan yang luas dan ruang AC yang terpisah. Saat itu komputer melibatkan konfigurasi ribuan kabel untuk menjalankan suatu program. Hal ini sangat merepotkan bagi para programmer. Pada tahun 1949, berhasil dibuat komputer yang mampu menyimpan program sehingga membuat pekerjaan untuk memasukkan program menjadi lebih mudah. Penemuan ini menjadi dasar pengembangan program yang mengarah ke AI.

2. Masa Persiapan AI (1943–1956)

Pada tahun 1943, Warren McCulloch dan Walter Pitts mengemukakan tiga hal: pengetahuan fisiologi dasar dan fungsi sel syaraf dalam otak, analisis formal tentang logika proposisi (propositional logic), dan teori komputasi Turing. Mereka berhasil membuat suatu model syaraf tiruan (artificial neuron) di mana setiap neuron digambarkan sebagai on dan off. Mereka menunjukkan bahwa setiap fungsi dapat dihitung dengan suatu jaringan sel syaraf dan bahwa semua hubungan logis dapat diimplementasikan dengan struktur jaringan yang sederhana. Pada tahun 1950, Norbert Wiener membuat penelitian mengenai prinsip-prinsip teori feedback. Contoh yang terkenal adalah thermostat. Penemuan ini juga merupakan awal perkembangan AI. Pada tahun 1956, John McCarthy (yang setelah lulus dari Princeton kemudian melanjutkan ke Dartmouth College) meyakinkan Minsky, Claude Shannon dan Nathaniel Rochester untuk membantunya melakukan penelitian dalam bidang Automata, jaringan sel syaraf dan pembelajaran intelektual. Mereka mengerjakan proyek ini selama dua bulan di Dartmouth. Hasilnya adalah program yang mampu ber-pikir non-numerik dan menyelesaikan masalah pemikiran, yang dinamakan Principia

Mathematica. Hal ini menjadikan McCarthy disebut sebagai Father of AI (Bapak AI).

3. Awal Perkembangan AI (1952–1969)

Pada tahun-tahun pertama pengembangannya, AI mengalami banyak kesuksesan. Diawali dengan kesuksesan Newell dan Simon dengan sebuah program yang disebut General Problem Solver. Program ini dirancang untuk memulai penyelesaian masalah secara manusiawi. Pada tahun 1958, McCarthy di MTT Lab Memo No. 1 mendefinisikan bahasa pemrograman tingkat tinggi yaitu LISP, yang sekarang mendominasi pembuatan program-program AI. Kemudian, McCarthy membuat program yang dinamakan Programs With Common Sense. Di dalam program tersebut, dibuat rancangan untuk menggunakan pengetahuan dalam mencari solusi. Pada tahun 1959, Nathaniel Rochester dari IBM dan mahasiswa-mahasiswanya mengeluarkan program AI Geometry Theorem Prover. Program ini dapat membuktikan suatu teorema menggunakan aksioma-aksioma yang ada. Pada tahun 1963, program yang dibuat James Slagle mampu menyelesaikan masalah integral tertutup untuk mata kuliah kalkulus. Pada tahun 1968, program analogi buatan Tom Evan menyelesaikan masalah analogi geometris yang ada pada tes IQ.

4. Perkembangan AI Melambat (1966–1974)

Prediksi Herbert Simon pada tahun 1957 yang menyatakan bahwa AI akan menjadi ilmu pengetahuan yang akan berkembang dengan pesat ternyata meleset. Pada 10 tahun kemudian, perkembangan AI melambat. Hal ini disebabkan adanya 3 kesulitan utama yang dihadapi AI, yaitu:

- (a) Masalah pertama: program-program AI yang bermunculan hanya mengandung sedikit atau bahkan tidak mengandung sama sekali pengetahuan (knowledge) pada subjeknya. Program-program AI berhasil hanya karena manipulasi sintetis yang sederhana. Sebagai contoh adalah Weizenbaum's ELIZA program (1965) yang dapat melakukan percakapan serius pada berbagai topik, sebenarnya hanyalah peminjaman dan manipulasi kalimat-kalimat yang diketikkan oleh manusia.
- (b) Masalah kedua: banyak masalah yang harus diselesaikan oleh AI. Karena terlalu banyaknya masalah yang berkaitan, maka tidak jarang banyak terjadi kegagalan pada pembuatan program AI.

(c) Masalah ketiga: ada beberapa batasan pada struktur dasar yang digunakan untuk menghasilkan perilaku intelegensia. Sebagai contoh adalah pada tahun 1969 buku Minsky dan Papert Perceptrons membuktikan bahwa program-program perceptrons dapat mempelajari segala sesuatu, tetapi program-program tersebut hanya mempresentasikan sejumlah kecil saja. Sebagai contoh dua masukan perceptrons yang berbeda tidak dapat dilatihkan untuk mengenali kedua masukan yang berbeda tersebut.

5. Sistem Berbasis Pengetahuan (1969–1979)

Pengetahuan adalah kekuatan pendukung AI. Hal ini dibuktikan dengan program yang dibuat oleh Ed Feigenbaum, Bruce Buchanan dan Joshua Lederberg yang membuat program untuk memecahkan masalah struktur molekul dari informasi yang didapatkan dari spectrometer massa. Program ini dinamakan Den-dral programs yang berfokus pada segi pengetahuan kimia. Dari segi diagnosis media juga sudah ada yang menemukannya, yaitu Saul Amarel dalam proyek computer in biomedicine. Proyek ini diawali dari keinginan untuk mendapatkan diagnosa penyakit berdasarkan pengetahuan yang ada pada mekanisme penyebab proses penyakit.

6. AI Menjadi Sebuah Industri (1980–1988)

Industrialisasi AI diawali dengan ditemukannya expert system (sistem pakar) yang dinamakan R1 yang mampu mengkonfigurasi sistem-sistem komputer baru. Program tersebut mulai dioperasikan di Digital Equipment Corporation(DEC), McDermott, pada tahun 1982. Pada tahun 1986, program ini telah berhasil menghemat US\$40 juta per tahun. Pada tahun 1988, kelompok AI di DEC menjalankan 40 sistem pakar. Hampir semua perusahaan besar di USA mempunyai divisi AI sendiri yang menggunakan ataupun mempelajari sistem pakar. Booming industri AI ini juga melibatkan perusahaan-perusahaan besar seperti Carnegie Group, Inference, Intellicorp, dan Technoledge yang menawarkan software tools untuk membangun sistem pakar. Perusahaan hardware seperti LISP dan Machines Inc., Texas Instruments, Symbolics, dan Xerox juga turut berperan dalam membangun workstation yang dioptimasi untuk pembangunan program LISP. Sehingga, perusahaan yang sejak tahun 1982 hanya menghasilkan beberapa juta US dolar per tahun meningkat menjadi 2 milyar US dolar per tahun pada tahun 1988.

7. Kembalinya Jaringan Syaraf Tiruan (1986–Sekarang)

Meskipun bidang ilmu komputer menolak jaringan syaraf tiruan setelah diterbitkan buku "perceptrons" karangan Minsky dan Papert, tetapi para ilmuwan masih mempelajari bidang ilmu tersebut dari sudut pandang yang lain yaitu fisika. Para ahli seperti Hopfield (1982) menggunakan teknik-teknik mekanika statistika untuk menganalisa sifat-sifat penyimpanan dan optimasi pada jaringan syaraf. Para ahli psikologi, David Rumelhart dan Geoff Hinton, melanjutkan penelitian mengenai model syaraf pada memori. Pada tahun 1985-an sedikitnya empat kelompok riset menemukan kembali algoritma belajar propagasi balik (Back-Propagation Learning). Algoritma ini berhasil diimplementasikan kedalam bidang ilmu komputer dan psikologi.

1.1.3 Supervised Learning

Supervised Learning adalah pembelajaran yang memiliki label di tiap datanya. Label maksudnya adalah tag dari data yang ditambahkan dalam machine learning model. Contohnya gambar kucing di tag "kucing" di tiap masing masing image kucing dan gambar anjing di tag "anjing" di tiap masing gambar anjing. Machine learning kategori dapat berupa clasification ("anjing", "kucing", "beruang", dsb) dan regression (berat badan, tinggi badan dsb). Supervised learning banyak digunakan dalam memprediksi pola dimana pola tersebut sudah ada contoh data yang lengkap, jadi pola yang terbentuk adalah hasil pembelajaran data lengkap tersebut. Tentunya jika kita memasukan data baru, setelah kita melakukan ETL (Extract Transform Load) maka kita mendapat info feature feature dari sample baru tersebut. Kemudian dari feature feature tersebut di compare dengan pattern clasification dari model yang didapat dari labeled data. Setiap label akan dicompare sampai selesai, dan yang memiliki percentage lebih banyak akan diambil sebagai prediksi akhir.

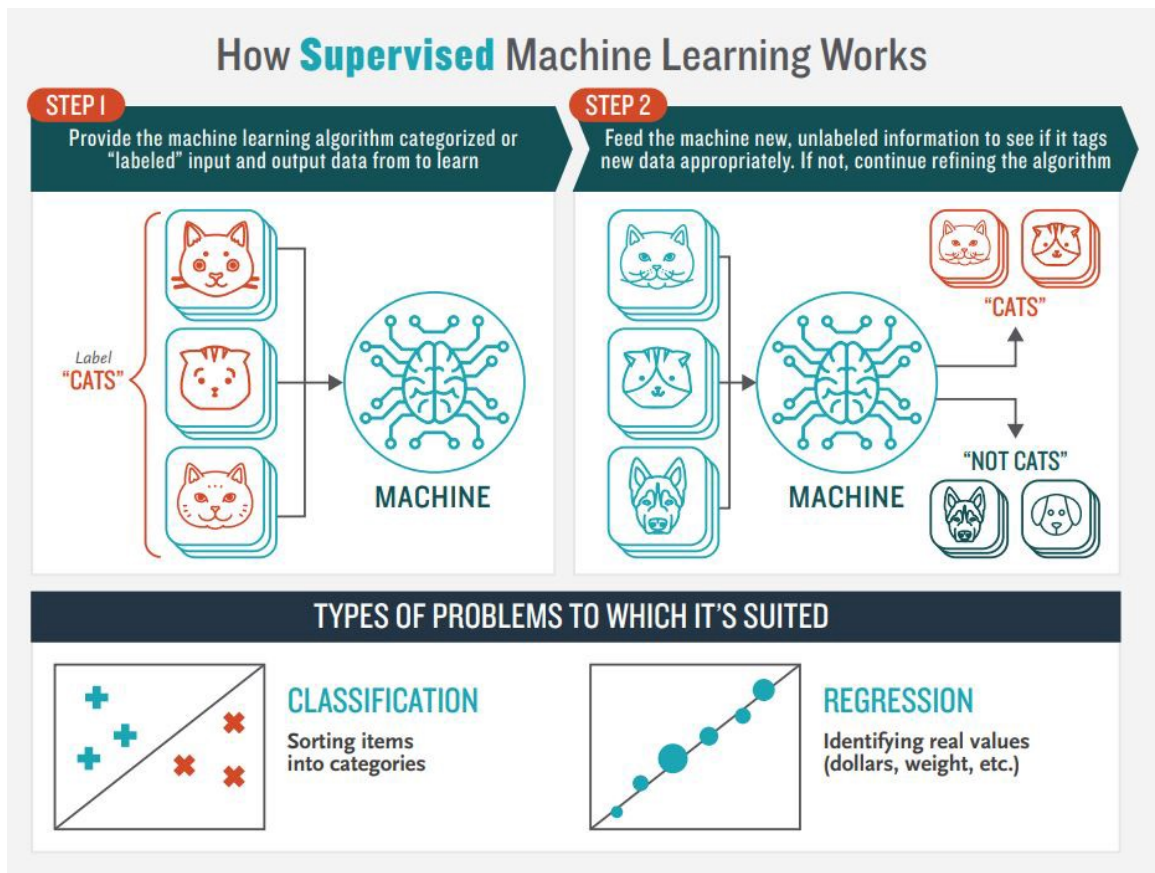


Figure 1.1: Supervised Learning.

Contoh algoritma yang digunakan pada supervised learning meliputi :

1. Clasification (Categorical) and Regression (Numerical)
2. Logistic Regression
3. Model Ensemble
4. Time series

1.1.4 Klasifikasi

Classification adalah tindakan untuk memberikan kelompok pada setiap keadaan. Setiap keadaan berisi sekelompok atribut, salah satunya adalah class attribute. Metode ini butuh untuk menemukan sebuah model yang dapat menjelaskan class attribute itu sebagai fungsi dari input attribute.

Class adalah attribute CollegePlans yang berisi dua pernyataan, Yes dan No, perhatikan ini. Sebuah Classification Model akan menggunakan atribut lain dari kasus tersebut (input attribut; yaitu kolom IQ, Gender, ParentIncome, dan ParentEncouragement) untuk dapat menentukan pola (pattern) class (Output Attribute; yaitu Kolom CollegePlans yang berisi Yes atau No). Algoritma Data Mining yang membutuhkan variabel target untuk belajar (sampai mendapatkan rule / pola yang berlaku pada data tersebut) kita standarkan dengan sebutan dengan Supervised Algorithm. Nah, yang termasuk kepada Classification Algorithm adalah Decision Trees, Neural Network dan Naives Bayes.

1.1.5 Regresi

Metode Regression mirip dengan metode Classification, yang membedakannya adalah metode regression tidak bisa mencari pola yang dijabarkan sebagai class (kelas). Metoda regression bertujuan untuk mencari pola dan menentukan sebuah nilai numerik. Sebuah Teknik Linear Line-fitting sederhana adalah sebuah contoh dari Regression, dimana hasilnya adalah sebuah fungsi untuk menentukan hasil yang berdasarkan nilai dari input. Bentuk yang lebih canggih dari regression sudah mendukung input berupa kategori, jadi tidak hanya input berupa numerik. Teknik paling populer yang digunakan untuk regression adalah linear regression dan logistic regression. Teknik lain yang didukung oleh SQL Server Data mining adalah Regression Trees (bagian dari algoritma Microsoft Decision Trees) dan Neural Network. Regression digunakan untuk memecahkan banyak problem bisnis – contohnya untuk memperkirakan metode distribusi, kapasitas distribusi, musim dan untuk memperkirakan kecepatan angin berdasarkan temperatur, tekanan udara, dan kelembaban.

1.1.6 Unsupervised Learning

Unsupervised learning memiliki keunggulan dari supervised learning. Jika supervised learning memiliki label sebagai dasar prediksi baik serta membuat clasification dan regression algorithm memungkinkan. Tetapi dalam realitanya, data real itu banyak yang tidak memiliki label. Label kebanyakan jika data sudah masuk ke ERP apapun bentuk ERPnya dan bagaimana kalo datanya berupa natural input seperti suara, gambar, dan video. Unsupervised learning tidak menggunakan label dalam memprediksi target feautres / variable. Melainkan menggunakan ke samaan dari at-

tribut attribut yang dimiliki. Jika attribut dan sifat-sifat dari data data feature yang diekstrak memiliki kemirip miripan, maka akan dikelompok kelompokan (clustering). Sehingga hal ini akan menimbulkan kelompok kelompok (cluster). Jumlah cluster bisa unlimited. Dari kelompok kelompok itu model melabelkan, dan jika data baru mau di prediksi, maka akan dicocok kan dengan kelompok yang mirip mirip featurenya.

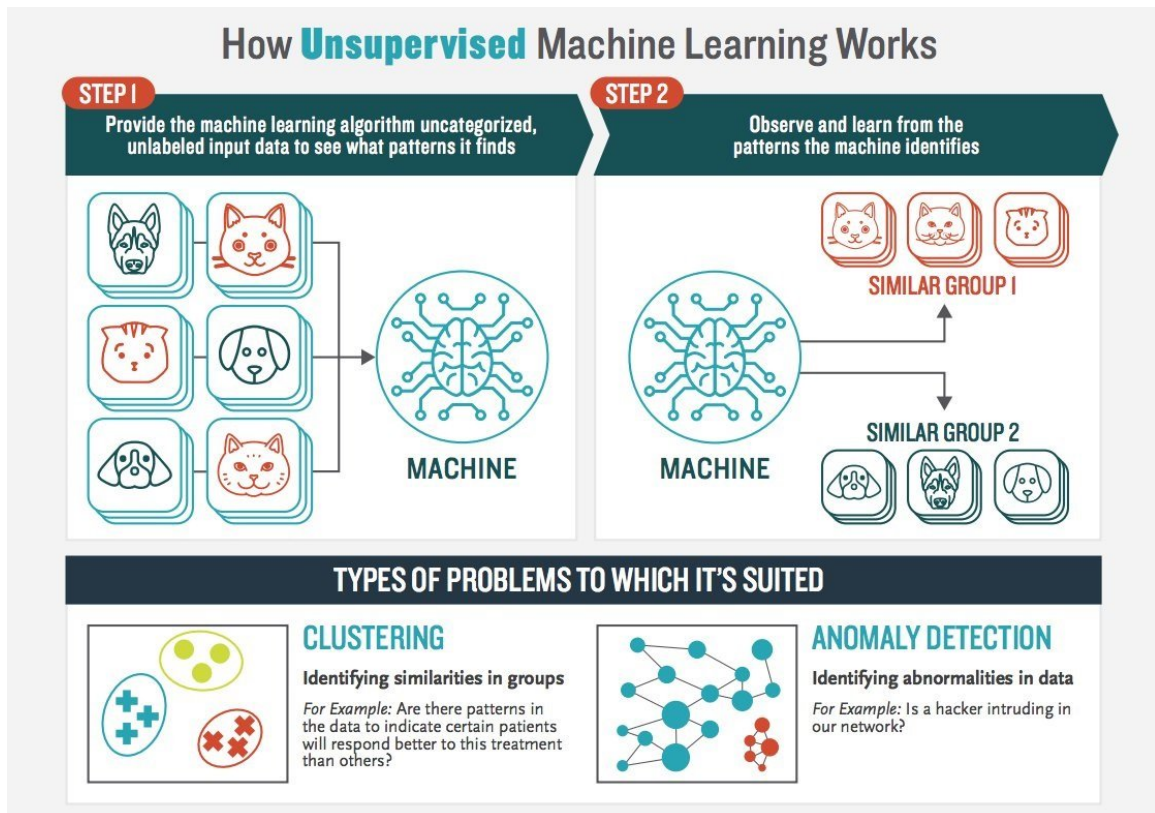


Figure 1.2: Unsupervised Learning.

Tetapi unsupervise learning tidak memiliki outcome yang spesifik layaknya di super- vise learning, hal ini dikarenakan tidak adanya ground truth / label dasar. Walaupun begitu, unsupervised learning masih dapat memprediksi dari ketidakadaan label dari kemiripan attribute yang dimiliki data. Algoritma yang digunakan di unsupervised learning :

1. Clustering
2. Anomaly Detection
3. Training Model
4. Association Discovery

1.1.7 Data Set

Dataset adalah objek yang merepresentasikan data dan relasinya di memory. Strukturnya mirip dengan data di database. Dataset berisi koleksi dari datatable dan data relation.

1.1.8 Training Set

Training set adalah bagian dataset yang kita latih untuk membuat prediksi atau menjalankan fungsi dari sebuah algoritma ML lainnya sesuai tujuannya masing-masing. Kita memberikan petunjuk melalui algoritma agar mesin yang kita latih bisa mencari korelasinya sendiri.

1.1.9 Testing Set

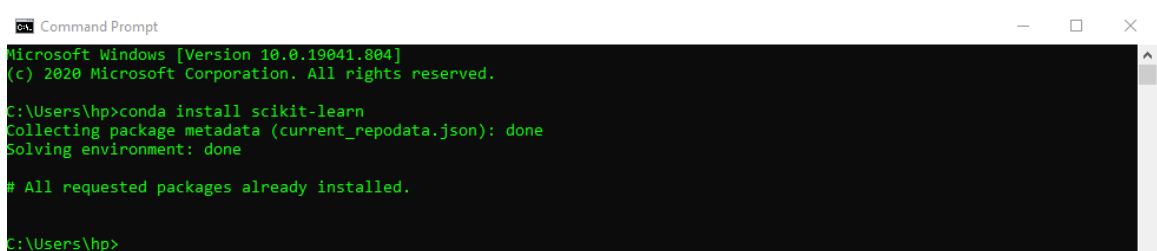
Test set adalah bagian dataset yang kita tes untuk melihat keakuratannya, atau dengan kata lain melihat performanya.

1.2 Praktek

1. Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer.

Instalasi Library Scikit-Learn melalui CMD

- (a) Pertama masuk ke command prompt di menu start.



```
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\hp>conda install scikit-learn
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

C:\Users\hp>
```

Figure 1.3: Instalasi Library Scikit-Learn.

- (b) Selanjutnya ketik "conda activate base" untuk masuk base anaconda.

- (c) Kemudian ketik conda install scikit-learn.
- (d) Lalu apabila sudah done , artinya library scikit sudah kita install

Mencoba Menggunakan Library scikit-Learn

- (a) Pertama jalankan aplikasi Spyder.
- (b) Kemudian buat file baru, lalu tambahkan kode berikut.

```

1  ### 1 Loading an example dataset
2
3  from sklearn import datasets # Digunakan Untuk Memanggil class
   datasets dari library sklearn
4  iris = datasets.load_iris() # Menggunakan contoh datasets iris
5  digits = datasets.load_digits() # Menyimpan nilai data sets
   iris pada variabel digits
6
7  print(digits.target) #menampilkan hasil dari variabel digits
8
9  ### 2 Learning and predicting
10
11 from sklearn import svm #digunakan untuk memanggil svm di
   library sklearn
12 clf = svm.SVC(gamma=0.001, C=100.) #memberikan nilai gamma
   secara manual
13 clf.fit(digits.data[:-1], digits.target[:-1]) #clf sebagai
   classifier dan kemudian set latihan dengan metode fit
14 clf.predict(digits.data[-1:]) #memprediksi nilai baru dar
   digits.data
15
16 ### 3 Model persistence
17
18 from sklearn import svm #digunakan untuk memanggil svm di
   library sklearn
19 from sklearn import datasets # Diguankan untuk class datasets
   dari library sklearn
20 clf = svm.SVC() # membuat variabel clf, dan memanggil class
   svm dan fungsi SVC
21 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
   iris dan mengembalikan nilainya.
22 clf.fit(X, y) #Perhitungan nilai label
23
24 ###
25 import pickle #memanggil library pickle
26 s = pickle.dumps(clf) #untuk membuat variabel s sebagai
   classifier
27 clf2 = pickle.loads(s) # variabel clf2 sebagai load(s)
28 clf2.predict(X[0:1]) # untuk prediksi
29 ###
30 from joblib import dump, load #memanggil dump, load melalui
   library joblib
31 dump(clf, '1184077.joblib') #Menyimpan model kedalam 1184077.
   joblib

```



```

32 clf = load('1184077.joblib') #Memanggil model 1184077
33
34 print(clf) #menampilkan hasil model clf
35
36 """ 4 Conventions
37 import numpy as np # memanggil library numpy dan dibuat alias
   np
38 from sklearn import random_projection #Memanggil class
   random_projection pada library sklearn
39
40 rng = np.random.RandomState(0) #Membuat variabel rng, dan
   mendefisikan np, fungsi random dan attr RandomState kedalam
   variabel
41 X = rng.rand(10, 2000) # untuk membuat variabel X, dan
   menentukan nilai random dari 10 – 2000
42 X = np.array(X, dtype='float32') # untuk menyimpan hasil nilai
   random sebelumnya, kedalam array, dan menentukan
   typedatanya sebagai float32
43 X.dtype # Mengubah data tipe menjadi float64
44
45 transformer = random_projection.GaussianRandomProjection() #
   membuat variabel transformer, dan mendefinisikan
   classrandom_projection dan memanggil fungsi
   GaussianRandomProjection
46 X_new = transformer.fit_transform(X) # membuat variabel baru
   dan melakukan perhitungan label pada variabel X
47 X_new.dtype # Mengubah data tipe menjadi float64
48
49 print(X_new) #menampilkan hasil

```

(c) Simpan dan jalankan.

(d) Hasil dari variabel explorernya sebagai berikut.

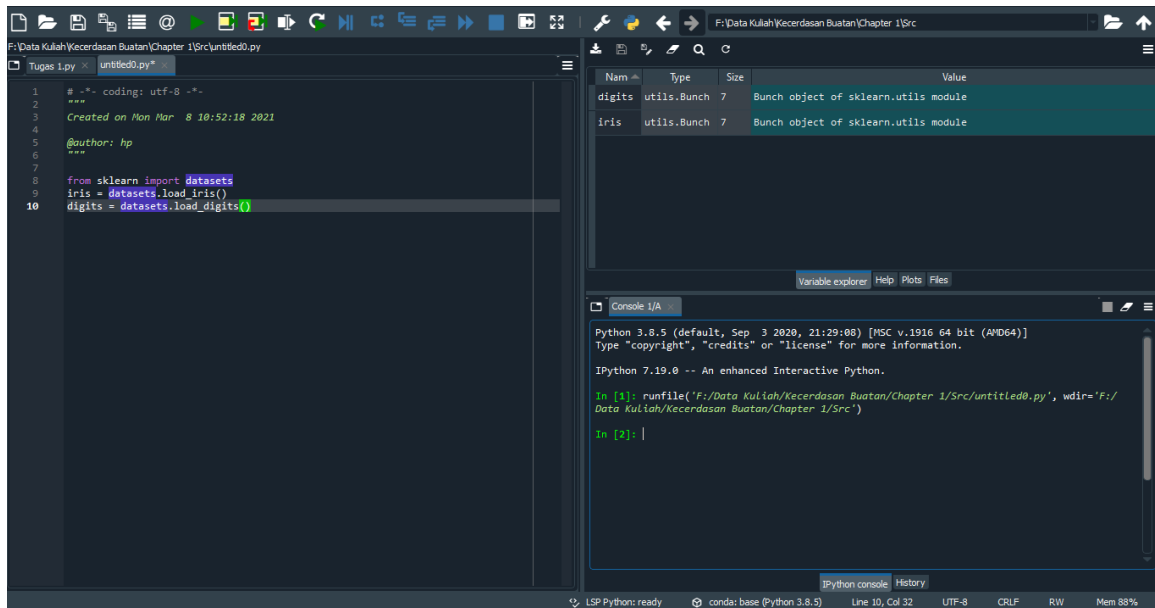


Figure 1.4: Variabel Explorer Library Scikit-Learn.

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

```
1 from sklearn import datasets # Digunakan Untuk Memanggil class
  datasets dari library sklearn
2 iris = datasets.load_iris() # Menggunakan contoh datasets iris
3 digits = datasets.load_digits() # Menyimpan nilai data sets iris
  pada variabel digits
4
5 print(digits.target) #menampilkan hasil dari variabel digits
```

Hasilnya akan seperti ini.

Key	Type	Size	Value
data	str	1	[[0. 0. 5. ... 0. 0. 0.] [0. 0. 0. ... 10. 0. 0.] ..._digits_dataset:
DESCR	Array of float64	(1797, 64)	
feature_names	list	64	['pixel_0_0', 'pixel_0_1', 'pixel_0_2', 'pixel_0_3', 'pixel_0_4', 'pix ...
frame	NoneType	1	NoneType object
images	Array of float64	(1797, 8, 8)	[[[0. 0. 5. ... 1. 0. 0.] [0. 0. 13. ... 15. 5. 0.]
target	Array of int32	(1797,)	[0 1 2 ... 8 9 8]
target_names	Array of int32	(10,)	[0 1 2 3 4 5 6 7 8 9]

Figure 1.5: Hasil Loading an Example Dataset.

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

```
1 from sklearn import svm #digunakan untuk memanggil svm di library
  sklearn
2 clf = svm.SVC(gamma=0.001, C=100.) #memberikan nilai gamma
  secara manual
```

```

3 clf.fit(digits.data[:-1], digits.target[:-1]) #clf sebagai
  classifier dan kemudian set latihan dengan metode fit
4 clf.predict(digits.data[-1:]) #memprediksi nilai baru dari digits.
  data

```

Hasilnya akan seperti ini.

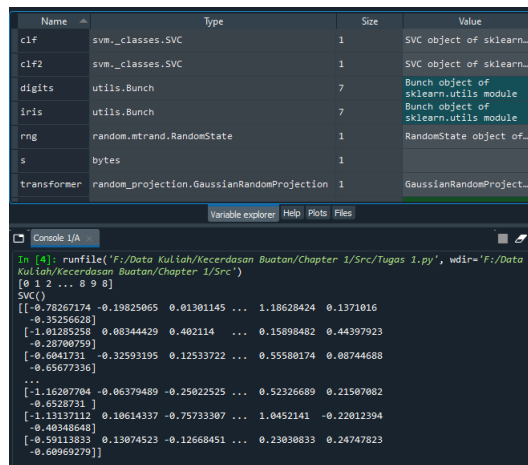


Figure 1.6: Hasil Learning and Predicting.

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris.

```

1 from sklearn import svm #digunakan untuk memanggil svm di library
  sklearn
2 from sklearn import datasets # Diguangkan untuk class datasets dari
  library sklearn
3 clf = svm.SVC() # membuat variabel clf, dan memanggil class svm
  dan fungsi SVC
4 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset iris
  dan mengembalikan nilainya.
5 clf.fit(X, y) #Perhitungan nilai label
6
7 #%%
8 import pickle #memanggil library pickle
9 s = pickle.dumps(clf) #untuk membuat variabel s sebagai classifier
10 clf2 = pickle.loads(s) # variabel clf2 sebagai load(s)
11 clf2.predict(X[0:1]) # untuk prediksi
12 #%%
13 from joblib import dump, load #memanggil dump, load melalui library
  joblib
14 dump(clf, '1184077.joblib') #Menyimpan model kedalam 1184077.
  joblib
15 clf = load('1184077.joblib') #Memanggil model 1184077
16
17 print(clf) #menampilkan hasil model clf

```

Hasilnya akan seperti ini.

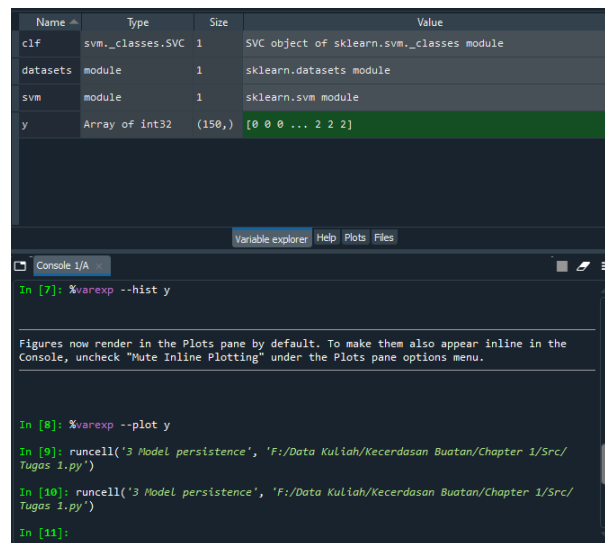


Figure 1.7: Hasil Model Persistence.

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

```

1
2 #%% 4 Conventions
3 import numpy as np # memanggil library numpy dan dibuat alias np
4 from sklearn import random_projection #Memanggil class
   random_projection pada library sklean
5
6 rng = np.random.RandomState(0) #Membuat variabel rng, dan
   mendefisikan np, fungsi random dan attr RandomState kedalam
   variabel
7 X = rng.rand(10, 2000) # untuk membuat variabel X, dan menentukan
   nilai random dari 10 - 2000
8 X = np.array(X, dtype='float32') # untuk menyimpan hasil nilai
   random sebelumnya, kedalam array, dan menentukan typedatanya
   sebagai float32
9 X.dtype # Mengubah data tipe menjadi float64
10
11 transformer = random_projection.GaussianRandomProjection() #membuat
   variabel transformer, dan mendefinisikan classrandom_projection
   dan memanggil fungsi GaussianRandomProjection
12 X_new = transformer.fit_transform(X) # membuat variabel baru dan
   melakukan perhitungan label pada variabel X
13 X_new.dtype # Mengubah data tipe menjadi float64
14
15 print(X_new) #menampilkan hasil

```

Hasilnya akan seperti ini.

Name	Type	Size	Value
np	module	1	numpy module
random_projection	module	1	sklearn.random_projection module
rng	random.mtrand.RandomState	1	RandomState object of nu...
transformer	random_projection.GaussianRandomProjection	1	GaussianRandomProjection...

Variable explorer	Help	Plots	Files
<pre> In [12]: runcell('4 Conventions', 'F:/Data Kuliah/Kecerdasan Buatan/Chapter 1/Src/Tugas 1.py') [[1.31564595 0.11178051 0.1349896 ... -0.32228499 -1.32341083 0.16500849] [0.45043086 -0.29639113 0.19651215 ... 0.08775351 -0.58515958 0.89849878] [0.86841726 0.43772714 0.1717716 ... -0.19614154 -1.21439754 0.80874076] ... [0.73044954 -0.01088444 0.6170979 ... -0.12095845 -0.96185453 0.50549161] [0.50211962 -0.47819985 0.71783534 ... 0.31082068 -0.72910495 0.50651192] [0.6101329 0.27417432 0.82680733 ... -0.04953308 -0.76975617 0.32800572]] In [13]: </pre>			

Figure 1.8: Hasil Conventions.

1.3 Penanganan Error

1. Screenshoot error yang terjadi pada saat melakukan running syntax pada saat melakukan praktikum ini.
 - Name Error dapat terjadi apabila terdapat kesalahan penulisan syntax, yang meliputi nama library, function etc.

```

In [13]: runcell('2 Learning and predicting', 'F:/Data Kuliah/Kecerdasan Buatan/Chapter 1/
Src/Tugas 1.py')
Traceback (most recent call last):

  File "F:/Data Kuliah/Kecerdasan Buatan/Chapter 1/Src/Tugas 1.py", line 13, in <module>
    clf.fit(digits.data[:-1], digits.target[:-1]) #clf sebagai classifier dan kemudian set
    latihan dengan metode fit
NameError: name 'clf' is not defined

```

Figure 1.9: Name Error.

- Import Error

```
File "F:\Data Kuliah\Kecerdasan Buatan\Chapter 1\Src\Tugas 1.py", line 3, in <module>
    from sklearn import dataset # Digunakan Untuk Memanggil class datasets dari library
    sklearn

ImportError: cannot import name 'dataset' from 'sklearn' (C:\Users\hp\anaconda3\lib\site-
packages\sklearn\__init__.py)
```

Figure 1.10: Import Error.

- Syntax Error

```
In [15]: runcell('2 Learning and predicting', 'F:/Data Kuliah/Kecerdasan Buatan/Chapter 1/
Src/Tugas 1.py')
Traceback (most recent call last):

File "F:\Data Kuliah\Kecerdasan Buatan\Chapter 1\Src\Tugas 1.py", line 13, in <module>
    clf.fit(digits.data[:-1], digits.target[:-1]) #clf sebagai classifier dan kemudian set
    latihan dengan metode fit

NameError: name 'digits' is not defined
```

Figure 1.11: Syntax Error.

2. Tuliskan kode error dan jenis errornya.

- Name Error
Name Error adalah exception yang terjadi saat syntax melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi.
- Import Error
Import Error adalah exception yang terjadi saat syntax melakukan import terhadap library yang tidak terdefinisi.
- Syntax Error
Syntax Error adalah exception yang terjadi saat ada kesalahan dalam mengetikkan syntax.

3. Solusi pemecahan masalah error tersebut.

- Name Error
Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- Import Error
Solusinya adalah memastikan library yang dipanggil ada atau tidak salah ketik.

- Syntax Error
Solusinya adalah memastikan syntax yang diketik tidak salah ketik.

1.3.1 Bukti Tidak Plagiat

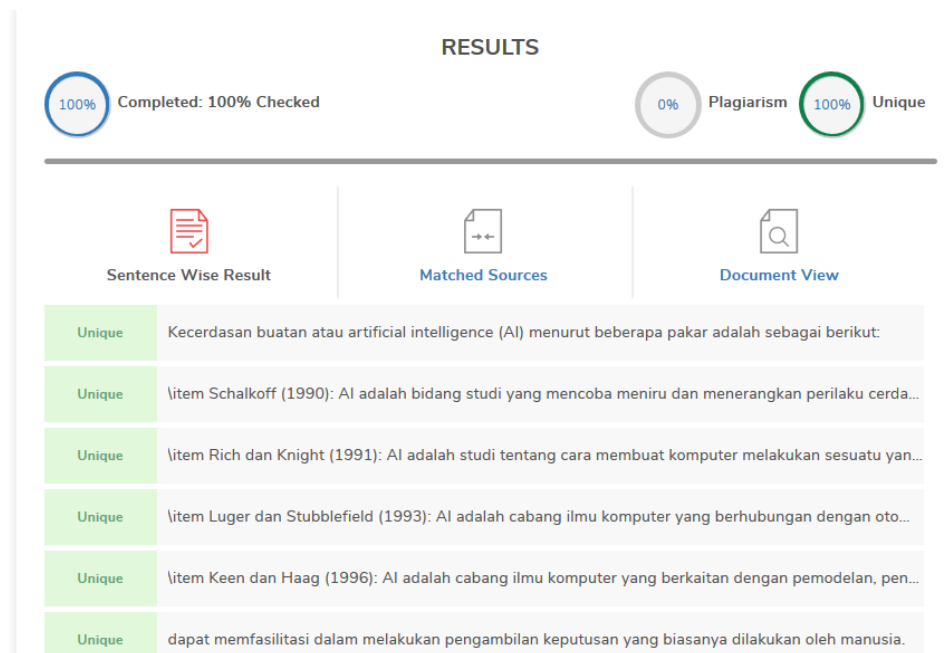


Figure 1.12: Bukti Tidak Plagiat.

Bibliography