

# 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

<b>1</b>	<b>Mengenal Kecerdasan Buatan dan ScikitLearn</b>	<b>1</b>
1.1	Teori . . . . .	1
1.1.1	Kecerdasan Buatan . . . . .	1
1.1.2	Definisi . . . . .	2
1.2	Instalasi . . . . .	6
1.2.1	Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer . . . . .	6
1.2.2	Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris . . . . .	7
1.3	Teori/Annisa Fathoroni-1164067 . . . . .	11
1.3.1	Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris . . . . .	11
1.3.2	Mencoba Model Persistance, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris . . . . .	12
1.3.3	Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris . . . . .	14
1.4	Penanganan Error . . . . .	18
1.5	Teori . . . . .	19
1.5.1	Kecerdasan Buatan . . . . .	19
1.5.2	Perkembangan Kecerdasan Buatan . . . . .	20
1.5.3	Supervised Learning, Klasifikasi dan Regres . . . . .	22
1.6	Instalasi/Annisa Fathoroni-1164067) . . . . .	22
1.7	Penanganan Error/Annisa Fathoroni-1164067) . . . . .	28
<b>2</b>	<b>Related Works</b>	<b>31</b>
2.1	Tasya Wiendhyra/1164086 . . . . .	31
2.1.1	binary classification dilengkapi ilustrasi gambar . . . . .	31

2.1.2	supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar . . . . .	32
2.1.3	evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar . . . . .	34
2.1.4	bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix . . . . .	34
2.1.5	bagaimana K-fold cross validation bekerja dengan gambar ilustrasi . . . . .	35
2.1.6	decision tree dengan gambar ilustrasi . . . . .	35
2.1.7	Information Gain dan entropi dengan gambar ilustrasi . . . . .	36
<b>3</b>	<b>Methods</b>	<b>38</b>
3.1	The data . . . . .	38
3.2	Method 1 . . . . .	38
3.3	Method 2 . . . . .	38
<b>4</b>	<b>Experiment and Result</b>	<b>39</b>
4.1	Experiment . . . . .	39
4.2	Result . . . . .	39
<b>5</b>	<b>Conclusion</b>	<b>40</b>
5.1	Conclusion of Problems . . . . .	40
5.2	Conclusion of Method . . . . .	40
5.3	Conclusion of Experiment . . . . .	40
5.4	Conclusion of Result . . . . .	40
<b>6</b>	<b>Discussion</b>	<b>41</b>
<b>7</b>	<b>Discussion</b>	<b>42</b>
<b>8</b>	<b>Discussion</b>	<b>43</b>
<b>9</b>	<b>Discussion</b>	<b>44</b>
<b>10</b>	<b>Discussion</b>	<b>45</b>
<b>11</b>	<b>Discussion</b>	<b>46</b>
<b>12</b>	<b>Discussion</b>	<b>47</b>

<b>13 Discussion</b>	<b>48</b>
<b>14 Discussion</b>	<b>49</b>
<b>A Form Penilaian Jurnal</b>	<b>50</b>
<b>B FAQ</b>	<b>53</b>

# List of Figures

1.1	Versi Anaconda Yang Digunakan . . . . .	7
1.2	Versi Python Yang Digunakan . . . . .	7
1.3	Instalasi Scikit Dari Anaconda . . . . .	8
1.4	Contoh Skrip . . . . .	9
1.5	Hasil Yang Muncul Di CMD . . . . .	9
1.6	Gambar Yang Muncul Dari Matplotlib . . . . .	10
1.7	Penjelasan . . . . .	10
1.8	Penjelasan 2 . . . . .	11
1.9	Penjelasan 3 . . . . .	11
1.10	Penjelasan 4 . . . . .	11
1.11	Membuka Python Shell . . . . .	12
1.12	Menggunakan Estimator Sklearn . . . . .	12
1.13	Mendefinisikan Classifier . . . . .	12
1.14	Memanggil Classifier Tanpa Baris Terakhir . . . . .	13
1.15	Memprediksi Nilai Baru . . . . .	13
1.16	Hasil Pengujian Classifier . . . . .	13
1.17	Hasil Pengujian Classifier . . . . .	14
1.18	Pickle Pada Python . . . . .	14
1.19	Pengujian Classifier Pickle . . . . .	14
1.20	Penggunaan Joblib . . . . .	15
1.21	Deklarasi Numpy . . . . .	15
1.22	Contoh Type Casting . . . . .	15
1.23	Menggunakan FitTransform . . . . .	16
1.24	Regresi Yang Dilempar . . . . .	16
1.25	Refitting dan Memperbaharui Parameter . . . . .	17
1.26	MultiClass Classifier . . . . .	17
1.27	MultiClass Classifier biner 2D . . . . .	17
1.28	MultiLabel Classifier . . . . .	18



1.29	Error Import . . . . .	18
1.30	Instal Library Joblib . . . . .	19
1.31	Berhasil Import Library Joblib . . . . .	19
1.32	capturing . . . . .	20
1.33	install anaconda 2 . . . . .	22
1.34	capturing . . . . .	23
1.35	capturing . . . . .	23
1.36	capturing . . . . .	24
1.37	capturing . . . . .	24
1.38	capturing . . . . .	25
1.39	capturing . . . . .	25
1.40	capturing . . . . .	26
1.41	capturing . . . . .	26
1.42	capturing . . . . .	26
1.43	capturing . . . . .	27
1.44	capturing . . . . .	27
1.45	capturing . . . . .	28
1.46	capturing . . . . .	28
1.47	capturing . . . . .	28
1.48	capturing . . . . .	29
1.49	capturing . . . . .	29
1.50	capturing . . . . .	30
1.51	capturing . . . . .	30
1.52	capturing . . . . .	30
1.53	capturing . . . . .	30
1.54	capturing . . . . .	30
2.1	Binary Classification . . . . .	31
2.2	Supervised Learning . . . . .	32
2.3	Unsupervised Learning . . . . .	33
2.4	Cluster . . . . .	34
2.5	Evaluasi dan Akurasi . . . . .	34
2.6	K-fold cross validation . . . . .	36
2.7	Decision Tree . . . . .	36
2.8	Information gain . . . . .	37
2.9	Entropi . . . . .	37

A.1	Form nilai bagian 1. . . . .	51
A.2	form nilai bagian 2. . . . .	52

# Chapter 1

## Mengenal Kecerdasan Buatan dan ScikitLearn

### 1.1 Teori

#### 1.1.1 Kecerdasan Buatan

/BAGIAN TASYA WIENDHYRA 1164086 RESUME 1

##### Definisi

Kecerdasan Buatan adalah kemampuan komputer digital atau robot yang dikendalikan komputer untuk melakukan tugas yang umumnya dikaitkan dengan sesuatu yang cerdas. Istilah ini sering diterapkan pada proyek pengembangan sistem yang diberkahi dengan karakteristik proses intelektual manusia, seperti kemampuan untuk berpikir, menemukan makna, menggeneralisasi, atau belajar dari pengalaman masa lalu.

##### Sejarah Singkat

Pada awal 50-an, studi tentang "mesin berpikir" memiliki berbagai nama seperti cybernetics, teori automata, dan pemrosesan informasi. Pada tahun 1956, para ilmuwan jenius seperti Alan Turing, Norbert Wiener, Claude Shannon dan Warren McCullough telah bekerja secara independen di bidang cybernetics, matematika, algoritma dan teori jaringan. Namun, seorang ilmuwan komputer dan kognitif John McCarthy adalah orang yang datang dengan ide untuk bergabung dengan upaya penelitian terpisah ini ke dalam satu bidang yang akan mempelajari topik baru untuk imajinasi manusia yaitu Kecerdasan Buatan. Dia adalah orang yang menciptakan istilah tersebut dan kemudian mendirikan laboratorium Kecerdasan Buatan di MIT

dan Stanford. Pada tahun 1956, McCarthy yang sama mendirikan Konferensi Dartmouth di Hanover, New Hampshire. Peneliti terkemuka dalam teori kompleksitas, simulasi bahasa, hubungan antara keacakan dan pemikiran kreatif, jaringan saraf diundang. Tujuan dari bidang penelitian yang baru dibuat adalah untuk mengembangkan mesin yang dapat mensimulasikan setiap aspek kecerdasan. Itulah sebabnya Konferensi Dartmouth 1956 dianggap sebagai kelahiran Kecerdasan Buatan. Sejak itu, Kecerdasan Buatan telah hidup melalui dekade kemuliaan dan cemoohan, yang dikenal luas sebagai musim panas dan musim dingin AI. Musim panasnya ditandai dengan optimisme dan dana besar, sedangkan musim dinginnya dihadapkan dengan pemotongan dana, ketidakpercayaan dan pesimisme.

### **Perkembangan Kecerdasan Buatan**

AI Summer 1 [1956-1973] Konferensi Dartmouth diikuti oleh 17 tahun kemajuan luar biasa. Proyek penelitian yang dilakukan di MIT, universitas di Edinburgh, Stanford dan Carnegie Mellon menerima dana besar-besaran, yang akhirnya membuahkan hasil. Selama tahun-tahun itulah komputer pemrograman mulai melakukan masalah aljabar, membuktikan teorema geometris, memahami dan menggunakan sintaks dan tata bahasa Inggris. Terlepas dari ditinggalkannya koneksionisme dan terjemahan mesin yang gagal, yang menunda penelitian Natural Language Processing (NLP) selama bertahun-tahun, banyak prestasi dari masa lalu yang membuat sejarah. Berikut ini beberapa di antaranya: Pelopor pembelajaran mesin, Ray Solomonoff meletakkan dasar-dasar teori matematika AI, memperkenalkan metode Bayesian universal untuk inferensi dan prediksi induktif Thomas Evans menciptakan program ANALOGI heuristik, yang memungkinkan komputer memecahkan masalah geometri-analogi Unimation, perusahaan robotika pertama di dunia, menciptakan robot industri Unimate, yang bekerja pada jalur perakitan mobil General Motors. Joseph Weizenbaum membangun ELIZA - program interaktif yang dapat membawa percakapan dalam bahasa Inggris tentang topik apa pun. Ross Quillian menunjukkan jaring semantik, sedangkan Jaime Carbonell (Sr.) mengembangkan Cendekia - program interaktif untuk instruksi yang dibantu komputer berdasarkan jaring semantik. Edward Feigenbaum dan Julian Feldman menerbitkan *Computers and Thought*, kumpulan artikel pertama tentang AI.

#### **1.1.2 Definisi**

RESUME 2

## Supervised Learning

Supervised Learning adalah tugas pengumpulan data untuk menyimpulkan fungsi dari data pelatihan berlabel. Data pelatihan terdiri dari serangkaian contoh pelatihan. Dalam supervised learning, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output yang diinginkan (juga disebut sinyal pengawasan super). Algoritma pembelajaran yang diawasi menganalisis data pelatihan dan menghasilkan fungsi yang disimpulkan, yang dapat digunakan untuk memetakan contoh-contoh baru. Skenario 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”. Supervised Learning menyediakan algoritma pembelajaran dengan jumlah yang diketahui untuk mendukung penilaian di masa depan. Chatbots, mobil self-driving, program pengenalan wajah, sistem pakar dan robot adalah beberapa sistem yang dapat menggunakan pembelajaran yang diawasi atau tidak diawasi. Supervised Learning sebagian besar terkait dengan AI berbasis pengambilan tetapi mereka juga mungkin mampu menggunakan model pembelajaran generatif. Data pelatihan untuk pembelajaran yang diawasi mencakup serangkaian contoh dengan subjek input berpasangan dan output yang diinginkan (yang juga disebut sebagai sinyal pengawasan). Dalam pembelajaran yang diawasi untuk pemrosesan gambar, misalnya, sistem AI mungkin dilengkapi dengan gambar berlabel kendaraan dalam kategori seperti mobil dan truk. Setelah jumlah pengamatan yang cukup, sistem harus dapat membedakan antara dan mengkategorikan gambar yang tidak berlabel, di mana waktu pelatihan dapat dikatakan lengkap. Model Supervised Learning memiliki beberapa keunggulan dibandingkan pendekatan tanpa pengawasan, tetapi mereka juga memiliki keterbatasan. Sistem lebih cenderung membuat penilaian bahwa manusia dapat berhubungan, misalnya, karena manusia telah memberikan dasar untuk keputusan. Namun, dalam kasus metode berbasis pengambilan, Supervised Learning mengalami kesulitan dalam menangani informasi baru. Jika suatu sistem dengan kategori untuk mobil dan truk disajikan dengan sepeda, misalnya, ia harus salah dikelompokkan dalam satu kategori atau yang lain. Namun, jika sistem AI bersifat generatif, ia mungkin tidak tahu apa sepeda itu tetapi akan dapat mengenalinya sebagai milik kategori yang terpisah.

## Regresi

Regresi adalah membahas masalah ketika variabel output adalah nilai riil atau berkelanjutan, seperti “gaji” atau “berat”. Banyak model yang berbeda dapat digu-

nakan, yang paling sederhana adalah regresi linier. Ia mencoba untuk menyesuaikan data dengan hyper-plane terbaik yang melewati poin.

## **Klasifikasi**

Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai terpisah. Label ( $y$ ) umumnya datang dalam bentuk kategorikal dan mewakili sejumlah kelas. Dalam pembelajaran mesin dan statistik, klasifikasi adalah pendekatan pembelajaran yang diawasi di mana program komputer belajar dari input data yang diberikan kepadanya dan kemudian menggunakan pembelajaran ini untuk mengklasifikasikan pengamatan baru. Kumpulan data ini mungkin hanya bersifat dua kelas (seperti mengidentifikasi apakah orang tersebut berjenis kelamin laki-laki atau perempuan atau bahwa surat itu spam atau bukan-spam) atau mungkin juga multi-kelas. Beberapa contoh masalah klasifikasi adalah: pengenalan ucapan, pengenalan tulisan tangan, identifikasi metrik, klasifikasi dokumen dll.

## **Unsupervised Learning**

Unsupervised Learning adalah pelatihan algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasikan atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut tanpa bimbingan. Dalam Unsupervised Learning, sistem AI dapat mengelompokkan informasi yang tidak disortir berdasarkan persamaan dan perbedaan meskipun tidak ada kategori yang disediakan. Sistem AI yang mampu pembelajaran tanpa pengawasan sering dikaitkan dengan model pembelajaran generatif, meskipun mereka juga dapat menggunakan pendekatan berbasis pengambilan (yang paling sering dikaitkan dengan pembelajaran yang diawasi). Chatbots, mobil yang bisa mengemudi sendiri, program pengenalan wajah, sistem pakar dan robot adalah beberapa sistem yang dapat menggunakan pendekatan pembelajaran yang diawasi atau tidak terawasi. Dalam Unsupervised Learning, sistem AI disajikan dengan data yang tidak berlabel, tidak terkategori dan algoritma sistem bekerja pada data tanpa pelatihan sebelumnya. Outputnya tergantung pada algoritma kode. Menundukkan suatu sistem pada Unsupervised Learning adalah salah satu cara untuk menguji AI. Algoritma Unsupervised Learning dapat melakukan tugas pemrosesan yang lebih kompleks daripada sistem pembelajaran yang diawasi. Namun, pembelajaran tanpa pengawasan bisa lebih tidak terduga daripada model alternatif. Sementara Unsupervised Learningi mungkin, misalnya, mencari tahu sendiri cara memilah kucing dari anjing, mungkin juga menambahkan kategori

yang tidak terduga dan tidak diinginkan untuk menangani breed yang tidak biasa, membuat kekacauan bukannya keteraturan.

## Data Set

mendapatkan data yang tepat berarti mengumpulkan atau mengidentifikasi data yang berkorelasi dengan hasil yang ingin Anda prediksi; yaitu data yang berisi sinyal tentang peristiwa yang Anda pedulikan. Data harus diselaraskan dengan masalah yang Anda coba selesaikan. Gambar kucing tidak terlalu berguna ketika Anda sedang membangun sistem identifikasi wajah. Memverifikasi bahwa data selaras dengan masalah yang ingin Anda selesaikan harus dilakukan oleh ilmuwan data. Jika Anda tidak memiliki data yang tepat, maka upaya Anda untuk membangun solusi AI harus kembali ke tahap pengumpulan data. Format ujung kanan untuk pembelajaran dalam umumnya adalah tensor, atau array multi-dimensi. Jadi jalur pipa data yang dibangun untuk pembelajaran mendalam umumnya akan mengkonversi semua data - baik itu gambar, video, suara, suara, teks atau deret waktu - menjadi vektor dan tensor yang dapat diterapkan operasi aljabar linier. Data itu seringkali perlu dinormalisasi, distandarisasi dan dibersihkan untuk meningkatkan kegunaannya, dan itu semua adalah langkah dalam ETL pembelajaran mesin. Deeplearning4j menawarkan alat ETV DataVec untuk melakukan tugas-tugas pemrosesan data tersebut. Pembelajaran yang dalam, dan pembelajaran mesin yang lebih umum, membutuhkan pelatihan yang baik agar bekerja dengan baik. Mengumpulkan dan membangun set pelatihan - badan yang cukup besar dari data yang diketahui - membutuhkan waktu dan pengetahuan khusus domain tentang di mana dan bagaimana mengumpulkan informasi yang relevan. Perangkat pelatihan bertindak sebagai tolok ukur terhadap mana jaring pembelajaran dalam dilatih. Itulah yang mereka pelajari untuk direkonstruksi sebelum mereka melepaskan data yang belum pernah mereka lihat sebelumnya. Pada tahap ini, manusia yang berpengetahuan luas perlu menemukan data mentah yang tepat dan mengubahnya menjadi representasi numerik yang dapat dipahami oleh algoritma pembelajaran mendalam, tensor. Membangun set pelatihan, dalam arti tertentu, pra-pelatihan. Set pelatihan yang membutuhkan banyak waktu atau keahlian dapat berfungsi sebagai keunggulan dalam dunia ilmu data dan pemecahan masalah. Sifat keahlian sebagian besar dalam memberi tahu algoritma Anda apa yang penting bagi Anda dengan memilih apa yang masuk ke dalam set pelatihan. Ini melibatkan menceritakan sebuah kisah - melalui data awal yang Anda pilih - yang akan memandu jaring pembelajaran mendalam Anda saat mereka mengekstraksi fitur-fitur penting, baik di set pelatihan maupun dalam data mentah yang

telah mereka ciptakan untuk dipelajari. Untuk membuat set pelatihan yang bermanfaat, Anda harus memahami masalah yang Anda selesaikan; yaitu apa yang Anda inginkan agar jaring pembelajaran mendalam Anda memperhatikan, di mana hasil yang ingin Anda prediksi.

## **Training Set**

Menjalankan pelatihan yang diatur melalui jaringan saraf mengajarkan pada net cara menimbang berbagai fitur, menyesuaikan koefisien berdasarkan kemungkinan mereka meminimalkan kesalahan dalam hasil Anda. Koefisien-koefisien tersebut, juga dikenal sebagai parameter, akan terkandung dalam tensor dan bersama-sama mereka disebut model, karena mereka mengkodekan model data yang mereka latih. Mereka adalah takeaways paling penting yang akan Anda dapatkan dari pelatihan jaringan saraf.

## **Test Set**

Ini berfungsi sebagai meterai persetujuan, dan Anda tidak menggunakannya sampai akhir. Setelah Anda melatih dan mengoptimalkan data Anda, Anda menguji jaringan saraf Anda terhadap pengambilan sampel acak akhir ini. Hasil yang dihasilkannya harus memvalidasi bahwa jaring Anda secara akurat mengenali gambar, atau mengenalinya setidaknya  $x$  dari jumlah tersebut. Jika Anda tidak mendapatkan prediksi yang akurat, kembalilah ke set pelatihan, lihat hyperparameter yang Anda gunakan untuk menyetel jaringan, serta kualitas data Anda dan lihat teknik pra-pemrosesan Anda. / AKHIR BAGIAN TEORI TASYA WIENDHYRA 1164086

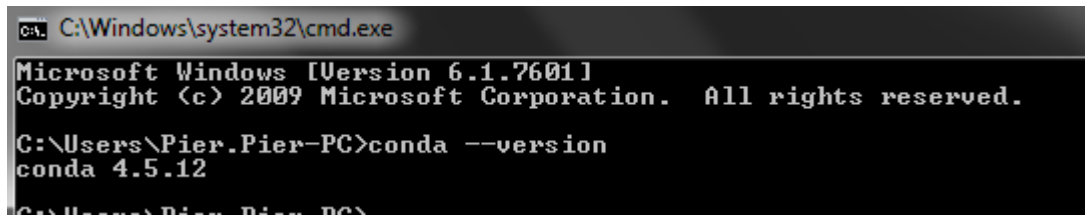
## **1.2 Instalasi**

/BAGIAN TASYA WIENDHYRA 1164086 Membuka <https://scikit-learn.org/stable/tutorial/basics/> Dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat. Dan wajib skrinsut dari komputer sendiri.

### **1.2.1 Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer**

1. Pastikan bahwa sudah terinstal Anaconda pada PC anda, caranya buka CMD lalu ketikkan "conda -version" jika hasilnya seperti

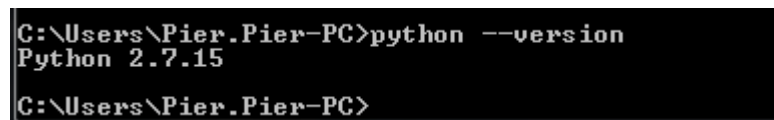




```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Pier.Pier-PC>conda --version
conda 4.5.12
```

Figure 1.1: Versi Anaconda Yang Digunakan



```
C:\Users\Pier.Pier-PC>python --version
Python 2.7.15
C:\Users\Pier.Pier-PC>
```

Figure 1.2: Versi Python Yang Digunakan

2. Pastikan juga Kebutuhan Scikit seperti Numpy, Scipy dan Python telah terinstal. untuk mengeceknya buka CMD dan ketikan seperti gambar berikut.
3. Pada CMD ketikan "conda install scikit-learn" kemudian tunggu sampai instalasi selesai.
4. Setelah itu, kita akan mencoba salah satu contoh dasar penggunaan scikit pada website sebelumnya. Dan disini menggunakan contoh Multilabel classification.
5. Salin skrip contoh tersebut ke Text Editor Visual Code atau yang anda miliki. File ini kemudian di save dengan nama "contoh.py"
6. Setelah tersimpan, jalankan di CMD dengan mengetikan "python contoh.py" maka akan muncul hasil seperti dibawah ini.

### 1.2.2 Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

1. Mengimport dataset, iris dan digit sebagai contoh data.
2. Misalnya, dalam kasus dataset digit, digits.data memberikan akses ke fitur yang dapat digunakan untuk mengklasifikasikan sampel digit.
3. digit.target memberikan kebenaran dasar untuk dataset digit, yaitu angka yang sesuai dengan setiap gambar digit yang dipelajari.
4. menggambarkan bagaimana mulai dari masalah awal seseorang dapat membentuk data untuk konsumsi di scikit-belajar.

```

C:\Users\Pier.Pier-PC>conda install scikit-learn
Solving environment: done

## Package Plan ##

  environment location: E:\Anaconda2

  added / updated specs:
    - scikit-learn

The following packages will be downloaded:



| package                   | build          |        |
|---------------------------|----------------|--------|
| scikit-learn-0.20.2       | py27hf381715_0 | 5.1 MB |
| ca-certificates-2019.1.23 | 0              | 158 KB |
| conda-4.6.7               | py27_0         | 1.7 MB |
| Total:                    |                | 7.0 MB |



The following packages will be UPDATED:

ca-certificates: 2018.03.07-0 --> 2019.1.23-0
conda: 4.5.12-py27_0 --> 4.6.7-py27_0
scikit-learn: 0.20.1-py27hf381715_0 --> 0.20.2-py27hf381715_0

Proceed <[y]/n>? y

Downloading and Extracting Packages
scikit-learn-0.20.2 : 5.1 MB : ##### : 100%
ca-certificates-2019 : 158 KB : ##### : 100%
conda-4.6.7 : 1.7 MB : ##### : 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

```

Figure 1.3: Instalasi Scikit Dari Anaconda

```

Contoh.py x app.py
    'Boundary\nfor class 1')
plot_hyperplane(classif.estimators_[1], min_x, max_x, 'k-.',
    'Boundary\nfor class 2')
plt.xticks(())
plt.yticks(())

plt.xlim(min_x - .5 * max_x, max_x + .5 * max_x)
plt.ylim(min_y - .5 * max_y, max_y + .5 * max_y)
if subplot == 2:
    plt.xlabel('First principal component')
    plt.ylabel('Second principal component')
    plt.legend(loc="upper left")

figure(figsize=(8, 6))

= make_multilabel_classification(n_classes=2, n_labels=1,

```

Figure 1.4: Contoh Skrip

```

C:\Users\Pier.Pier-PC>E:
E:\>python Contoh.py
=====
Isotonic Regression
=====

An illustration of the isotonic regression on generated data. The
isotonic regression finds a non-decreasing approximation of a function
while minimizing the mean squared error on the training data. The benefit
of such a model is that it does not assume any form for the target
function such as linearity. For comparison a linear regression is also
presented.

```

Figure 1.5: Hasil Yang Muncul Di CMD

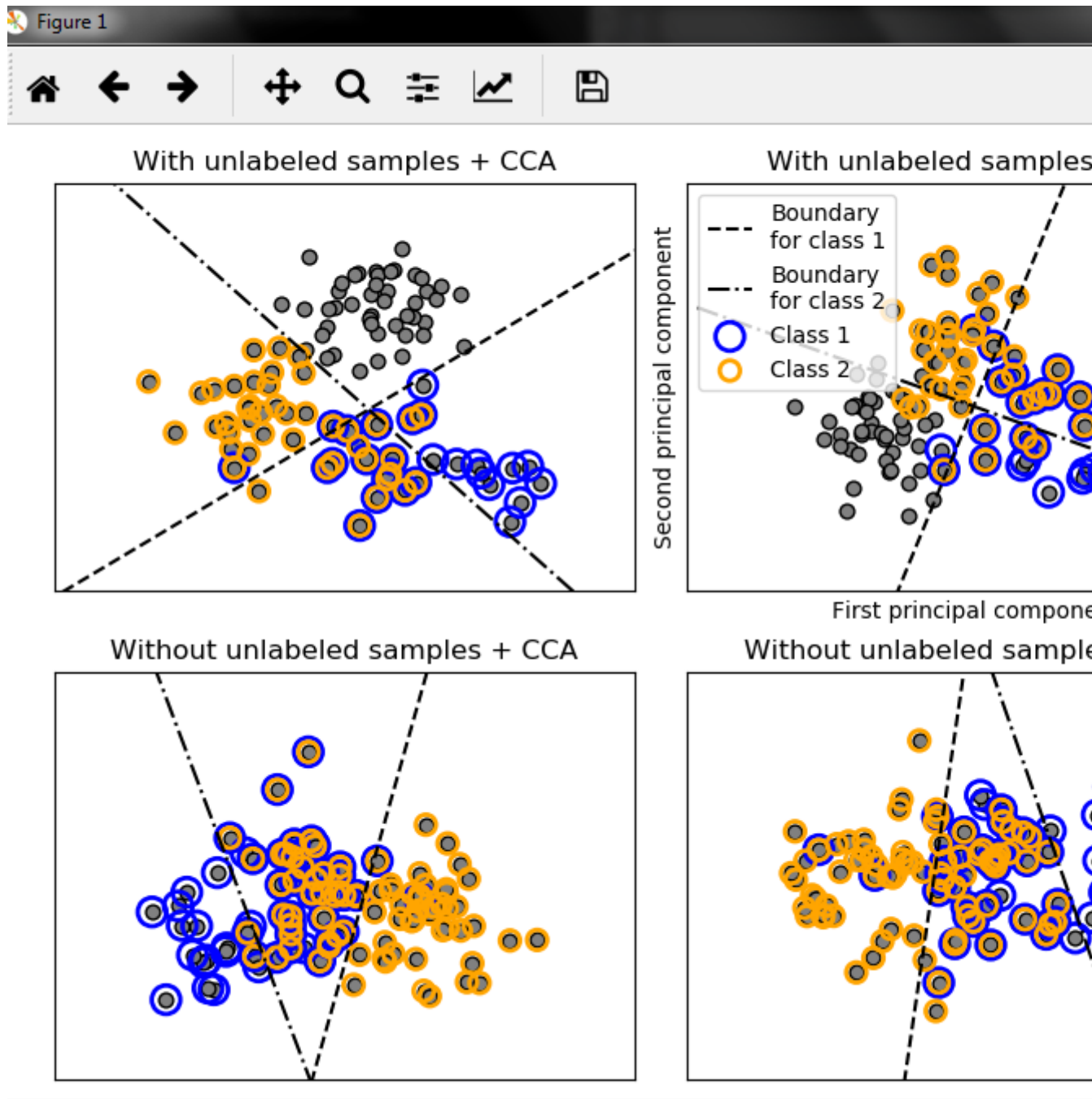


Figure 1.6: Gambar Yang Muncul Dari Matplotlib

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

Figure 1.7: Penjelasan

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

Figure 1.8: Penjelasan 2

```
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
```

Figure 1.9: Penjelasan 3

## 1.3 Teori/Annisa Fathoroni-1164067

### 1.3.1 Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

HARI 2

1. Pada website sebelumnya, cari Learning And Predicting dan ikuti langkah - langkahnya.
2. Buka Python Shell, atau dengan membukan Command Prompt di PC dan mengetikan python. maka akan masuk ke Python Shellnya.
3. Pada python shell ketikan "from sklearn import svm" yang dimana artinya akan memanggil dan menggunakan estimator dari kelas sklearn.svm.SVC
4. Kemudian, kita definisikan clf sebagai classfier, disini gamma didefinisikan secara manual
5. 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,

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

Figure 1.10: Penjelasan 4

```
C:\Users\Pier.Pier-PC>python
Python 2.7.15 |Anaconda, Inc.| <default, Dec 10 2018, 21:57:18> [MSC v.1500 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 1.11: Membuka Python Shell

```
C:\Users\Pier.Pier-PC>python
Python 2.7.15 |Anaconda, Inc.| <default, Dec 10 2018, 21:57:18> [MSC v.1500 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import svm
>>>
```

Figure 1.12: Menggunakan Estimator Sklearn

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`

6. Pada penggalan skrip dibawah, ini menunjukan prediksi nilai baru menggunakan gambar terakhir dari `digits.data`. Dengan prediksi akan menentukan gambar dari set pelatihan yang paling cocok dengan gambar terakhir.

/AKHIR BAGIAN INSTALASI TASYA WIENDHYRA 1164086

### 1.3.2 Mencoba Model Persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

/BAGIAN TASYA WIENDHYRA 1164086

1. Pada Python Shell ketikan "from sklearn import svm" artinya akan mengimport sebuah Support Vector Machine(SVM) yang merupakan algoritma classification yang akan diambil dari Scikit-Learn.
2. Kemudian, lanjutkan dengan "from sklearn import datasets" yang artinya akan mengambil package datasets dari Scikit-Learn.

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

Figure 1.13: 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 1.14: Memanggil Classifier Tanpa Baris Terakhir

```

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

```

Figure 1.15: Memprediksi Nilai Baru

- ketikan, `clf = svm.SVC(gamma='scale')` berfungsi untuk mendeklarasikan suatu value yang bernama `clf` yang berisi `gamma`. Parameter `gamma` menentukan seberapa jauh pengaruh dari satu contoh training.
- Ketikan, `X, y = iris.data, iris.target`, artinya `X` sebagai data iris, dan `y` merupakan larik target.
- 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)`.

- Setelah mengetikan 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.

```

>>> 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 1.16: Hasil Pengujian Classifier

```
clf=0.001, verbose=False)
6. >>> import pickle
>>> s = pickle.dumps(clf)
```

Figure 1.17: Hasil Pengujian Classifier

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

Figure 1.18: Pickle Pada Python

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

9. Dalam kasus khusus scikit-learn, mungkin lebih menarik untuk menggunakan joblib (dump dan load) untuk menggantikan Pickle, yang lebih efisien pada data besar tetapi hanya bisa di Pickle ke disk dan tidak ke string. untuk menggunakan Joblib pertama ketikan "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

### 1.3.3 Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

1. Import numpy as np, digunakan untuk mengimport Numpy sebagai np.  
From sklearn import randomprojection artinya modul yang mengimplemen-  
tasikan cara sederhana dan efisien secara komputasi untuk mengurangi dimensi  
data dengan memperdagangkan sejumlah akurasi yang terkendali (sebagai var-  
ian 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 ran-  
dom number generator.

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

Figure 1.19: Pengujian Classifier Pickle



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

Figure 1.20: Penggunaan Joblib

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

Figure 1.21: Deklarasi Numpy

$X = \text{rng.rand}(10, 2000)$  artinya akan merandom value antara 10 sampai 2000.  $X = \text{np.array}(X, \text{dtype}='float32')$  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.

$\text{list}(\text{clf.predict}(\text{irisdata}[:,3]))$ , akan memprediksi 3 data dari iris.

$\text{clf.fit}(\text{irisdata}, \text{iristargetnames}[\text{iristarget}]$  menguji classifier dengan ada targetnya yaitu irisnya sendiri.

$\text{list}(\text{clf.predict}(\text{irisdata}[:,3]))$ , 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

$y = \text{rngbinomial}(1, 0.5, 100)$  , random value dengan angka binomial atau suku dua untuk  $y$

$\text{clf.setparams}(\text{kernel}='linear')$ fit( $X, y$ ) mengubahn kernel default menjadi linear

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

2.

Figure 1.22: Contoh Type Casting

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

Figure 1.23: Menggunakan 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 1.24: Regresi Yang Dilempar

`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).

`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.

```

>>> 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 1.25: Refitting dan 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 1.26: MultiClass Classifier

```

>>> 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 1.27: MultiClass Classifier biner 2D

```

>>> 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 1.28: MultiLabel Classifier

```

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

```

Figure 1.29: Error Import

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.

/AKHIR BAGIAN PEGUJIAN TASYA WIENDHYRA 1164086

## 1.4 Penanganan Error

/BAGIAN TASYA WIENDHYRA 1164086 HARI KEDUA

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

```
C:\Users\Pier.Pier-PC>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 799kB/s
Installing collected packages: joblib
Successfully installed joblib-0.13.2
```

Figure 1.30: Instal Library Joblib

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

Figure 1.31: Berhasil Import Library Joblib

5. Apabila sudah terinstall, dapat dilakukan lagi import library joblib, maka akan berhasil seperti dibawah berikut

/AKHIR BAGIAN PENANGANAN EROR TASYA WIENDHYRA 1164086

## 1.5 Teori

Teori mencakup resume dari beberapa pembahasan. yaitu :

### 1.5.1 Kecerdasan Buatan

- Definisi Kecerdasan Buatan.

Jadi yang dimaksud dengan kecerdasan buatan yaitu salah satu dari cabang Ilmu pengetahuan yang kita punya dan berhubungan juga dengan pemanfaatan mesin yaitu untuk dapat memecahkan suatu persoalan yang rumit dengan menggunakan cara yang lebih manusiawi.

Jadi yang dimaksud dengan supervised learning adalah pembelajaran mesin yang diawasi menciptakan model yang melancarkan prediksi berdasarkan bukti adanya ketidakpastian. Algoritma pembelajaran yang diawasi memerlukan seperangkat data masukan dan tanggapan yang diketahui terhadap data (output) dan melatih model untuk menghasilkan prediksi yang masuk akal untuk respon terhadap data baru. Sedangkan untuk klasifikasi yaitu nilai output yang

bernilai diskrit (kelas) dan Bertujuan mengklasifikasi data baru dengan akurat diawasi dengan menggunakan teknik klasifikasi dan regresi untuk mengembangkan model prediktif. Yang dimaksud dengan regresi adalah nilai output yang bernilai kontinu (riil), Bertujuan memprediksi output dengan akurat untuk data baru.

Unsupervised learning tidak menggunakan data latih atau data training untuk melakukan prediksi maupun klasifikasi. Berdasarkan model matematisnya, algoritma ini tidak memiliki target variabel. Salah satu tujuan dari algoritma ini adalah mengelompokkan objek yang hampir sama dalam suatu area tertentu. Dan yang dimaksud dengan Dataset adalah objek yang merepresentasikan data dan relasinya di memory. Strukturnya mirip dengan data di database. Dataset berisi koleksi dari datatable dan datarelation. Sedangkan testing set digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Dan training set digunakan oleh algoritma klasifikasi untuk membentuk sebuah model classifier. Model yang dimaksud ini merupakan representasi pengetahuan yang akan digunakan untuk prediksi kelas data baru yang belum pernah ada.

Kecerdasan buatan (Artificial Intelligence) atau disingkat menjadi AI adalah suatu pengetahuan yang membuat suatu komputer dapat menirakan kecerdasan manusia sehingga diharapkan atau diinginkan komputer dapat melakukan hal-hal yang apabila dikerjakan manusia memerlukan kecerdasan. Misalnya melakukan penalaran untuk mencapai suatu kesimpulan atau melakukan translasi dari satu bahasa manusia ke bahasa manusia yang lain.

## 1.5.2 Perkembangan Kecerdasan Buatan

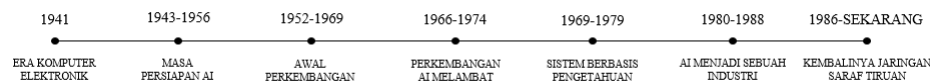


Figure 1.32: capturing

Ditemukannya pertama kali alat penyimpanan dan pemrosesan informasi yang disebut komputer elektronik. Tahun 1943, Warren McCulloch dan Walter Pitts berhasil membuat suatu model saraf tiruan di mana setiap neuron digambarkan sebagai 'on' dan 'off'. Pada tahun 1950, Norbert Wiener membuat penelitian mengenai prinsip-prinsip teori feedback. Pada tahun 1956, John McCarthy meyakinkan Minsky,

Claude Shannon, dan Nathaniel Rochester untuk membantunya melakukan penelitian dalam bidang automata, jaringan saraf, dan pembelajaran inteligensia. Mereka mengerjakan proyek ini selama 2 bulan di Universitas Dartmouth. Hasilnya adalah program yang mampu berpikir non-numerik dan menyelesaikan masalah pemikiran, yang dinamakan Principia Mathematica. Hal ini menjadikan McCarthy disebut sebagai father of Artificial Intelligence/ Bapak Kecerdasan Buatan.

Pada tahun 1958, McCarthy di MIT AI Lab mendefinisikan bahasa pemrograman tingkat tinggi yaitu LISP, yang sekarang mendominasi pembuatan program-program AI. Kemudian, McCarthy membuat program yang dinamakan programs with common sense. Pada tahun 1959, Program komputer General Problem Solver berhasil dibuat oleh Herbert A. Simon, J.C. Shaw, dan Allen Newell. Program ini dirancang untuk memulai penyelesaian masalah secara manusiawi. Pada tahun yg sama Nathaniel Rochester dari IBM dan para mahasiswanya merilis program AI yaitu geometry theorem prover. 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 geometri yang ada pada tes IQ.

Perkembangan AI melambat disebabkan adanya beberapa kesulitan yang di hadapi seperti Program-program AI yang bermunculan hanya mengandung sedikit atau bahkan tidak mengandung sama sekali pengetahuan pada subjeknya, banyak terjadi kegagalan pada pembuatan program AI, terdapat beberapa batasan pada struktur dasar yang digunakan untuk menghasilkan perilaku inteligensia. Pada tahun 1960an, Ed Feigenbaum, Bruce Buchanan, dan Joshua Lederberg merintis proyek DENDRAL yaitu program untuk memecahkan masalah struktur molekul dari informasi yang didapatkan dari spectrometer massa. Pada tahun 1986, program ini telah berhasil menghemat 40 juta dolar 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. Perusahaan yang sejak tahun 1982 hanya menghasilkan beberapa juta US dollar per tahun meningkat menjadi 2 milyar US dollar per tahun pada tahun 1988.

Meskipun bidang ilmu komputer menolak jaringan saraf tiruan setelah diterbitkannya buku 'Perceptrons' karangan Minsky dan Papert, tetapi para ilmuwan masih mempelajari bidang ilmu tersebut dari sudut pandang yang lain, yaitu fisika. Ahli fisika seperti Hopfield (1982) menggunakan teknik-teknik mekanika statistika untuk menganalisa sifat-sifat penyimpanan dan optimasi pada jaringan saraf. Para ahli psikolog, David Rumelhart dan Geoff Hinton melanjutkan penelitian mengenai

model jaringan saraf pada memori. Pada tahun 1985-an sedikitnya empat kelompok riset menemukan algoritma Back-Propagation. Algoritma ini berhasil diimplementasikan ke dalam ilmu bidang komputer dan psikologi.

### 1.5.3 Supervised Learning, Klasifikasi dan Regres

- Supervised learning adalah 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.
- Dalam klasifikasi, keluaran dari setiap data adalah bilangan bulat atau diskrit. Misalnya pengambilan keputusan untuk main sepak bola atau tidak maka keluaran bisa diubah kedalam bilangan bulat 1 (main bola), dan -1 (tidak main). Regresi, keluaran dari setiap data adalah bilangan kontinu. Misalnya peramalan harga rumah berdasarkan lokasi, umur rumah dan luas rumah, maka keluarannya berupa bilangan kontinu berupa bilangan Rp 120 juta, Rp 100 juta atau Rp 51 juta.

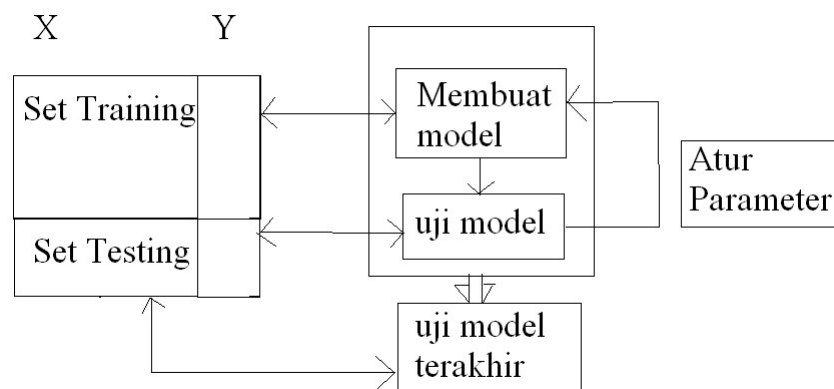


Figure 1.33: install anaconda 2

## 1.6 Instalasi/Annisa Fathoroni-1164067)

Untuk Instalasinya mencakup beberapa pembahasan dan tutorial. yaitu :

### 1. Instalasi Scikit-Learn Dari Anaconda

- Instalasi Anaconda
- Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10) Pada gambar diatas dapat



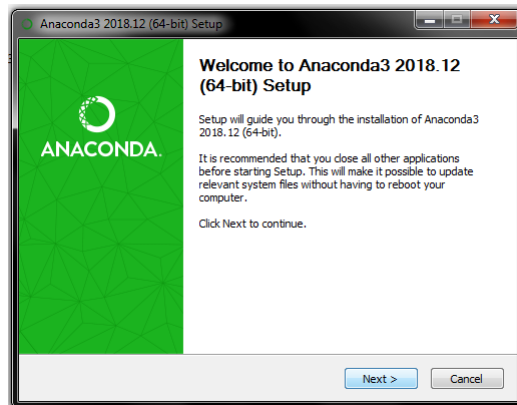


Figure 1.34: capturing

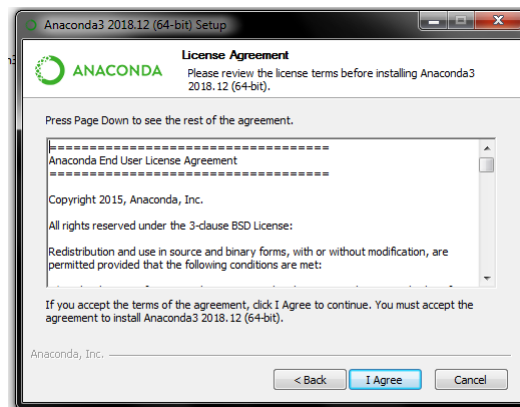


Figure 1.35: capturing

dijelaskan bahwa : Baris yang pertama kita memasukkan dan memanggil svm dari sklearn. Baris yang kedua kita membuat variable.

- Pada gambar dibawah dapat dijelaskan bahwa : Baris yang pertama itu clf dipasang pada model fit metode. Baris yang kedua kita implementasikan klasifikasi dukungan vektor.
- Pada gambar dibawah dapat dijelaskan bahwa : Baris yang pertama itu untuk prediksi nilai baru. Baris yang kedua untuk set array.
- Setelah proses download selesai, run package yang telah didownload lalu next.  
Pilih 'I Agree' untuk menyetujui persyaratan dan peraturan mengenai aplikasi ini.  
Lalu, pilih 'All Users' untuk dapat diakses oleh semua user, dan pilih "Just Me" hanya untuk dapat diakses oleh 1 user pc.

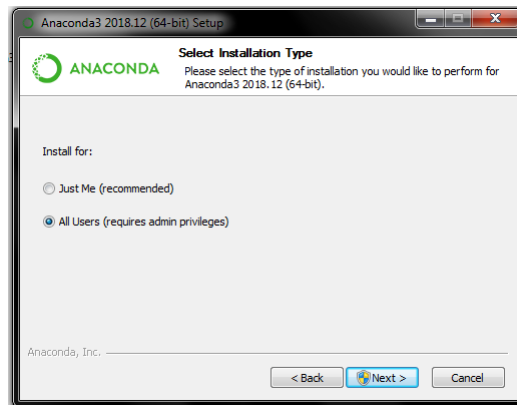


Figure 1.36: capturing

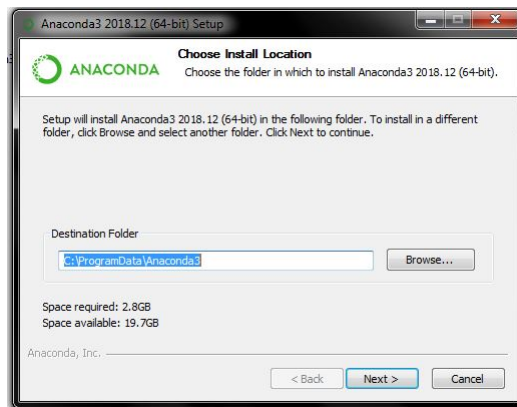


Figure 1.37: capturing

Kemudian, pilih penyimpanan package.

Ceklis pada bagian cek box pertama untuk otomatis pengaturan dan penambahan enviroment variabel pada PATH dan cek box kedua untuk register anaconda sistem.

Proses instalasi Lalu pada proses ini, skip untuk mempercepat proses penginstalan.

Proses instalasi selesai.

Setelah proses instalasi selesai, cek penginstalan conda dan python.

Lalu install scikit dengan perintah 'pip install -U scikit-learn" atau 'conda install scikit-learn'

- Loading an Example Dataset
- mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10) Pada gambar diatas dapat dijelaskan bahwa :

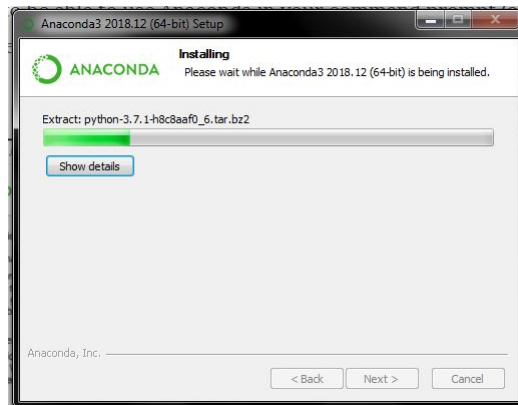


Figure 1.38: capturing

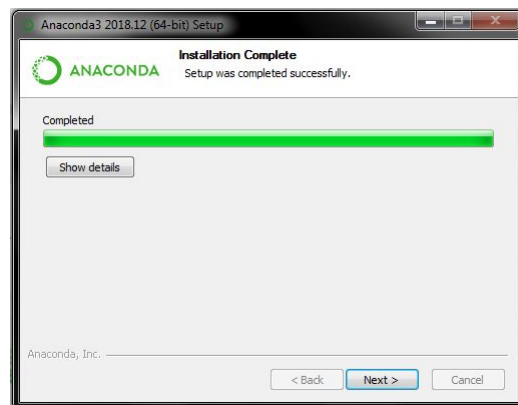


Figure 1.39: capturing

Baris yang pertama kita memasukkan dan memanggil datasets dari sklearn.  
 Baris yang kedua kita memasukkan dan memanggil svm dari sklearn.  
 Baris yang ketiga kita membuat variable clf.  
 Baris yang keempat kita membuat variable iris.  
 Baris yang kelima kita membuat variable x, y.  
 Baris yang keenam clf dipasang pada model fit metode.  
 Baris yang ke tujuh kita implementasikan klasifikasi dukungan vektor.  
 Baris yang kedelapan kita memanggil library pickle.  
 Baris yang kesembilan untuk membuat variable s.  
 Baris yang kesepuluh untuk membuat variable clf2.  
 Baris yang keseblas untuk prediksi nilai baru.  
 Baris yang keduabelas untuk set array.

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

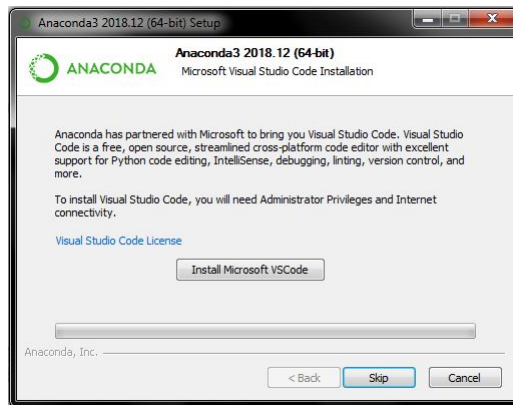


Figure 1.40: capturing

```
from sklearn import svm
clf = svm.SVC(gamma=0.001, C=100.)
```

Figure 1.41: capturing

2. Baris 1: Memanggil data dari sklearn.
3. Baris 2: Terdapat variabel 'iris', dimana variabel iris memanggil datasets dan di dalamnya akan memproses digits.
4. Baris 3: Kemudian ada variabel lainnya 'digits' yang digunakan untuk memanggil dataset dan di dalamnya akan memproses digits.
5. Kemudian untuk perintah `Print( digits.data )` untuk menampilkan output dari hasil variabel digits dan akan berupa data.

- Mencoba Learning and Predicting.

Baris 1 = Memasukkan dan memanggil SVM dari sklearn.

Baris 2 = Membuat variabel.

Baris 1 = clf dipasang pada model fit metode.

Baris 2 = Mengimplementasikan klasifikasi dukungan vektor.

Baris 1 = Prediksi nilai baru.

Baris 2 = set array.

```
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 1.42: capturing

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

Figure 1.43: capturing

```
C:\Windows\system32>pip install -U scikit-learn
Collecting scikit-learn
  Using cached https://files.pythonhosted.org/packages/d1/6c/6ddh21e203ff95d7080
aeee2105b4f6610a02483e00d4ac950f3630969c9/scikit_learn-0.20.2-cp37-cp37m-win_and
64.whl
Requirement already satisfied, skipping upgrade: numpy>=1.8.2 in c:\programdata\
anaconda3\lib\site-packages (from scikit-learn) (1.15.4)
Requirement already satisfied, skipping upgrade: scipy>=0.13.3 in c:\programdata\
anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
Installing collected packages: scikit-learn
  Found existing installation: scikit-learn 0.20.1
    Uninstalling scikit-learn-0.20.1:
      Successfully uninstalled scikit-learn-0.20.1
  Successfully installed scikit-learn-0.20.2
C:\Windows\system32>
```

Figure 1.44: capturing

- Model Persistence

Baris 1 = Memasukkan dan memanggil datasets dari sklearn.

Baris 2 = Memasukkan dan memanggil SVM dari sklearn.

Baris 3 = Membuat variable clf.

Baris 4 = Membuat variable iris.

Baris 5 = Membuat variable x dan y.

Baris 6 = clf dipasang pada model fit metode.

Baris 7 = Mengimplementasikan klasifikasi dukungan vektor.

Baris 8 = Memanggil library pickle.

Baris 9 = Membuat variable s.

Baris 10 = Membuat variable clf2.

Baris 11 = Memprediksikan nilai baru.

Baris 12 = Set array.

- Conventions

Baris 1 = Memasukkan dan memanggil numpy sebagai np.

Baris 2 = Memasukkan dan memanggil randomprojection dari sklearn.

Baris 3 = Membuat variable rng.

Baris 4 = Membuat variable rng.

Baris 5 = Membuat variable x.

Baris 6 = Proses pemanggilan variable x.

Baris 7 = Proses pemanggilan dtype.

```
C:\Windows\system32\python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Ana
conda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print ('Annisa Fathoroni')
Annisa Fathoroni
>>>
```

Figure 1.45: capturing

```
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)
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)

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

Figure 1.46: capturing

Baris 8 = Membuat variable tranformer.

Baris 9 = Membuat variable xnew.

Baris 10 = Proses pemanggilan xnew.

Baris 11 = Pemanggilan dtype.

## 1.7 Penanganan Error/(Annisa Fathoroni-1164067)

Untuk penanganan error mencakup beberapa pembahasan dan tutorial. yaitu :

1. Screenshoot Error
2. Kode Error dan Jenis Errornya

SyntaxError: invalid syntax

```
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
dtype('float32')

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

Figure 1.47: capturing

```
from sklearn import svm
clf = svm.SVC(gamma=0.001, C=100.)
```

Figure 1.48: capturing

```
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 1.49: capturing

3. Solusi Pemecahan Error. Pada gambar dibawah dapat dijelaskan bahwa :
  - Baris yang pertama kita memasukkan dan memanggil numpy sebagai np.
  - Baris yang kedua kita memasukkan dan memanggil randomprojection dari sklearn.
  - Baris yang ketiga kita membuat variable rng.
  - Baris yang keempat untuk membuat variable rng.
  - Baris yang kelima untuk membuat variable x.
  - Baris yang keenam untuk pemanggilan variable x.
  - Baris yang ketujuh untuk pemanggilan dtype.
  - Baris yang kedelapan kita membuat variable tranformer.
  - Baris yang kesembilan kita membuat variable xnew.
  - Baris yang kesempuluh untuk pemanggilan xnew.
  - Baris yang keseblas untuk pemanggilan dtype.

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

Figure 1.50: capturing

```
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)
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)

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

Figure 1.51: capturing

```
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
dtype('float32')

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

Figure 1.52: capturing

```
>>> print(Annisa Fathoroni)
File "<stdin>", line 1
    print(Annisa Fathoroni)
          ^
SyntaxError: invalid syntax
```

Figure 1.53: capturing

```
>>> print('Annisa Fathoroni')
Annisa Fathoroni
>>> _
```

Figure 1.54: capturing



# Chapter 2

## Related Works

Your related works, and your purpose and contribution which.

### 2.1 Tasya Wiendhyra/1164086

#### 2.1.1 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).

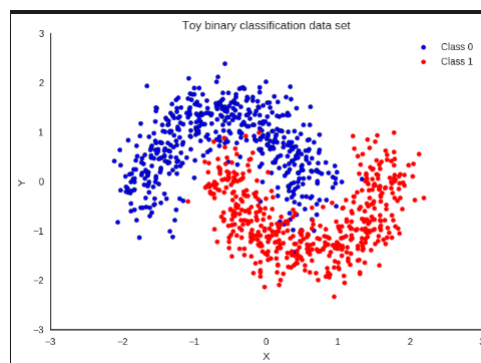


Figure 2.1: Binary Classification

### 2.1.2 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. Skenario 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.

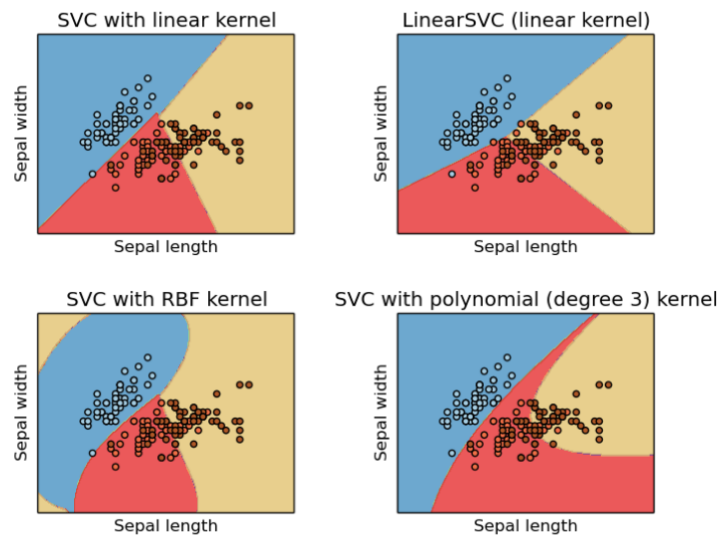


Figure 2.2: Supervised Learning

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.

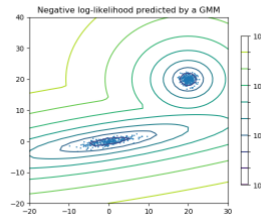


Figure 2.3: Unsupervised Learning

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.

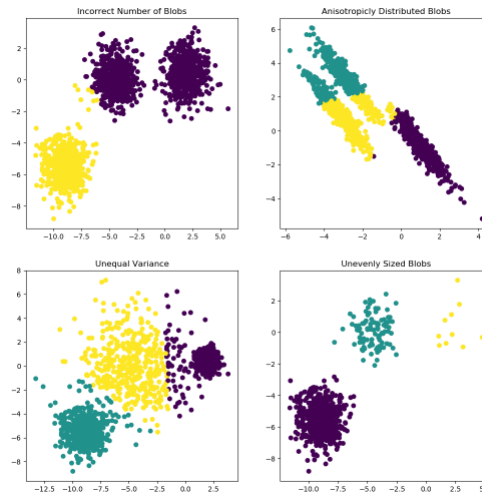


Figure 2.4: Cluster

### 2.1.3 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.

```
>>> from sklearn import svm, datasets
>>> from sklearn.model_selection import cross_val_score
>>> iris = datasets.load_iris()
>>> X, y = iris.data, iris.target
>>> clf = svm.SVC(gamma='scale', random_state=0)
>>> cross_val_score(clf, X, y, scoring='recall_macro',
...                 cv=5)
array([0.96..., 0.96..., 0.96..., 0.93..., 1.      ])
>>> model = svm.SVC()
>>> cross_val_score(model, X, y, cv=5, scoring='wrong_choice')
```

Figure 2.5: Evaluasi dan Akurasi

### 2.1.4 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 =  $(1+1)/(5+1+1+3) = 0,2$

### **2.1.5 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.

### **2.1.6 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 sederhana yang disimpulkan dari fitur data.

Misalnya, dalam contoh di bawah ini, decision tree belajar dari data untuk

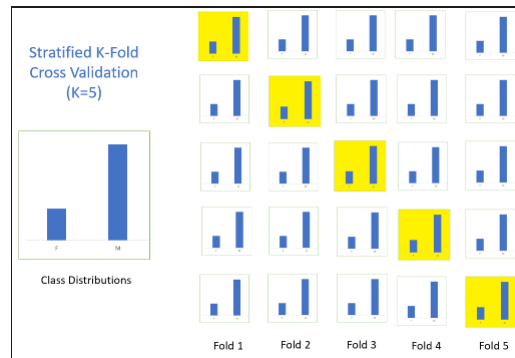


Figure 2.6: K-fold cross validation

memperkirakan kurva sinus dengan seperangkat aturan keputusan if-then-else. Semakin dalam pohon, semakin rumit aturan keputusan dan semakin bugar modelnya.

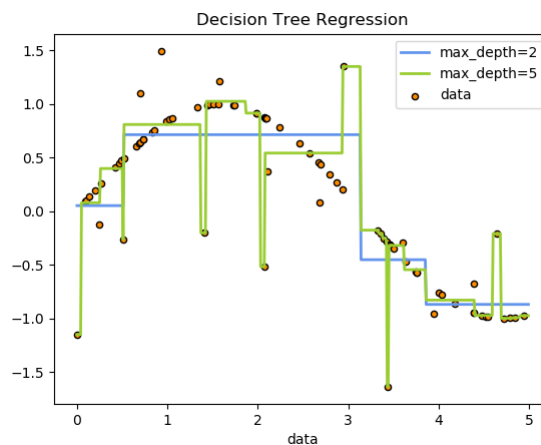


Figure 2.7: Decision Tree

### 2.1.7 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).
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

```

from scipy.stats import entropy
import numpy as np

def information_gain(X, y):

    def _entropy(labels):
        counts = np.bincount(labels)
        return entropy(counts, base=None)

    def _ig(X, y):
        # indices where x is set/not set
        x_set = np.nonzero(X)[1]
        x_not_set = np.delete(np.arange(X.shape[1]), x_set)

        h_x_set = _entropy(y[x_set])
        h_x_not_set = _entropy(y[x_not_set])

        return entropy_full - (((len(x_set) / f_size) * h_x_set)
                                + ((len(x_not_set) / f_size) * h_x_not_set))

    entropy_full = _entropy(y)

    f_size = float(X.shape[0])

    scores = np.array([_ig(x, y) for x in X.T])
    return scores

```

Figure 2.8: Information gain

dari sejumlah lemparan sulit diprediksi. Mengapa? Karena tidak ada hubungan antara membalik dan hasilnya. Inilah inti dari entropi.

```

import matplotlib.pyplot as plt
import numpy as np

from skimage import data
from skimage.util import img_as_ubyte
from skimage.filters.rank import entropy
from skimage.morphology import disk

# First example: object detection.

noise_mask = np.full((128, 128), 28, dtype=np.uint8)
noise_mask[32:-32, 32:-32] = 30

noise = (noise_mask * np.random.random(noise_mask.shape) - 0.5 *
         noise_mask).astype(np.uint8)
img = noise + 128

entr_img = entropy(img, disk(10))

fig, (ax0, ax1, ax2) = plt.subplots(nrows=1, ncols=3, figsize=(10, 4))

ax0.imshow(noise_mask, cmap='gray')
ax0.set_xlabel("Noise mask")
ax1.imshow(img, cmap='gray')
ax1.set_xlabel("Noisy image")
ax2.imshow(entr_img, cmap='viridis')
ax2.set_xlabel("Local entropy")

fig.tight_layout()

```

Figure 2.9: Entropi

# 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



# 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.