

Tugas Mata Kuliah Kecerdasan Buatan Chapter 2

March 27, 2022

Muhammad Nazhim Maulana (1194025)

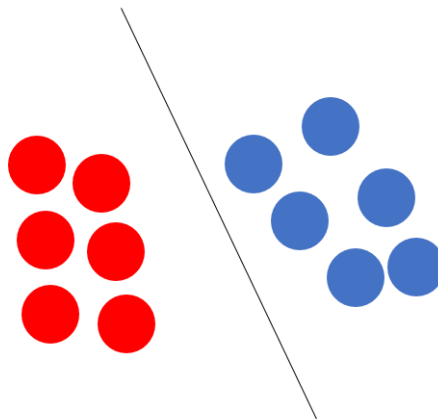
Binary Classification

Binary Classification merupakan salah satu bentuk klasifikasi yang dimana data-data yang diklasifikasikan itu hanya dibagi menjadi dua buah kelas saja. Jadi klasifikasi ini sama halnya dengan memprediksi dikelompok manakah satu benda itu berada. Klasifikasi ini hanya membagi data menjadi dua kelompok saja dan tidak lebih.

Tabel Ilustrasi *Binary Classification*

No.	Aplikasi	Label A	Label B
1	Diagnosa Kesehatan	Sehat	Sakit
2	Analisa Email	Tidak Spam	Spam
3	Analisa Keuangan	Tidak Curang	Curang
4	Pemasaran	Tidak Beli	Beli

Figure 1: Gambar Ilustrasi



Pada tabel dan juga gambar diatas dapat dilihat dengan jelas bagaimana *Binary Classification* itu. Dapat dilihat bahwa pembagian kelompok setiap benda pasti hanya akan menjadi dua saja apakah A atau B dan tidak ada satu kelas tambahan yang lainnya.

Supervised, Unsupervised Learning, dan Clustering

Supervised Learning adalah salah satu jenis dari algoritma yang digunakan

untuk mengajar sebuah *Machine Learning*, untuk pembelajarannya ini akan diawasi langsung oleh seorang *Supervisor* atau pengawas. Algoritma ini akan memerlukan sebuah data berlabel untuk membangun model yang memiliki tingkat akurasi yang terus menerus dapat ditingkatkan.

Seperti yang telah dijelaskan pada chapter satu sebelumnya, *Unsupervised Learning* merupakan kebalikan dari *Supervised Learning*. Jikalau *Supervised Learning* itu memiliki pengawas, maka *Unsupervised Learning* sendiri tidak diawasi. Dengan begitu algoritma ini cenderung lebih bebas dalam proses eksplorasi data karena tiap data yang ada tidak memiliki label sehingga lebih mudah dalam melakukan eksplorasi dan menemukan data yang tersembunyi.

Figure 2: Analogi Supervised Learning

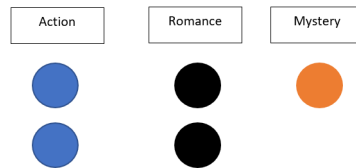


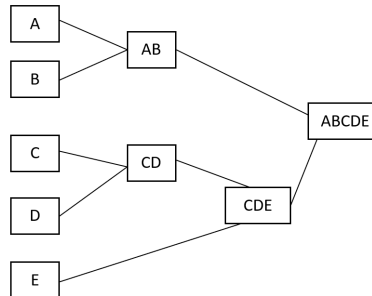
Figure 3: Analogi Unsupervised Learning



Perhatikan dua buah gambar diatas. keduanya merupakan analogi dari *Supervised Learning* dan *Unsupervised Learning*. Ketika mendownload beberapa film kemudian menyimpannya ke dalam folder berdasarkan genre, dengan begitu kita bisa langsung dengan mudah menyimpan film baru ke dalam folder yang di awal tadi telah dibuat karena genre dari filmnya juga diketahui.

Itu untuk *Supervised Learning*, sedangkan untuk *Unsupervised Learning*, genre tadi tidak ada maka film-film akan dikelompokkan secara acak karena tidak ada label yang disediakan untuk membedakan film-film itu maka dibuatlah label-label baru seperti Bagus, kurang bagus dan sebagainya. Dari penjelasan tadi maka dapat dilihat dengan cukup jelas mengenai perbedaan antara *Supervised* dan *Unsupervised Learning*

Figure 4: Ilustrasi clustering



Clustering adalah proses pengelompokan data ke dalam beberapa *cluster* atau kelompok, dengan begitu dalam satu *cluster* memiliki tingkat kemiripan yang maksimum dan data antar *cluster* yang berbeda memiliki kemiripan minimum. Seperti pada gambar diatas yang merupakan salah satu contoh clustering dari beberapa huruf. Pertama semua huruf berada pada kelompok yang berbeda-beda kemudian di kelompokkan lagi hingga ada kelompok dimana semua huruf ada.

Evaluasi dan Akurasi

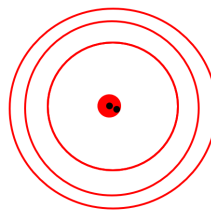


Figure 5: Accuracy

Berdasarkan buku, evaluasi adalah penilaian yang dilakukan untuk mengukur seberapa baik keakuratan dari sebuah model. Dengan adanya evaluasi maka kesalahan-kesalahan yang dibuat oleh model bisa dilihat dan juga di analisis. Untuk Akurasi sendiri, merupakan sebuah pengukuran yang penting untuk dilakukan selama melatih sebuah model atau mesin dan juga untuk melihat seberapa baik hasil akhir yang diperoleh dari sebuah mesin.

Membuat dan Membaca *Confusion Matrix*

Confusion Matrix adalah pengukuran performa klasifikasi *machine learning* dengan keluaran berupa dua kelas atau lebih. *Confusion Matrix* berisi tabel dengan empat kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada

empat istilah dalam *Confusion Matrix* yaitu:

1. *True Positive* (TP)
Artinya prediksi yang dibuat itu positif dan memang benar
2. *True Negative* (TN)
Artinya prediksi yang dibuat itu negatif dan memang benar
3. *False Positive* (FP)
Artinya prediksi yang dibuat itu positif dan salah
4. *False Negative* (FN)
Artinya prediksi yang dibuat itu negatif dan memang salah

Tabel Ilustrasi *Confusion Matrix*

n = 175	Aktual: Positif (1)	Negatif (0)
Prediksi: Positif (1)	TP: 125	FP: 20
Prediksi: Negatif (0)	FN: 25	TN: 5
	150	25

Sebagai sebuah disini ada sebuah perusahaan yang membuat model untuk membuat prediksi apakah karyawan di perusahaan tersebut terjangkit covid-19 atau tidak. Diasumsikan bahwa terdapat 175 karyawan dengan prediksi 145 karyawan dan yang negatif sebanyak 30 namun pada kenyataannya yang positif ada 150 dan yang negatif ada 25 orang.

K-fold *Cross Validation*

Cross Validation Merupakan sebuah teknik validasi dari pengambilan model *Split Validation* dimana validasinya mengukur *training error* dengan menguji data uji. Untuk contoh dari *Cross Validation* dapat dilihat seperti pada gambar tabel yang ada di bawah ini.

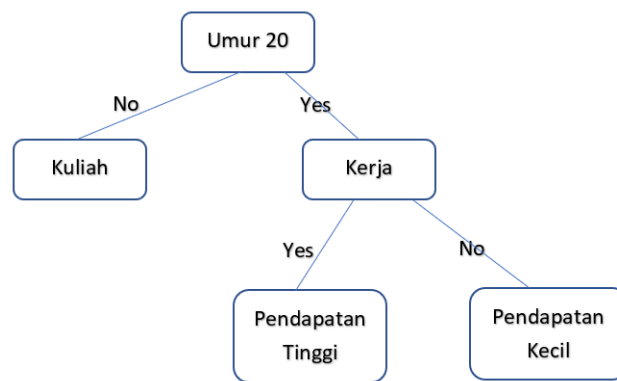
Tabel Ilustrasi *Cross Validation*

Percobaan 1	Test	Test	Test	Test
Percobaan 2	Test	Test	Test	Test
Percobaan 3	Test	Test	Test	Test
Percobaan 4	Test	Test	Test	Test

Decision Tree

Decision Tree Merupakan sebuah alat yang menjadi pendukung dengan struktur yang mirip dengan pohon untuk memodelkan kemungkinan hasil dan juga hingga kemungkinan konsekuensi yang mungkin dapat terjadi ketika mengambil atau memilih salah satu hal yang bisa saja berpengaruh dengan keadaan kita kedepannya. Untuk salah satu ilustrasinya dapat dilihat pada gambar berikut ini:

Figure 6: Contoh Decision Tree



Information Gain dan Entropi

Information Gain adalah ukuran efektivitas satu atribut dalam mengklasifikasi data dan biasanya digunakan untuk menentukan urutan atribut yang atributnya memiliki nilai informasi yang terbesar. Kemudian ada entropi yang merupakan informasi yang menyatakan ukuran ketidakpastian dari atribut dari satu kumpulan objek dalam satuan bit. Untuk setiap latihannya disini saya menggunakan variabel nama kota karena hasil mod dari NPM saya adalah 1 dan berikut adalah hasil-hasilnya:

1. Load Dataset

Berikut adalah hasil untuk load data setnya

Figure 7: Hasil Load Dataset

```
In [69]: runfile('D:/Pendidikan/Tugas/AI/Chapter 2.py', wdir='D:/Pendidikan/Tugas/AI')
395
In [70]:
```

2. Generate Binary Label

Untuk praktikum yang kedua hasilnya adalah sebagai berikut:

Figure 8: Menabhakan Binary Label

```

In [71]: runfile('D:/Pendidikan/Tugas/AI/Chapter 2.py', wdir='D:/Pendidikan/Tugas/AI')
school sex age address famsize ... Dalc Walc health absences pass
0 GP F 18 U GT3 ... 1 1 3 6 0
1 GP F 17 U GT3 ... 1 1 3 4 0
2 GP F 15 U LE3 ... 2 3 3 10 0
3 GP F 15 U GT3 ... 1 1 5 2 1
4 GP F 16 U GT3 ... 1 2 5 4 0
[5 rows x 31 columns]

```

3. *Use encode on categorical column*
Selanjutnya melangkah ke praktikum yang ketiga dengan hasil seperti ini:

Figure 9: Hasil

```

In [73]: runfile('D:/Pendidikan/Tugas/AI/Chapter 2.py', wdir='D:/Pendidikan/Tugas/AI')
age Medu Fedu ... internet_yes romantic_no romantic_yes
0 18 4 4 ... 0 1 0
1 17 1 1 ... 1 1 0
2 15 1 1 ... 1 1 0
3 15 4 2 ... 1 0 1
4 16 3 3 ... 0 1 0
[5 rows x 57 columns]

```

4. *Shuffle Rows*
Lanjut ke hasil praktikum yang selanjutnya yang hasilnya seperti pada gambar berikut ini:

```

In [75]: runfile('D:/Pendidikan/Tugas/AI/Chapter 2.py', wdir='D:/Pendidikan/Tugas/AI')
Passing: 166 out of 395 (42.03%)

```

5. *Visualize Tree*
Visulaize tree ini adalah memvisualisasikan decision tree yang sebelumnya itu telah dibuat

```

In [76]: runfile('D:/Pendidikan/Tugas/AI/Chapter 2.py', wdir='D:/Pendidikan/Tugas/AI')
digraph Tree {
    node [shape=box, style="filled, rounded", color="black", fontname=helvetica] ;
    edge [fontname=helvetica] ;
    0 [label="failures <= 0.5\nsamples = 100.0%\nvalue = [0.58, 0.42]\nnclass = fail",
    fillcolor="#f8dccc"] ;
    1 [label="schoolsup_no <= 0.5\nsamples = 79.0%\nvalue = [0.519, 0.481]\nnclass = fail",
    fillcolor="#fd6f0"] ;
    0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
    2 [label="Mjob_services <= 0.5\nsamples = 10.1%\nvalue = [0.825, 0.175]\nnclass = fail",
    fillcolor="#eb9c63"] ;
    1 -> 2 ;
    3 [label="Fedu <= 2.5\nsamples = 6.6%\nvalue = [0.923, 0.077]\nnclass = fail", fillcolor="#e78c49"] ;
    2 -> 3 ;
    4 [label="reason_reputation <= 0.5\nsamples = 2.3%\nvalue = [0.778, 0.222]\nnclass = fail",
    fillcolor="#eca572"] ;
    3 -> 4 ;
    5 [label="samples = 1.3%\nvalue = [1.0, 0.0]\nnclass = fail", fillcolor="#e58139"] ;
    4 -> 5 ;
}

```

6. *Praktikum 8*
Untuk praktikum yang ketujuh hasilnya itu berupa dokumen saja yang isinya itu adalah hasil dari
7. *Visualize Tree*
Visulaize tree ini adalah memvisualisasikan decision tree yang sebelumnya

itu telah dibuat pada *Visualize Tree* jadi langsung saja ke praktikum yang kedelapan yang hasilnya adalah seperti ini:

```
In [77]: runfile('D:/Pendidikan/Tugas/AI/Chapter 2.py', wdir='D:/Pendidikan/Tugas/AI')
0.7063291139240506
```

8. Accuracy

Untuk praktikum mengenai Accuracy, hasilnya adalah seperti berikut ini:

```
In [78]: runfile('D:/Pendidikan/Tugas/AI/Chapter 2.py', wdir='D:/Pendidikan/Tugas/AI')
Accuracy: 0.58 (+/- 0.08)
```

9. Praktikum 10

Untuk hasil dari praktikum 10 adalah sebagai berikut:

```
In [79]: runfile('D:/Pendidikan/Tugas/AI/Chapter 2.py', wdir='D:/Pendidikan/Tugas/AI')
Max depth: 1, Accuracy: 0.58 (+/- 0.01)
Max depth: 2, Accuracy: 0.55 (+/- 0.07)
Max depth: 3, Accuracy: 0.53 (+/- 0.10)
Max depth: 4, Accuracy: 0.54 (+/- 0.21)
Max depth: 5, Accuracy: 0.55 (+/- 0.11)
Max depth: 6, Accuracy: 0.52 (+/- 0.21)
Max depth: 7, Accuracy: 0.56 (+/- 0.21)
Max depth: 8, Accuracy: 0.53 (+/- 0.19)
Max depth: 9, Accuracy: 0.57 (+/- 0.17)
Max depth: 10, Accuracy: 0.58 (+/- 0.14)
Max depth: 11, Accuracy: 0.56 (+/- 0.18)
Max depth: 12, Accuracy: 0.57 (+/- 0.12)
Max depth: 13, Accuracy: 0.57 (+/- 0.16)
Max depth: 14, Accuracy: 0.59 (+/- 0.15)
Max depth: 15, Accuracy: 0.58 (+/- 0.14)
Max depth: 16, Accuracy: 0.58 (+/- 0.17)
Max depth: 17, Accuracy: 0.59 (+/- 0.14)
Max depth: 18, Accuracy: 0.61 (+/- 0.15)
```

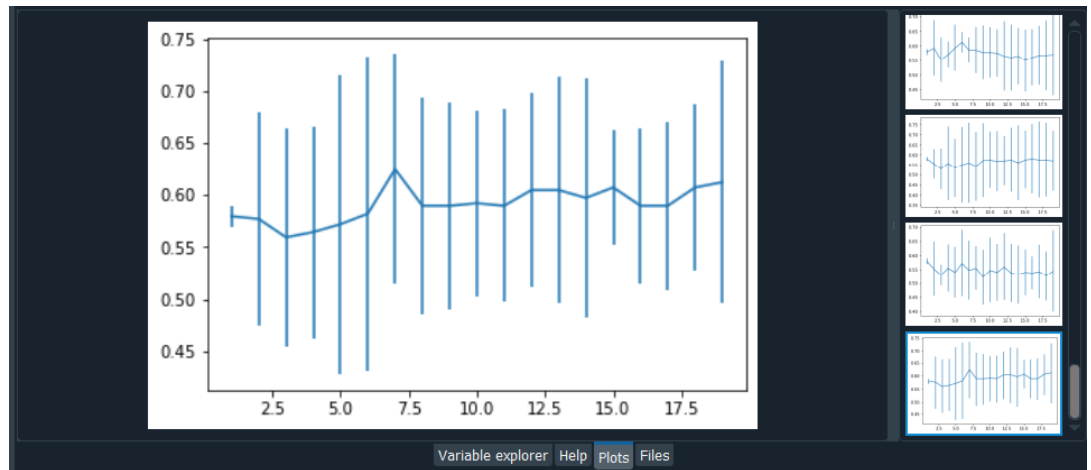
10. Praktikum 11

Praktikum 11 ini hasilnya adalah sebagai berikut:

```
In [80]: runfile('D:/Pendidikan/Tugas/AI/Chapter 2.py', wdir='D:/Pendidikan/Tugas/AI')
[[1.00000000e+000 5.79746835e-001 1.01265823e-002]
 [1.48219694e-323 1.97626258e-323 2.47032823e-323]
 [2.96439388e-323 3.45845952e-323 3.95252517e-323]
 [4.44659081e-323 4.94065646e-323 5.43472210e-323]
 [5.92878775e-323 0.00000000e+000 4.94065646e-324]
 [9.88131292e-324 1.48219694e-323 1.97626258e-323]
 [2.47032823e-323 2.96439388e-323 3.45845952e-323]
 [3.95252517e-323 4.44659081e-323 4.94065646e-323]
 [5.43472210e-323 5.92878775e-323 6.42285340e-323]
 [6.91691904e-323 7.41098469e-323 7.90505033e-323]
 [8.39911598e-323 8.89318163e-323 9.38724727e-323]
 [9.88131292e-323 1.03753786e-322 1.08694442e-322]
 [1.13635099e-322 1.18575755e-322 1.23516411e-322]
 [1.28457068e-322 1.33397724e-322 1.38338381e-322]
 [1.43279037e-322 1.48219694e-322 1.53160350e-322]
 [1.58101007e-322 1.63041663e-322 1.67982320e-322]
 [1.72922976e-322 1.77863633e-322 1.82804289e-322]
 [1.87744945e-322 1.92685602e-322 1.97626258e-322]]
```


11. Praktikum 12

Praktikum 12 ini akan dibuat sebuah diagram dan hasilnya adalah sebagai berikut:



Jadi untuk chapter 2 mulai dari teori hingga praktikum scikit-learnnya sudah selesai dikerjakan.