

Tugas Chapter 2

Artificial Intelligence



Dyning Aida Batrishya

1184030

D4 Teknik Informatika 3B

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

Assalamualaikum warahmatullahi wabarakaatuh. Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga dapat menyelesaikan laporan Tugas Chapter 2 ini, tanpa bantuan-Nya maka penulis tidak dapat menyelesaikannya dengan baik dan tepat pada waktunya. Shalawat serta salam semoga terlimpahkan kepada junjungan Nabi Muhammad SAW yang akan kita nantikan syafaatnya di yaumul qimayah nanti.

Laporan ini disusun guna memenuhi kelulusan matakuliah Artificial Intelligence Program Studi DIV Teknik Informatika. Proses penyelesaian laporan ini tidak luput dari bantuan berbagai pihak. Oleh karenanya, penulis mengucapkan terima kasih kepada :

1. Allah SWT yang telah memberikan rahmat dan hidayah-Nya
2. Orang tua yang selalu memberikan dukungan dan motivasi dalam penyelesaian laporan
3. Bapak Rolly Awangga yang telah memberikan dukungan dan bantuan dalam penyelesaian laporan
4. Teman-teman yang saya sayangi yang selalu memberikan dukungan dan motivasinya kepada penulis

Penulis mengharapkan kritik dan saran dari pembaca jika terdapat kesalahan dalam penyusunan laporan ini sehingga penulis dapat memperbaiki penyelesaian tugas yang selanjutnya

Bandung, 2 Mei 2021

Penulis

Chapter 1

Membangun Model Prediksi

1.1 Teori

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

1. binary classificarion ialah suatu bentuk klasifikasi supervised learning dengan berdasarkan label class, dimana pada binary classification, terdapat satu atribut class yang berisikan 2 value. Value tersebut dapat dipresentasikan sebagai nilai positif/negatif, lulus/tidak, true/false, 0/1, spam/tidak, dan sebagainya. hal ini disebut juga sebagai binary classifier.

tujuan digunakannya binary classification ialah untuk dapat mencari batas data secara optimal berdasarkan kelas yang ditentukan. biasanya, binary classifier ini digunakan sebagai label suatu tabel, namun tidak menutup kemungkinan bahwa dalam suatu tabel terdapat lebih dari 1 binary classifier, karena dengan penggunaan binary classifier ini dapat memudahkan dalam menentukan label dari suatu tabel.

contoh implementasi binary classification yaitu di antaranya :

- (a) ketika hendak mengklasifikasikan email, apakah email tersebut termasuk ke dalam spam atau tidak spam
 - (b) ketika hendak mengklasifikasikan mahasiswa, apakah lulus atau tidak lulus, berdasar atribut tabel yang lainnya
2. Jelaskan apa itu supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar sendiri.

supervised learning Supervised learning merupakan suatu algoritma yang digunakan untuk menentukan suatu prediksi dan klasifikasi berdasarkan variabel x dan variabel y telah diketahui. Algoritma ini seolah-olah sudah dilatih sehingga dapat menghasilkan suatu bentuk prediksi dan klasifikasi. Supervised learning menggunakan label di setiap datanya, data yang diklasifikasi dapat berupa klasifikasi (contoh : "lulus", "tidak lulus") ataupun regresi (periode kuliah, nilai). Algoritma yang dapat digunakan pada supervised learning di antaranya yaitu :

- (a) Classification and Regression
- (b) Logistic Regression
- (c) Time Series
- (d) Model Ensemble

salah satu contohnya, ketika dimiliki data klasifikasi hewan, dengan fitur bentuk gigi, bentuk kuku, ketahanan suhu dan label jenis dengan value karnivora, herbivora, dan omnivora. dari data tersebut, diklasifikasikan dengan metode classification karena data yang dimiliki ialah data yang bersifat categorical.

Unsupervised learning yaitu suatu algoritma penentuan prediksi dimana data tidak memiliki output/target variabelnya (label), sehingga tidak yang mengendalikan jalannya algoritma suatu program. Berbeda dengan supervised learning, algoritma ini tidak perlu dilatih dahulu untuk dapat menghasilkan prediksi maupun klasifikasi. cara kerja algoritma unsupervised ini dilakukan dengan menggunakan perhitungan kesamaan dari atribut yang dimiliki, ketika setiap atribut dan sifatnya diekstrak dan memiliki kemiripan atau kemiripan terdekat, maka akan dikelompokkan menjadi 1 kelompok (cluster).

Algoritma yang dapat digunakan pada metode ini di antaranya yaitu :

- (a) Clustering
- (b) Training Model
- (c) Anomaly Detection
- (d) Association Discovery

Contohnya yaitu diketahui data gambar hewan, maka dari data gambar tersebut akan diekstrak dalam machine learning, untuk kemudian dilakukan clustering berdasar kemiripan yang dimiliki di setiap data gambar tersebut.

Clustering ialah suatu metode yang digunakan untuk pengelompokan data berdasarkan kemiripan data tersebut setelah diekstrak, clustering ini umumnya digunakan untuk metode unsupervised learning. contohnya yaitu K-Means Clustering, KNN.

3. Evaluasi dan Akurasi

- Evaluasi

Evaluasi ialah suatu kegiatan/proses yang bertujuan untuk dapat menentukan/membuat keputusan berdasarkan sejauh mana tujuan program telah tercapai.

Menurut Abdul Basir (1996), evaluasi ialah suatu proses pengimpulan data secara deskriptif, informatif dan prediktif yang dilaksanakan dengan sistematis dan bertahap sehingga dapat menentukan kebijaksanaan dalam usaha untuk memperbaiki pendidikan.

- Akurasi

Akurasi ialah perbandingan prediksi benar (baik positif atau negatif) terhadap keseluruhan data yang digunakan.

contohnya, seperti pada confusion matriks di bawah ini

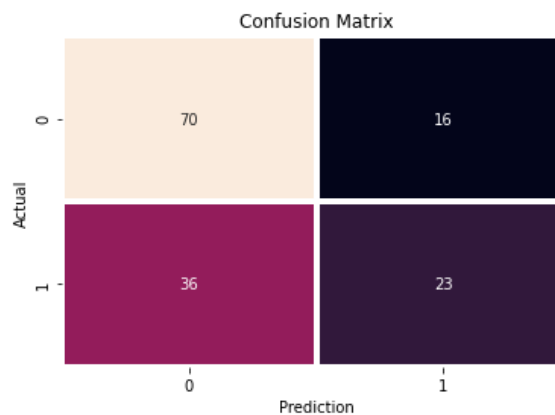


Figure 1.1: Contoh

dari confusion matrix tersebut, didapatkan

- $TP = 70$
- $FP = 16$
- $TN = 36$

$$- FN = 23$$

yang berarti total data prediksi benar yakni $TP + FN = 70 + 23 = 93$ data. oleh karenanya, akurasi bisa didapatkan dengan

$$\begin{aligned} 1 \text{ Akurasi} &= (TP + TN) / (TP+FP+FN+TN) * 100\% \\ 2 \text{ Akurasi} &= 93/145*100\% \\ 3 \text{ Akurasi} &= 64\% \end{aligned}$$

4. Confusion Matrix

confusion matrix merupakan salah satu model klasifikasi yang digunakan untuk mengukur kinerja suatu model atau klasifikasi.

confusion matriks memberikan perbandingan hasil klasifikasi yang dihasilkan oleh model, dengan hasil klasifikasi yang sebenarnya dengan bentuk tabel matriks yang berisikan kombinasi nilai prediksi dan nilai aktual seperti berikut :

		Actual Values	
		1 (Postive)	0 (Negative)
Predicted Values	1 (Postive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Figure 1.2: Tabel Confusion Matriks

confusion matrix dibagi menjadi 2, yakni :

(a) binary classification confusion matrix

ialah confusion matrix yang hanya memiliki 2 value pada label yang digunakan. contohnya ya/tidak, lulus/tidak lulus, baik/buruk, recommend/not recommend.

(b) multiclass classification confusion matrix

ialah confusion matrix yang memiliki multiclass classification pada labelnya, yakni valuenya lebih dari 2. contohnya yaitu sangat buruk/buruk/baik/sangat baik, A/B/C/D/E, sangat tidak recommend/recommend/sangat recommend.

berikut ini kode program untuk membuat visualisasi confusion matrix dengan matplotlib dan seaborn

```
1 import pandas as pd, seaborn as sns, matplotlib.pyplot as plt,
   numpy as np
2 from sklearn.metrics import confusion_matrix
3 sns.heatmap(confusion_matrix(d_jeruk_test_pass, y_pred), annot=True,
   linewidths=3, cbar=False)
4 plt.title('Confusion Matrix')
5 plt.ylabel('Actual')
6 plt.xlabel('Prediction')
7 plt.show()
```

berikut ini contoh hasil confusion matrixnya :

(a) binary

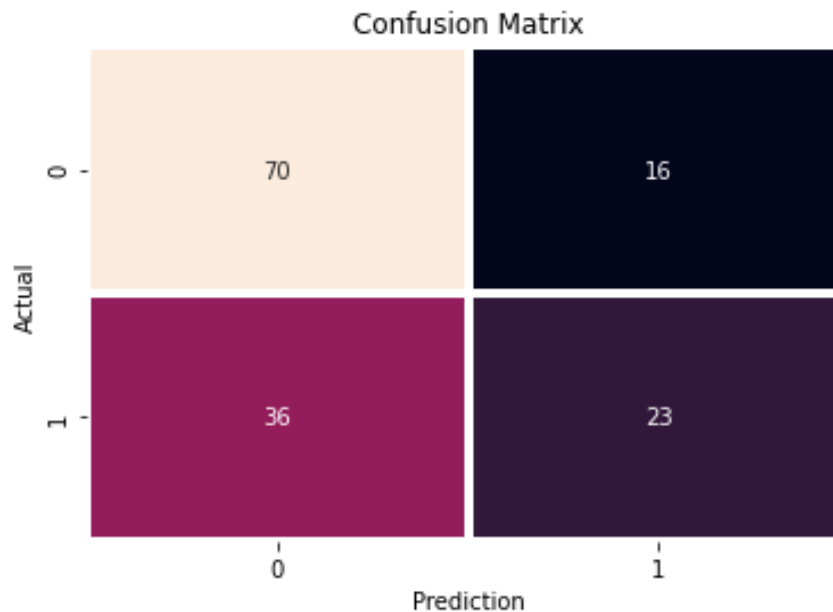


Figure 1.3: Hasil Visualisasi Binary Confusion Matriks

dari confusion matrix tersebut, didapatkan :

- $TP = 70$
- $FP = 16$
- $TN = 36$
- $FN = 23$

(b) multiclass

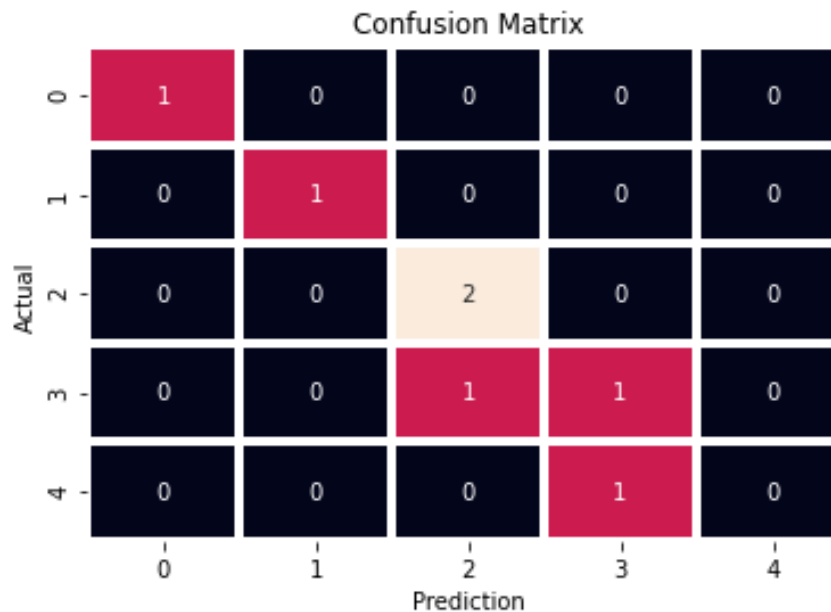


Figure 1.4: Hasil Visualisasi Multiclass Confusion Matriks

Dari confusion matriks tersebut, didapat:

- Nilai actual dan nilai predict = 0 berjumlah 1 data
- Nilai actual dan nilai predict = 1 berjumlah 1 data
- Nilai actual dan nilai predict = 1 berjumlah 1 data
- Nilai actual dan nilai predict = 2 berjumlah 2 data
- Nilai actual = 3, namun nilai predict = 2 berjumlah 1 data
- Nilai actual dan nilai predict = 3, berjumlah 1 data
- Nilai actual = 4, namun nilai predict = 3 berjumlah 1 data

Berikut hasil value metrics True Positive, False Positive, True Negative, dan False Negative dari confusion matrix di atas :

- Huruf = 0 (A), memiliki representasi :
 - TP0 = 1
 - FP0 = 0 + 0 + 0 + 0 = 0
 - TN0 = 1 + 0 + 0 + 0 + 0 + 2 + 0 + 0 + 0 + 1 + 1 + 0 + 0 + 0 + 1 + 0 = 6
 - FN0 = 0 + 0 + 0 + 0 = 0
- Huruf = 1 (B), memiliki representasi :
 - TP1 = 1

- ii. $FP1 = 0 + 0 + 0 + 0 = 0$
- iii. $TN1 = 1 + 0 + 0 + 0 + 0 + 2 + 0 + 0 + 0 + 1 + 1 + 0 + 0 + 0 + 1 + 0 = 6$
- iv. $FN1 = 0 + 0 + 0 + 0 = 0$
- Huruf = 2 (C), memiliki representasi :
 - i. $TP2 = 2$
 - ii. $FP2 = 0 + 0 + 0 + 0 = 0$
 - iii. $TN2 = 1 + (0*3) + 0 + 1 + (0*2) + (0*2) + 1 + 0 + (0*2) + 1 + 0 = 4$
 - iv. $FN2 = 0 + 0 + 1 + 0 = 1$
- Huruf = 3 (D), memiliki representasi :
 - i. $TP3 = 1$
 - ii. $FP3 = 0 + 0 + 1 + 0 = 1$
 - iii. $TN3 = 1 + (0*3) + 0 + 1 + (0*2) + (0*2) + 2 + (0*2) + (0*4) = 4$
 - iv. $FN3 = 0 + 0 + 0 + 1 = 1$
- Huruf = 4 (huruf E), memiliki representasi :
 - i. $TP4 = 0$
 - ii. $FP4 = 0 + 0 + 0 + 1 = 1$
 - iii. $TN4 = 1 + (0*3) + 0 + 1 + (0*2) + (0*2) + 2 + (0*2) + 1 + 1 = 6$
 - iv. $FN4 = 0 + 0 + 0 + 0 = 0$

5. K-fold Cross Validation Jelaskan bagaimana K-fold cross validation bekerja dengan gambar ilustrasi contoh buatan sendiri.

k-fold cross validation ialah metode tambahan yang digunakan untuk dapat diperoleh hasil akurasi yang lebih maksimal. dinamakan k-fold cross validation karena dilakukan percobaan sebanyak k kali dengan menggunakan model dan parameter yang sama.

cross validation juga ialah teknik pengukuran validasi yang merupakan bentuk pengembangan dari model Split validation, yakni dilakukan dengan mengukur training error dengan melakukan uji menggunakan data testing. berikut ini fungsi dari digunakannya k-fold cross validation, yakni :

- (a) mengukur performa suatu model dengan melalui percobaan sebanyak k kali
- (b) meningkatkan tingkat performansi model tersebut
- (c) mengolah dataset dengan menggunakan kelas yang seimbang
- (d) pengambilan sampel test yang lebih terstruktur
- (e) menjangkau pengujian yang lebih efisien

berikut ini contoh implementasi k-fold cross validation, seperti pada tabel berikut :

	Data					
Percobaan 1	Test	Train	Train	Train	Train	Train
Percobaan 2	Train	Test	Train	Train	Train	Train
Percobaan 3	Train	Train	Test	Train	Train	Train
Percobaan 4	Train	Train	Train	Test	Train	Train
Percobaan 5	Train	Train	Train	Train	Test	Train
Percobaan 6	Train	Train	Train	Train	Train	Test

Figure 1.5: contoh K-fold cross validation

dari contoh gambar di atas, dapat dibaca seperti berikut :

- pada percobaan 1, data 1 digunakan untuk data testing, sisanya digunakan sebagai data training
- pada percobaan 2, data 2 digunakan untuk data testing, sisanya digunakan sebagai data training
- pada percobaan 3, data 3 digunakan untuk data testing, sisanya digunakan sebagai data training
- pada percobaan 4, data 4 digunakan untuk data testing, sisanya digunakan sebagai data training
- pada percobaan 5, data 5 digunakan untuk data testing, sisanya digunakan sebagai data training
- pada percobaan 6, data 6 digunakan untuk data testing, sisanya digunakan sebagai data training

sehingga, dapat disimpulkan bahwa, pada setiap percobaan testing validation selama k kali, data testing yang digunakan berbeda, begitu pula dengan data trainnya, sehingga memungkinkan agar semua data dapat dilakukan uji validasi supaya memaksimalkan akurasi dari model yang digunakan.

6. Jelaskan apa itu decision tree dengan gambar ilustrasi contoh buatan sendiri.
Decision tree ialah suatu model klasifikasi yang digunakan untuk melakukan pengambilan keputusan dengan menggunakan struktur pohon/hierarki. metode ini menggabungkan 2 jenis pohon, yakni

(a) Classification tree

(b) Regression tree

berikut ini contoh decision tree

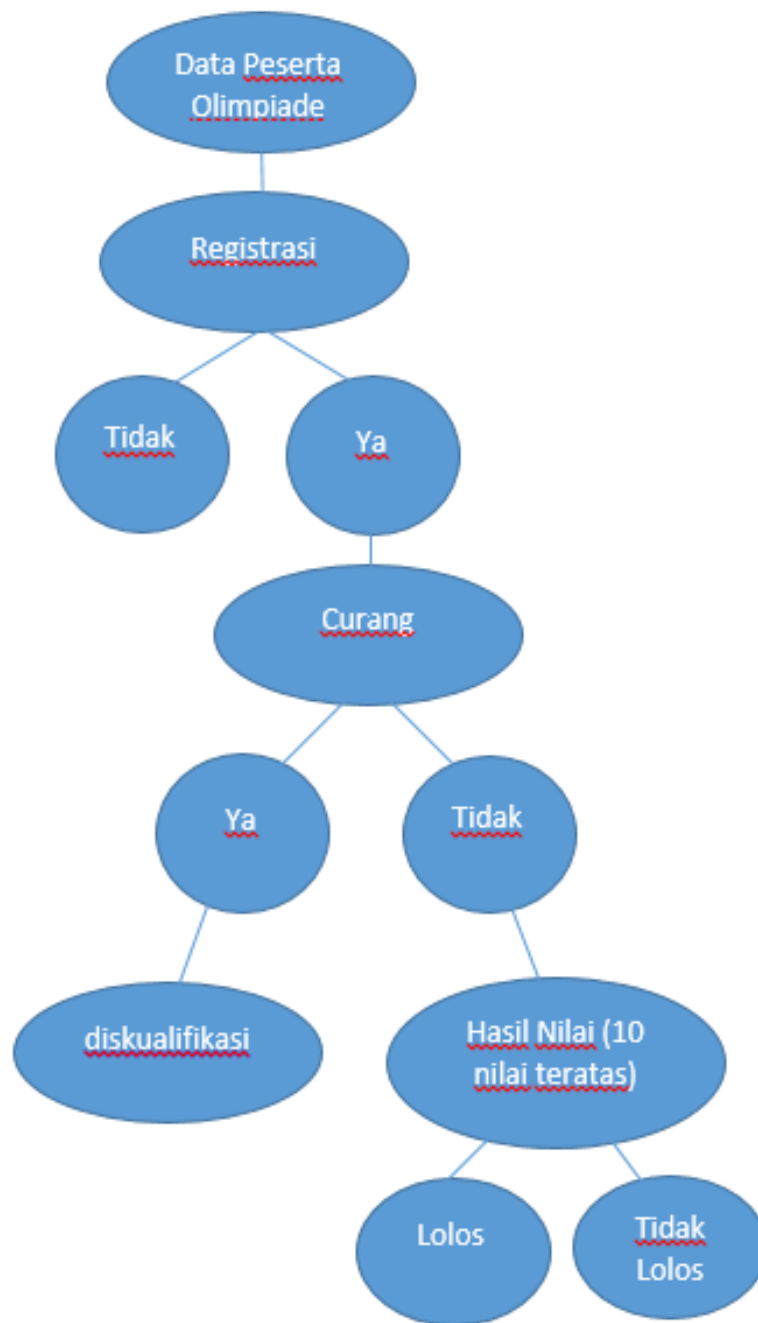


Figure 1.6: contoh decision dengan classification tree

7. Information Gain dan Entropy

(a) Information Gain

Information Gain ialah salah satu metode yang digunakan untuk seleksi fitur, yakni mengukur efektifitas dari atribut yang digunakan dalam melakukan pengklasifikasian data.

(b) Entropy

Entropy ialah suatu nilai yang berisikan informasi ukuran ketidakpastian (impurity) atribut dalam kumpulan objek data dalam satuan bit. semakin sedikit value dari atribut label, maka makin kecil pula nilai entropy yang dihasilkan. Sebaliknya, apabila nilai label multiclass, maka semakin besar pula nilai entropy yang dihasilkan.

Contohnya, ketika dataset A dengan label yang bernilai positif dan negatif, dibandingkan dengan dataset B yang memiliki label tidak direkomendasi, direkomendasikan, dan sangat direkomendasikan. Maka, dari contoh tersebut, entropy dari dataset A lebih kecil dibandingkan dengan dataset B.

Contoh 1	Contoh 2
Yes	Not Recommend
No	Very Recommend
Yes	Recommend
Yes	Recommend
No	Not Recommend

Figure 1.7: contoh atribut

dengan contoh tabel di atas, atribut contoh 1 akan memiliki nilai entropy yang lebih kecil dibandingkan dengan atribut contoh 2.

1.2 Praktikum Scikit-learn

Dataset ambil di <https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners> folder Chapter01. Tugas anda adalah, dataset ganti menggunakan **student-mat.csv** dan mengganti semua nama variabel dari kode di bawah ini dengan nama-nama makanan (NPM mod 3=0), kota (NPM mod 3=1), buah (NPM mod 3=2), . Jalankan satu per satu kode tersebut di spyder dengan menggunakan textitRun current cell. Kemudian Jelaskan dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat dan wajib skrinsut dari komputer sendiri masing masing nomor di bawah ini(nilai 5 masing masing pada hari kedua).

```

1. # NPM = 1184030
2. # NPM%3
3. # import library pandas
4. import pandas as pd

```

```

5 # load dataset student-mat.csv
6 d_apel = pd.read_csv('student-mat.csv', sep=';')
7 # menghitung length dataset csv
8 len(d_apel)
9 # generate binary label (pass/fail) berdasar nilai G1+G2+G3,
  apabila total >= 35, maka bernilai 1, jika tidak maka 0
10 d_apel['pass'] = d_apel.apply(lambda row: 1 if (row['G1']+row['G2']
  ]+row['G3'])
11                               >= 35 else 0, axis=1)
12 # drop row G1, G2 dan G3
13 d_apel = d_apel.drop(['G1', 'G2', 'G3'], axis=1)
14 # menampilkan 5 data teratas
15 d_apel.head()
16 # use one-hot encoding on categorical columns
17 d_apel = pd.get_dummies(d_apel, columns=['sex', 'school', 'address',
  ,
18                               'famsize',
19                               'Pstatus', 'Mjob', 'Fjob', 'reason', 'guardian', '
  schoolsup',
20                               'famsup', 'paid', 'activities', 'nursery', 'higher',
  'internet',
21                               'romantic'])
22 # menampilkan 5 data teratas
23 d_apel.head()
24 # shuffle rows
25 d_jeruk = d_apel.sample(frac=1)
26 # split training and testing data
27 d_jeruk_train = d_jeruk[:250]
28 d_jeruk_test = d_jeruk[250:]
29 # train atribut drop row pass
30 d_jeruk_train_att = d_jeruk_train.drop(['pass'], axis=1)
31 # train label menggunakan row pass
32 d_jeruk_train_pass = d_jeruk_train['pass']
33 # test atribut drop row pass
34 d_jeruk_test_att = d_jeruk_test.drop(['pass'], axis=1)
35 # test label menggunakan row pass
36 d_jeruk_test_pass = d_jeruk_test['pass']
37 # atribut drop row pass
38 d_jeruk_att = d_jeruk.drop(['pass'], axis=1)
39 # menggunakan row pass untuk label
40 d_jeruk_pass = d_jeruk['pass']
41
42 # import library
43 import numpy as np
44 # print number of passing students in whole dataset:
45 print("Passing: %d out of %d (%.2f%%)" % (np.sum(d_jeruk_pass), len
  (d_jeruk_pass),
46       100*float(np.sum(d_jeruk_pass)) / len(d_jeruk_pass)))
47
48 # import library
49 from sklearn import tree
50 # instansiasi desicion tree classifier
51 melon = tree.DecisionTreeClassifier(criterion="entropy", max_depth
  =5)

```

```

52 # fit decision tree
53 melon = melon.fit(d_jeruk_train_att, d_jeruk_train_pass)
54 # import library graphviz untuk visualisasi
55 import graphviz
56 # instansiasi graphviz dari fit decision tree sebelumnya
57 mangga = tree.export_graphviz(melon, out_file=None, label="all",
58                               impurity=False, proportion=True,
59                               feature_names=list(
60                                   d_jeruk_train_att),
61                                   class_names=["fail", "pass"],
62                                   filled=True, rounded=True)
63 # buat variabel grafik visualisasi tree
64 graph = graphviz.Source(mangga)
65 # jalankan visualisasinya
66 graph

```

```

1 # save tree
2 tree.export_graphviz(melon, out_file="student-performance.dot",
3                     label="all", impurity=False,
4                     proportion=True,
5                     feature_names=list(d_train_att),
6                     class_names=["fail", "pass"],
7                     filled=True, rounded=True)

```

3. # cek score

```

2 melon.score(d_jeruk_test_att, d_jeruk_test_pass)

```

```

1 # import cross val score untuk cek cross validation score
2 from sklearn.model_selection import cross_val_score
3 # cek cross validation score
4 angka_score = cross_val_score(melon, d_jeruk_att, d_jeruk_pass, cv
5                               =5)
6 # show average score and +/- two standard deviations away
7 # (covering 95% of scores)
8 # print akurasi
9 print("Accuracy: %0.2f (+/- %0.2f)" % (angka_score.mean(),
10                                       angka_score.std() * 2))

```

4. # buat akurasi max depth

```

2 for max_depth in range(1, 20):
3     melon = tree.DecisionTreeClassifier(criterion="entropy",
4     max_depth=max_depth)
5     angka_score = cross_val_score(melon, d_jeruk_att,
6     d_jeruk_pass, cv=5)
7     print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" %
8     (max_depth, angka_score.mean(), angka_score.std() * 2))

```

```

1 # buat akurasi depth acc
2 depth_acc = np.empty((19,3), float)
3 i = 0
4 for max_depth in range(1, 20):

```



```

5     melon = tree.DecisionTreeClassifier(criterion="entropy",
6     max_depth=max_depth)
7     angka_score = cross_val_score(melon, d_jeruk_att, d_jeruk_pass
8     , cv=5)
9     depth_acc[i,0] = max_depth
10    depth_acc[i,1] = angka_score.mean()
11    depth_acc[i,2] = angka_score.std() * 2
12    i += 1
13 # jalankan dept_acc
14 depth_acc

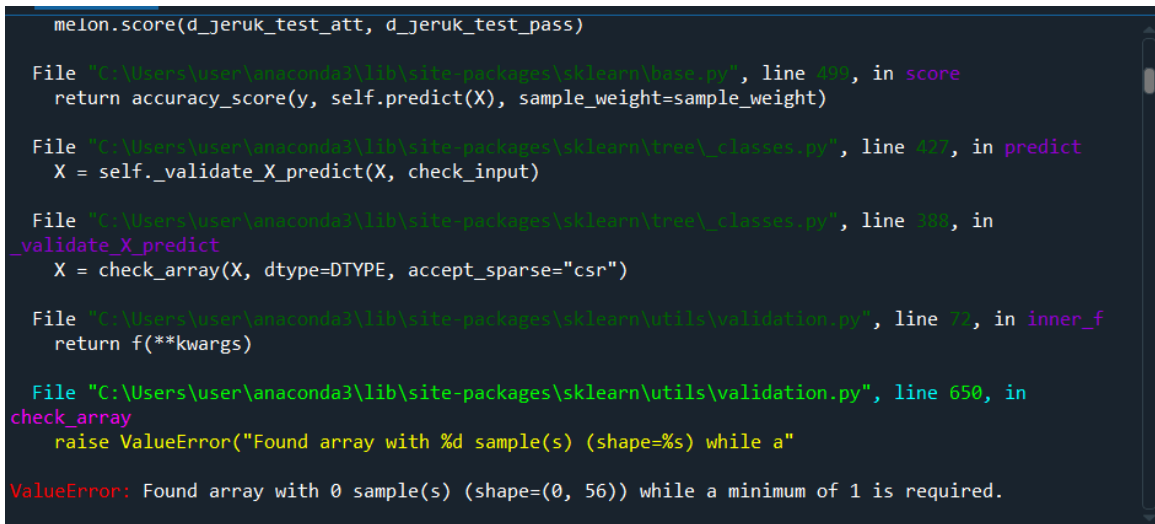
15 import matplotlib.pyplot as plt
16 fig, ax = plt.subplots()
17 ax.errorbar(depth_acc[:,0], depth_acc[:,1], yerr=depth_acc[:,2])
18 plt.show()

```

1.3 Penanganan Error

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

1. Screenshoot Error



```

melon.score(d_jeruk_test_att, d_jeruk_test_pass)

File "C:\Users\user\anaconda3\lib\site-packages\sklearn\base.py", line 499, in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)

File "C:\Users\user\anaconda3\lib\site-packages\sklearn\tree\_classes.py", line 427, in predict
    X = self._validate_X_predict(X, check_input)

File "C:\Users\user\anaconda3\lib\site-packages\sklearn\tree\_classes.py", line 388, in
_validate_X_predict
    X = check_array(X, dtype=DTYPE, accept_sparse="csr")

File "C:\Users\user\anaconda3\lib\site-packages\sklearn\utils\validation.py", line 72, in inner_f
    return f(**kwargs)

File "C:\Users\user\anaconda3\lib\site-packages\sklearn\utils\validation.py", line 650, in
check_array
    raise ValueError("Found array with %d sample(s) (shape=%s) while a"

ValueError: Found array with 0 sample(s) (shape=(0, 56)) while a minimum of 1 is required.

```

Figure 1.8: Error

2. Kode eror dan jenis errornya

```

1 ValueError: Found array with 0 sample(s) (shape=(0, 56)) while a
  minimum of 1 is required.

```

3. Solusi pemecahan masalah error

solusi yang dilakukan ialah dengan melakukan pengecekan kembali terhadap data yang dites dan ditrain, di source code yang sebelumnya, digunakan 500 data awal pada data train, dan setelah 500 data awal di data test, namun pada praktikum kali ini, data yang digunakan bahkan tidak lebih dari 500, melainkan hanya 395. sehingga memunculkan error tersebut. sehingga solusinya yakni mengubah banyak data yang displit untuk data train dan data test dengan menyesuaikan jumlah datanya, disini saya mengubah menjadi 250 data awal untuk data training, dan sisanya untuk data testingnya