

Modul Praktikum Kecerdasan Buatan



Rolly Maulana Awangga
0410118609

Applied Bachelor of Informatics Engineering
Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering
Politeknik Pos Indonesia

Bandung 2019

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

Acknowledgements

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

Abstract

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

Contents

1	Mengenal Kecerdasan Buatan dan Scikit-Learn	1
1.1	Teori	1
1.2	Instalasi	2
1.3	Penanganan Error	2
2	Membangun Model Prediksi	3
2.1	Teori	3
2.2	scikit-learn	4
2.3	Penanganan Error	7
3	Prediksi dengan Random Forest	8
3.1	Teori	8
3.1.1	Random Forest	9
3.1.2	Confusion Matrix	11
3.1.3	Mencoba dengan metode Decission Tree dan SVM	12
3.1.4	Pengecekan Cross Validation	13
3.1.5	Mengatahui komponen komponen informasi	13
3.2	Soal Teori	14
3.3	scikit-learn	15
3.4	Penanganan Error	17
4	Experiment and Result	19
4.1	Experiment	19
4.2	Result	19
5	Conclusion	20
5.1	Conclusion of Problems	20
5.2	Conclusion of Method	20
5.3	Conclusion of Experiment	20

5.4 Conclusion of Result	20
6 Discussion	21
7 Discussion	22
8 Discussion	23
9 Discussion	24
10 Discussion	25
11 Discussion	26
12 Discussion	27
13 Discussion	28
14 Discussion	29
A Form Penilaian Jurnal	30
B FAQ	33
Bibliography	35

List of Figures

A.1	Form nilai bagian 1.	31
A.2	form nilai bagian 2.	32

Chapter 1

Mengenai Kecerdasan Buatan dan Scikit-Learn

Buku umum teori lengkap yang digunakan memiliki judul *Artificial intelligence: a modern approach*[3]. Untuk pratikum sebelum UTS menggunakan buku *Python Artificial Intelligence Projects for Beginners*[1]. Buku pelengkap penunjang penggunaan python menggunakan buku *Python code for Artificial Intelligence: Foundations of Computational Agents*[2]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti definisi kecerdasan buatan, sejarah kecerdasan buatan, perkembangan dan penggunaan di perusahaan
2. Memahami cara instalasi dan pemakaian sci-kit learn
3. Memahami cara penggunaan variabel explorer di spyder

Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten riset.

1.1 Teori

Praktek teori penunjang yang dikerjakan :

1. Buat Resume Definisi, Sejarah dan perkembangan Kecerdasan Buatan, dengan bahasa yang mudah dipahami dan dimengerti. Buatan sendiri bebas plagiat[hari ke 1](10)
2. Buat Resume mengenai definisi supervised learning, klasifikasi, regresi dan unsupervised learning. Data set, training set dan testing set.[hari ke 1](10)

1.2 Instalasi

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

1. Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer[hari ke 1](10)
2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 1](10)
3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)
4. mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)
5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)

1.3 Penanganan Error

Dari percobaan yang dilakukan di atas, apabila mendapatkan error maka:

1. skrinsut error[hari ke 2](10)
2. Tuliskan kode eror dan jenis errornya [hari ke 2](10)
3. Solusi pemecahan masalah error tersebut[hari ke 2](10)

Chapter 2

Membangun Model Prediksi

Untuk pratikum saati ini menggunakan buku *Python Artificial Intelligence Projects for Beginners*[1]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Dataset ada di <https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners> . Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti implementasi klasifikasi
2. Memahami data set, training dan testing data
3. Memahami Decission tree.
4. Memahami information gain dan entropi.

Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten riset. Kode program menggunakan input listing ditaruh di folder src ekstensi .py dan dipanggil ke latex dengan input listings. Tulisan dan kode tidak boleh plagiat, menggunakan bahasa indonesia yang sesuai dengan gaya bahasa buku teks.

2.1 Teori

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

1. Jelaskan apa itu binary classification dilengkapi ilustrasi gambar sendiri
2. Jelaskan apa itu supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar sendiri.

3. Jelaskan apa itu evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar sendiri
4. Jelaskan bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix buatan sendiri.
5. Jelaskan bagaimana K-fold cross validation bekerja dengan gambar ilustrasi contoh buatan sendiri.
6. Jelaskan apa itu decision tree dengan gambar ilustrasi contoh buatan sendiri.
7. Jelaskan apa itu information gain dan entropi dengan gambar ilustrasi buatan sendiri.

2.2 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.

```
# load dataset (student mat pakenya)
import pandas as pd
d = pd.read_csv('student-mat.csv', sep=';')
len(d)
```
2.

```
# generate binary label (pass/fail) based on G1+G2+G3
# (test grades, each 0-20 pts); threshold for passing is sum>=30
d['pass'] = d.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3'])
>= 35 else 0, axis=1)
d = d.drop(['G1', 'G2', 'G3'], axis=1)
d.head()
```
3.

```
# use one-hot encoding on categorical columns
d = pd.get_dummies(d, columns=['sex', 'school', 'address',
```

```

    'famsize',
    'Pstatus', 'Mjob', 'Fjob',
                                'reason', 'guardian', 'schoolsup',
    'famsup', 'paid', 'activities',
                                'nursery', 'higher', 'internet',
    'romantic'])
d.head()

4. # shuffle rows
d = d.sample(frac=1)
# split training and testing data
d_train = d[:500]
d_test = d[500:]

d_train_att = d_train.drop(['pass'], axis=1)
d_train_pass = d_train['pass']

d_test_att = d_test.drop(['pass'], axis=1)
d_test_pass = d_test['pass']

d_att = d.drop(['pass'], axis=1)
d_pass = d['pass']

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

5. # fit a decision tree
from sklearn import tree
t = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
t = t.fit(d_train_att, d_train_pass)

6. # visualize tree
import graphviz
dot_data = tree.export_graphviz(t, out_file=None, label="all",
    impurity=False, proportion=True,

```

```

feature_names=list(d_train_att),
class_names=["fail", "pass"],
filled=True, rounded=True)

graph = graphviz.Source(dot_data)
graph

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

8. t.score(d_test_att, d_test_pass)

9. from sklearn.model_selection import cross_val_score
scores = cross_val_score(t, d_att, d_pass, cv=5)
# show average score and +/- two standard deviations away
#(covering 95% of scores)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))

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

11. depth_acc = np.empty((19,3), float)
i = 0
for max_depth in range(1, 20):
    t = tree.DecisionTreeClassifier(criterion="entropy",
max_depth=max_depth)
    scores = cross_val_score(t, d_att, d_pass, cv=5)
    depth_acc[i,0] = max_depth
    depth_acc[i,1] = scores.mean()

```

```

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

depth_acc

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

```

2.3 Penanganan Error

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

1. skrinsut error
2. Tuliskan kode eror dan jenis errornya
3. Solusi pemecahan masalah error tersebut

Chapter 3

Prediksi dengan Random Forest

Untuk pratikum saati ini menggunakan buku *Python Artificial Intelligence Projects for Beginners*[1]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Dataset ada di <https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners> . Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti implementasi klasifikasi dan teknik evaluasi
2. Memprediksi spesies burung dengan random forest
3. Memahami Confusion Matrix.

Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten riset. Kode program menggunakan input listing ditaruh di folder src ekstensi .py dan dipanggil ke latex dengan input listings. Tulisan dan kode tidak boleh plagiat, menggunakan bahasa indonesia yang sesuai dengan gaya bahasa buku teks.

3.1 Teori

Random Forest adalah hasil voting dari beberapa decision tree yang masing-masing memegang atribut yang berbeda. Jadi setiap decision tree spesifik terhadap atribut tersebut yang merupakan bagian kecil dari keseluruhan atribut di data set. Hindari RF jika atribut terlalu sedikit untuk membentuk beberapa tree. Pada praktek kali ini menggunakan dataset spesies burung yang diambil dari situs (<http://www.vision.caltech.edu/visip200-2011.html>). Didalmnya terdapat 12.000 foto dari 200 spesies yang berbeda. Yang akan kita pakai untuk RF hanya atribut dari burunynya saja seperti ukuran, bentuk

dan warna. Data tersebut diberi label secara manual oleh manusia dengan memanfaatkan jasa dari Amazon's Mechanical Turk.

3.1.1 Random Forest

Pertama dataset kita baca terlebih dahulu.

```
1 import pandas as pd
2
3 # some lines have too many fields (?), so skip bad lines
4 imgatt = pd.read_csv("data/CUB_200_2011/attributes/
    image_attribute_labels.txt",
5                      sep='\s+', header=None, error_bad_lines=False,
6                      warn_bad_lines=False,
7                      usecols=[0,1,2], names=['imgid', 'attid', 'present'
8                      ])
9
```

Listing 3.1: Fungsi Sederhana

Melihat sebagian data awal, dengan menggunakan listing 3.2.

```
1 imgatt.head()
```

Listing 3.2: Melihat sebagian data awal

Melihat jumlah data menggunakan listing 3.3.

```
1 imgatt.shape
```

Listing 3.3: Mengetahui jumlah data

Merubah atribut menjadi kolom dengan menggunakan pivot layaknya excel. lalu kita cek isinya dengan menggunakan perintah pada listing 3.4.

```
1 imgatt2 = imgatt.pivot(index='imgid', columns='attid', values='present')
2
3 imgatt2.head()
4 imgatt2.shape
```

Listing 3.4: Pivot dataset

Sekarang kita akan meload jawabannya yang berisi apakah burung itu termasuk dalam spesies yang mana. Dua kolomnya adalah imgid dan label. Dan melakukan pivot yang mana imgid menjadi index yang artinya unik perintahnya ada di listing 3.5. Lalu kita cek kembali datanya.

```
1 imglabels = pd.read_csv("data/CUB_200_2011/image_class_labels.txt",
2                          sep=' ', header=None, names=['imgid', 'label'])
3
4 imglabels = imglabels.set_index('imgid')
5
6
7 imglabels.head()
8 imglabels.shape
```

Listing 3.5: Fungsi Sederhana

Karena isinya sama kita bisa melakukan join antara dua data. Sehingga kita akan mendapatkan data ciri dan data jawabannya atau labelnya sehingga bisa dikategorikan supervised learning. maka perintah untuk menggabungkan kedua data dan kemudian kita melakukan pemisahan antara data set untuk training dan test dengan perintah di listing 3.6.

```
1 df = imgatt2.join(imglabels)
2 df = df.sample(frac=1)
```

Listing 3.6: Fungsi Sederhana

Kemudian drop label yang didepan, dan gunakan label yang paling belakang yang baru di join dengan perintah listing 3.7.

```
1 df_att = df.iloc[:, :312]
2 df_label = df.iloc[:, 312:]
```

Listing 3.7: Fungsi Sederhana

Kita bisa mengecek isinya dengan perintah listing 3.8.

```
1 df_att.head()
2 df_label.head()
```

Listing 3.8: Fungsi Sederhana

Kita bagi menjadi dua bagian, 8000 row pertama sebagai data training sisanya sebagai data testing dengan perintah listing 3.9.

```
1 df_train_att = df_att[:8000]
2 df_train_label = df_label[:8000]
3 df_test_att = df_att[8000:]
4 df_test_label = df_label[8000:]
5
6 df_train_label = df_train_label['label']
7 df_test_label = df_test_label['label']
```

Listing 3.9: Fungsi Sederhana

Kita panggil kelas RandomForestClassifier. max features diartikan sebagai berapa banyak kolom pada setiap tree dengan perintah listing 3.10.

```
1 from sklearn.ensemble import RandomForestClassifier
2 clf = RandomForestClassifier(max_features=50, random_state=0,
    n_estimators=100)
```

Listing 3.10: Fungsi Sederhana

Kemudian lakukan fit untuk membangun random forest yang sudah ditentukan dengan maksimum fitur sebanyak 50 untuk perpohonnya dengan perintah listing 3.11.

```
1 clf.fit(df_train_att, df_train_label)
```

Listing 3.11: Fungsi Sederhana

Hasilnya bisa kita dapatkan dengan perintah predict dengan perintah listing 3.12.

```
1 print(clf.predict(df_train_att.head()))
```

Listing 3.12: Fungsi Sederhana

Untuk besaran akurasinya dengan perintah listing 3.13

```
1 clf.score(df_test_att, df_test_label)
```

Listing 3.13: Fungsi Sederhana

3.1.2 Confusion Matrix

Dari RF kita coba petakan ke dalam Confusion Matrix dan lihat hasilnya dengan perintah listing 3.14.

```
1 from sklearn.metrics import confusion_matrix
2 pred_labels = clf.predict(df_test_att)
3 cm = confusion_matrix(df_test_label, pred_labels)
4
5 cm
```

Listing 3.14: Fungsi Sederhana

Kemudian kita plot dengan perintah

```
1 import matplotlib.pyplot as plt
2 import itertools
3 def plot_confusion_matrix(cm, classes,
4                           normalize=False,
5                           title='Confusion matrix',
6                           cmap=plt.cm.Blues):
7     """
8     This function prints and plots the confusion matrix.
9     Normalization can be applied by setting 'normalize=True'.
10    """
11     if normalize:
12         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
13         print("Normalized confusion matrix")
14     else:
15         print('Confusion matrix, without normalization')
16
17     print(cm)
18
19     plt.imshow(cm, interpolation='nearest', cmap=cmap)
20     plt.title(title)
21     #plt.colorbar()
22     tick_marks = np.arange(len(classes))
23     plt.xticks(tick_marks, classes, rotation=90)
24     plt.yticks(tick_marks, classes)
25
26     fmt = '.2f' if normalize else 'd'
27     thresh = cm.max() / 2.
```

```

28     #for i, j in itertools.product(range(cm.shape[0]), range(cm.shape
    [1])):
29     #     plt.text(j, i, format(cm[i, j], fmt),
30     #             horizontalalignment="center",
31     #             color="white" if cm[i, j] > thresh else "black")
32
33     plt.tight_layout()
34     plt.ylabel('True label')
35     plt.xlabel('Predicted label')

```

Listing 3.15: Fungsi Sederhana

Agar plot sumbunya sesuai dengan nama datanya maka kita set dengan perintah

```

1 birds = pd.read_csv("data/CUB_200_2011/classes.txt",
2                     sep='\s+', header=None, usecols=[1], names=['
    birdname'])
3 birds = birds['birdname']
4 birds

```

Listing 3.16: Fungsi Sederhana

Lalu kita plot

```

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

```

Listing 3.17: Fungsi Sederhana

3.1.3 Mencoba dengan metode Decission Tree dan SVM

Kita coba menggunakan Decission tree

```

1 from sklearn import tree
2 clftree = tree.DecisionTreeClassifier()
3 clftree.fit(df_train_att, df_train_label)
4 clftree.score(df_test_att, df_test_label)

```

Listing 3.18: Fungsi Sederhana

Kita coba menggunakan SVM

```

1 from sklearn import svm
2 clfsvm = svm.SVC()
3 clfsvm.fit(df_train_att, df_train_label)
4 clfsvm.score(df_test_att, df_test_label)

```

Listing 3.19: Fungsi Sederhana

3.1.4 Pengecekan Cross Validation

Pengecekan Cross Validation untuk random forest

```
1 from sklearn.model_selection import cross_val_score
2 scores = cross_val_score(clf, df_train_att, df_train_label, cv=5)
3 # show average score and +/- two standard deviations away (covering 95%
  of scores)
4 print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```

Listing 3.20: Fungsi Sederhana

untuk decision tree

```
1 scorestree = cross_val_score(clftree, df_train_att, df_train_label, cv
  =5)
2 print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(), scorestree.std
  () * 2))
```

Listing 3.21: Fungsi Sederhana

untuk SVM

```
1 scoressvm = cross_val_score(clfsvm, df_train_att, df_train_label, cv=5)
2 print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(), scoressvm.std()
  * 2))
```

Listing 3.22: Fungsi Sederhana

3.1.5 Mengatahui komponen komponen informasi

Untuk mengetahui berapa banyak tree yang dibuat, berapa banyak atribut yang dipakai dan informasi lainnya menggunakan kode

```
1 max_features_opts = range(5, 50, 5)
2 n_estimators_opts = range(10, 200, 20)
3 rf_params = np.empty((len(max_features_opts)*len(n_estimators_opts),4),
  float)
4 i = 0
5 for max_features in max_features_opts:
6     for n_estimators in n_estimators_opts:
7         clf = RandomForestClassifier(max_features=max_features,
  n_estimators=n_estimators)
8         scores = cross_val_score(clf, df_train_att, df_train_label, cv
  =5)
9         rf_params[i,0] = max_features
10        rf_params[i,1] = n_estimators
11        rf_params[i,2] = scores.mean()
12        rf_params[i,3] = scores.std() * 2
13        i += 1
14        print("Max features: %d, num estimators: %d, accuracy: %0.2f
  (+/- %0.2f)" % (max_features, n_estimators, scores.
  mean(), scores.std() * 2))
```

Listing 3.23: Fungsi Sederhana

Dan kita bisa melakukan plot informasi ini dengan kode

```
1 import matplotlib.pyplot as plt
2 from mpl_toolkits.mplot3d import Axes3D
3 from matplotlib import cm
4 fig = plt.figure()
5 fig.clf()
6 ax = fig.gca(projection='3d')
7 x = rf_params[:,0]
8 y = rf_params[:,1]
9 z = rf_params[:,2]
10 ax.scatter(x, y, z)
11 ax.set_zlim(0.2, 0.5)
12 ax.set_xlabel('Max features')
13 ax.set_ylabel('Num estimators')
14 ax.set_zlabel('Avg accuracy')
15 plt.show()
```

Listing 3.24: Fungsi Sederhana

3.2 Soal Teori

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

1. Jelaskan apa itu random forest, sertakan gambar ilustrasi buatan sendiri.
2. Jelaskan cara membaca dataset dan artikan makna setiap file dan isi filenya.
3. Jelaskan apa itu evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar sendiri
4. Jelaskan bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix buatan sendiri.
5. Jelaskan bagaimana K-fold cross validation bekerja dengan gambar ilustrasi contoh buatan sendiri.
6. Jelaskan apa itu decision tree dengan gambar ilustrasi contoh buatan sendiri.
7. Jelaskan apa itu information gain dan entropi dengan gambar ilustrasi buatan sendiri.

3.3 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.

```
# load dataset (student mat pakenya)
import pandas as pd
d = pd.read_csv('student-mat.csv', sep=';')
len(d)
```
2.

```
# generate binary label (pass/fail) based on G1+G2+G3
# (test grades, each 0-20 pts); threshold for passing is sum>=30
d['pass'] = d.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3'])
>= 35 else 0, axis=1)
d = d.drop(['G1', 'G2', 'G3'], axis=1)
d.head()
```
3.

```
# use one-hot encoding on categorical columns
d = pd.get_dummies(d, columns=['sex', 'school', 'address',
'famsize',
'Pstatus', 'Mjob', 'Fjob',
'reason', 'guardian', 'schoolsup',
'famsup', 'paid', 'activities',
'nursery', 'higher', 'internet',
'romantic'])
d.head()
```
4.

```
# shuffle rows
d = d.sample(frac=1)
# split training and testing data
d_train = d[:500]
d_test = d[500:]
```

```

d_train_att = d_train.drop(['pass'], axis=1)
d_train_pass = d_train['pass']

d_test_att = d_test.drop(['pass'], axis=1)
d_test_pass = d_test['pass']

d_att = d.drop(['pass'], axis=1)
d_pass = d['pass']

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

5. # fit a decision tree
from sklearn import tree
t = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
t = t.fit(d_train_att, d_train_pass)

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

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

```

```

8. t.score(d_test_att, d_test_pass)

9. from sklearn.model_selection import cross_val_score
   scores = cross_val_score(t, d_att, d_pass, cv=5)
   # show average score and +/- two standard deviations away
   #(covering 95% of scores)
   print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))

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

11. depth_acc = np.empty((19,3), float)
    i = 0
    for max_depth in range(1, 20):
        t = tree.DecisionTreeClassifier(criterion="entropy",
        max_depth=max_depth)
        scores = cross_val_score(t, d_att, d_pass, cv=5)
        depth_acc[i,0] = max_depth
        depth_acc[i,1] = scores.mean()
        depth_acc[i,2] = scores.std() * 2
        i += 1

    depth_acc

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

```

3.4 Penanganan Error

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

1. skrinshot error
2. Tuliskan kode eror dan jenis errornya
3. Solusi pemecahan masalah error tersebut

Chapter 4

Experiment and Result

brief of experiment and result.

4.1 Experiment

Please tell how the experiment conducted from method.

4.2 Result

Please provide the result of experiment

Chapter 5

Conclusion

brief of conclusion

5.1 Conclusion of Problems

Tell about solving the problem

5.2 Conclusion of Method

Tell about solving using method

5.3 Conclusion of Experiment

Tell about solving in the experiment

5.4 Conclusion of Result

tell about result for purpose of this research.

Chapter 6

Discussion

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

Chapter 7

Discussion

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

Chapter 8

Discussion

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

Chapter 9

Discussion

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

Chapter 10

Discussion

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

Chapter 11

Discussion

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

Chapter 12

Discussion

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

Chapter 13

Discussion

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

Chapter 14

Discussion

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

Appendix A

Form Penilaian Jurnal

gambar A.1 dan A.2 merupakan contoh bagaimana reviewer menilai jurnal kita.

NO	UNSUR	KETERANGAN	MAKS	KETERANGAN
1	Keefektifan Judul Artikel	Maksimal 12 (dua belas) kata dalam Bahasa Indonesia atau 10 (sepuluh) kata dalam Bahasa Inggris	2	a. Tidak lugas dan tidak ringkas (0) b. Kurang lugas dan kurang ringkas (1) c. Ringkas dan lugas (2)
2	Pencantuman Nama Penulis dan Lembaga Penulis		1	a. Tidak lengkap dan tidak konsisten (0) b. Lengkap tetapi tidak konsisten (0,5) c. Lengkap dan konsisten (1)
3	Abstrak	Dalam Bahasa Indonesia dan Bahasa Inggris yang baik, jumlah 150-200 kata. Isi terdiri dari latar belakang, metode, hasil, dan kesimpulan. Isi tertuang dengan kalimat yang jelas.	2	a. Tidak dalam Bahasa Indonesia dan Bahasa Inggris (0) b. Abstrak kurang jelas dan ringkas, atau hanya dalam Bahasa Inggris, atau dalam Bahasa Indonesia saja (1) c. Abstrak yang jelas dan ringkas dalam Bahasa Indonesia dan Bahasa Inggris (2)
4	Kata Kunci	Maksimal 5 kata kunci terpenting dalam paper	1	a. Tidak ada (0) b. Ada tetapi kurang mencerminkan konsep penting dalam artikel (0,5) c. Ada dan mencerminkan konsep penting dalam artikel (1)
5	Sistematika Pembahasan	Terdiri dari pendahuluan, tinjauan pustaka, metode penelitian, hasil dan pembahasan, kesimpulan dan saran, daftar pustaka	1	a. Tidak lengkap (0) b. Lengkap tetapi tidak sesuai sistematika (0,5) c. Lengkap dan bersistem (1)
6	Pemanfaatan Instrumen Pendukung	Pemanfaatan Instrumen Pendukung seperti gambar dan tabel	1	a. Tidak dimanfaatkan (0) b. Kurang informatif atau komplementer (0,5) c. Informatif dan komplementer (1)
7	Cara Pengacuan dan Pengutipan		1	a. Tidak baku (0) b. Kurang baku (0,5) c. Baku (1)
8	Penyusunan Daftar Pustaka	Penyusunan Daftar Pustaka	1	a. Tidak baku (0) b. Kurang baku (0,5) c. Baku (1)
9	Peristilahan dan Kebahasaan		2	a. Buruk (0) b. Baik (1) c. Cukup (2)
10	Makna Sumbangan bagi Kemajuan		4	a. Tidak ada (0) b. Kurang (1) c. Sedang (2) d. Cukup (3) e. Tinggi (4)

Figure A.1: Form nilai bagian 1.

11	Dampak Ilmiah		7	a. Tidak ada (0) b. Kurang (1) c. Sedang (3) d. Cukup (5) e. Besar (7)
12	Nisbah Sumber Acuan Primer berbanding Sumber lainnya	Sumber acuan yang langsung merujuk pada bidang ilmiah tertentu, sesuai topik penelitian dan sudah teruji.	3	a. < 40% (1) b. 40-80% (2) c. > 80% (3)
13	Derajat Kemutakhiran Pustaka Acuan	Derajat Kemutakhiran Pustaka Acuan	3	a. < 40% (1) b. 40-80% (2) c. > 80% (3)
14	Analisis dan Sintesis	Analisis dan Sintesis	4	a. Sedang (2) b. Cukup (3) c. Baik (4)
15	Penyimpulan	Sangat jelas relevasinya dengan latar belakang dan pembahasan, dirumuskan dengan singkat	3	a. Kurang (1) b. Cukup (2) c. Baik (3)
16	Unsur Plagiat		0	a. Tidak mengandung plagiat (0) b. Terdapat bagian-bagian yang merupakan plagiat (-5) c. Keseluruhannya merupakan plagiat (-20)
TOTAL			36	
Catatan : Nilai minimal untuk diterima 25				

Figure A.2: form nilai bagian 2.

Appendix B

FAQ

M : Kalo Intership II atau TA harus buat aplikasi ? D : Ga harus buat aplikasi tapi harus ngoding

M : Pa saya bingung mau ngapain, saya juga bingung mau presentasi apa? D : Makanya baca de, buka jurnal topik ‘ganteng’ nah kamu baca dulu sehari 5 kali ya, 4 hari udah 20 tuh. Bingung itu tanda kurang wawasan alias kurang baca.

M : Pa saya sudah cari jurnal terindeks scopus tapi ga nemu. D : Kamu punya mata de? coba dicolok dulu. Kamu udah lakuin apa aja? tolong di list laporkan ke grup Tingkat Akhir. Tinggal buka google scholar klik dari tahun 2014, cek nama jurnalnya di scimagojr.com beres.

M : Pa saya belum dapat tempat intership, jadi ga tau mau presentasi apa? D : kamu kok ga nyambung, yang dipresentasikan itu yang kamu baca bukan yang akan kamu lakukan.

M : Pa ini jurnal harus yang terindex scopus ga bisa yang lain ? D : Index scopus menandakan artikel tersebut dalam standar semantik yang mudah dipahami dan dibaca serta bukan artikel asal jadi. Jika diluar scopus biasanya lebih sukar untuk dibaca dan dipahami karena tidak adanya proses review yang baik dan benar terhadap artikel.

M : Pa saya tidak mengerti D : Coba lihat standar alasan

M : Pa saya bingung D : Coba lihat standar alasan

M : Pa saya sibuk D : Mbahmu....

M : Pa saya ganteng D : Ndasmu....

M : Pa saya kece D : wes karepmu lah....

Biasanya anda memiliki alasan tertentu jika menghadapi kendala saat proses bimbingan, disini saya akan melakukan standar alasan agar persepsi yang diterima sama dan tidak salah kaprah. Penggunaan kata alasan tersebut antara lain :

1. Tidak Mengerti : anda boleh menggunakan alasan ini jika anda sudah melakukan tahapan membaca dan meresumekan 15 jurnal. Sudah mencoba dan mempraktekkan teorinya dengan mencari di youtube dan google minimal 6 jam sehari selama 3 hari berturut-turut.

2. Bingung : anda boleh mengatakan alasan bingung setelah maksimal dalam berusaha menyelesaikan tugas bimbingan dari dosen(sudah dilakukan semua). Anda belum bisa mengatakan alasan bingung jika anda masih belum menyelesaikan tugas bimbingan dan poin nomor 1 diatas. Setelah anda menyelesaikan tugas bimbingan secara maksimal dan tahap 1 poin diatas, tapi anda masih tetap bingung maka anda boleh memakai alasan ini.

Bibliography

- [1] Joshua Eckroth. *Python Artificial Intelligence Projects for Beginners: Get up and running with Artificial Intelligence using 8 smart and exciting AI applications*. Packt Publishing Ltd, 2018.
- [2] David L Poole and Alan K Mackworth. Python code for artificial intelligence: Foundations of computational agents, 2017.
- [3] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.