

Modul Praktikum Kecerdasan Buatan



Eni Lestari

1194012

Informatics Engineering

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

Politeknik Pos Indonesia

Bandung 2022

Contents

1	Sejarah dan Perkembangan Kecerdasan Buatan	1
1.1	Supervised Learning	2
1.2	Klasifikasi dan Regresi	2
1.3	Unsupervised Learning	3
1.4	Data set, Training set, Testing set	3
2	Membangun Model Prediksi	4
2.1	Binary Classification	4
2.2	Supervised Learning, Unsupervised Learning dan Clustering	4
2.3	Evaluasi dan Akurasi	6
2.4	Confusion Matrix	6
2.5	K-fold cross validation	7
2.6	Decision Tree	8
2.7	Information Gain dan Entropi	8
2.8	Praktikum	9
2.8.1	Scikit-Learn	9
3	Prediksi dengan Random Forest	18
3.1	Dataset	18
3.2	Cross Validation	19
3.3	Arti Score 44% Pada Random Forest, 27% Pada Decision Tree Dan 29% Dari SVM	19
3.4	Confusion Matrix	19
3.5	Voting Pada Random Forest	20
3.6	Praktikum	20
3.6.1	Aplikasi Sederhana Menggunakan Pandas	20
3.6.2	Aplikasi Sederhana Menggunakan Matplotlib	21
3.6.3	Aplikasi Sederhana Menggunakan Numpy	22

3.7	Program Klasifikasi Random Forest	22
3.8	Program Confusion Matrix	25
3.9	Program Klasifikasi SVM dan Decission Tree	25
3.10	Program Cross Validation	28
3.11	Program Pengamatan Komponen Informasi	28

List of Figures

2.1	Binary Classification	5
2.2	Supervised Learning : SVM Model Using Linear Kernel	5
2.3	Confusion Matrix	7
2.4	K-fold cross validation	8
2.5	Decision Tree	9
2.6	Load Dataset	12
2.7	Generate Binary Label	13
2.8	Use one-hot encoding on categorical columns	13
2.9	Shuffle Rows	14
2.10	Fit a decision tree	14
2.11	Visualize tree - graph	14
2.12	Score	15
2.13	Evaluasi Score - Cross Val Score	15
2.14	Max Depth	16
2.15	Depth In Range	17
2.16	Matplotlib	17
3.1	Aplikasi sederhana menggunakan pandas	21
3.2	Aplikasi sederhana menggunakan matplotlib	21
3.3	Aplikasi sederhana menggunakan numpy	22
3.4	Membaca dataset file txt	23
3.5	Mengetahui jumlah data	23
3.6	Pivot dataset	23
3.7	Membaca Dataset label	24
3.8	Menggabungkan field dari file yang terpisah	24
3.9	Memisahkan dan memilih label	24
3.10	Pembagian data training dan tes	25
3.11	Instansiasi kelas Random Forest	25
3.12	Plotting Confusion Matrix	26

3.13 Membaca file classes	26
3.14 Plot hasil perubahan label	27
3.15 Klasifikasi Decission Tree	27
3.16 Klasifikasi SVM	27
3.17 Hasil Cross Validation Random Forest	28
3.18 Hasil Cross Validation Decission Tree	28
3.19 Hasil Cross Validation SVM	28
3.20 Hasil plotting komponen	29
3.21 Hasil plotting komponen	30

Chapter 1

Sejarah dan Perkembangan Kecerdasan Buatan

Buku umum teori lengkap yang digunakan memiliki Semakin maju nya teknologi membuat banyak sekali perangkat pintar yang kemudian mengadopsi teknologi Kecerdasan buatan atau biasa disebut dengan Artificial Intelligence (AI). Dengan adanya kehadiran AI ini menimbulkan banyak sekali manfaat tentunya, ia dapat meringankan beban pekerjaan masuia serta membuatnya lebih efektif dan juga efisien. Kecerdasan buatan dapat disimpulkan sebagai suatu kecerdasan atau keahlian yang pada dasarnya merupakan buatan manusia. Yang mana kecerdasan otak manusia itu ialah alami dimiliki dan tumbuh sepanjang seseorang tersebut bernafas. Setelah itu, seseorang dengan kecerdasan alami ini lah kemudian mulai merangkai sebuah sistem atau perangkat, yang bertujuan supaya perangkat ini dapat memudahkan suatu pekerjaan sendiri maupun orang lain dalam tanda kutip manusia. Untuk alat yang berhasil diciptakan dan teknologi yang berhasil dilahirkan inilah yang kemudian disebut sebagai AI. Pada awalnya ai mulai dikenal publik itu karena ia memiliki kemampuan baik dalam menitu kegiatan manusia. Yang kemudian memunculkan banyak anggapan negetif, dimana manusia akan digeser oleh mesin. Akan tetapi semakin berkembangnya waktu, anggapan itupun berubah menjadi anggapan positif, karena dengan adanya AI ini dapat mempermudah dan mempuat pekerjaan menjadi mudah, cepat dan juga efektif efisien. Dengan adanya AI ini dinilai dapat membantu pekerjaan sehari-hari dan juga mudah untuk dikendalikan. Yang kemudian AI sudah dijadikan sebagai teman dan bukan lagi dianggap sebagai musuh atau pun ancaman bagi manusia.

Adapun manfaat dari teknologi AI ini antara lain sebagai berikut :

1. Membantu meminimalkan kesalahan

2. Solusi untuk hemat energi
3. Berperan dalam eksplorasi kekayaan alam
4. Hemat SDM
5. Bermanfaat di bidang kesehatan

1.1 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. Supervised Learning menyediakan algoritma pembelajaran dengan jumlah yang diketahui untuk mendukung penilaian dimasa depan. Chatbots, mobil self-driving, program pengenalan wajah, sistem pakar dan robot adalah beberapa sistem yang dapat menggunakan pembelajaran yang diawasi atau tidak. 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

1.2 Klasifikasi dan Regresi

Klasifikasi yaitu pendekatan pembelajaran yang diawasi dimana program komputer belajar dari input data yang diberikan kepadanya dan kemudian menggunakan pembelajaran ini untuk mengklarifikasikan pengamatan baru. Regresi adalah membahas mengenai masalah ketika variable output adalah nilai riil atau berkelanjutan contohnya seperti "gaji" atau "berat". banyak model yang berbeda dapat digunakan

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

1.3 Unsupervised Learning

Unsupervised Learning berbeda dengan Supervised Learning. Perbedaannya ialah unsupervised learning tidak memiliki data latih, sehingga dari data yang ada kita mengelompokkan data tersebut menjadi 2 ataupun 3 bagian dan seterusnya. 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

1.4 Data set, Training set, Testing set

Dataset adalah objek yang merepresentasikan data dan juga relasi yang ada di memory. Strukturnya mirip dengan data di database, namun bedanya dataset berisi koleksi dari data table dan data relation. Training Set adalah set digunakan oleh algoritma klasifikasi. Dapat dicontohkan dengan : decision tree, bayesian, neural network dll. Testing Set adalah set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Ini berfungsi sebagai meterai persetujuan, dan Anda tidak menggunakannya sampai akhir.

Chapter 2

Membangun Model Prediksi

2.1 Binary Classification

Binary Classification biasanya melibatkan satu kelas yang mana dalam keadaan normal dan kelas lainnya yang merupakan keadaan abnormal atau dapat berarti binary classification berupa kelas positif dan kelas negatif. Contohnya, pada email terdeteksi spam email, ada keadaan dimana email tersebut dapat berupa spam atau bukan. Misalnya bukan spam berarti keadaan normal dan spam berarti keadaan abnormal. Kelas dengan keadaan normal atau positif diberi label kelas 0 dan kelas dengan keadaan abnormal atau negatif diberi label kelas 1.

Adapun Algoritma yang digunakan untuk binary classification antarlain sebagai berikut :

1. Decision Trees
2. Support Vector Machine
3. Naive Bayes
4. Logistic Regression
5. K-Nearest Neighbors

2.2 Supervised Learning, Unsupervised Learning dan Clustering

1. Supervised Learning

Supervised Learning merupakan sebuah pemodelan dimana algoritmanya dapat

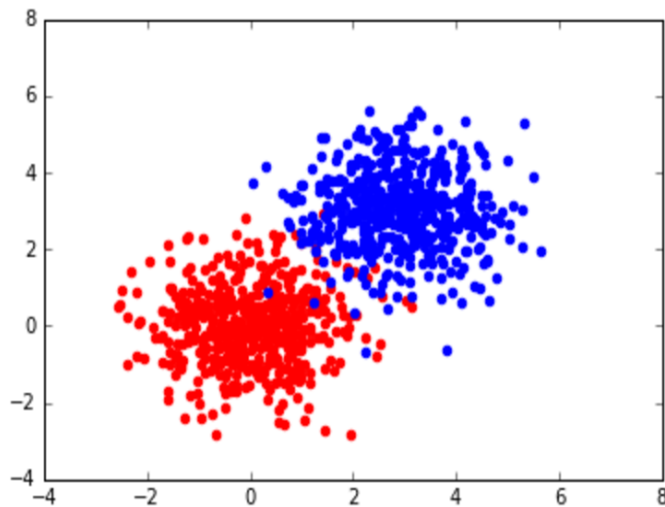


Figure 2.1: Binary Classification

membangkitkan suatu fungsi yang memetakan input ke output yang diinginkan. Pada Supervised Learning kita mengolah data yang memiliki label sehingga tujuan pengolahan tersebut adalah mengelompokkan data ke data yang sudah ada. Supervised Learning dalam kehidupan sehari-hari bisa ditemukan pada kasus prediksi harga saham, klasifikasi pelanggan, klasifikasi gambar dan lain-lain.

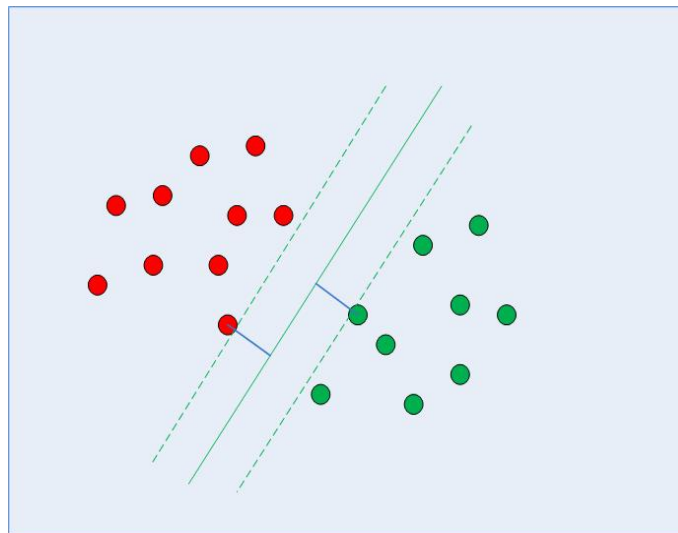


Figure 2.2: Supervised Learning : SVM Model Using Linear Kernel

2. Unsupervised Learning

Unsupervised Learning merupakan sebuah pemodelan dimana algoritmanya

memodelkan sekumpulan input secara otomatis tanpa adanya panduan output yang diinginkan. Unsupervised Learning kita mengolah data yang tidak memiliki label, sehingga tujuan kita dalam menggunakan Unsupervised Learning adalah mengelompokkan suatu data yang hampir sama dengan data tertentu. Adapun Algoritma yang digunakan untuk Unsupervised Learning antarlain sebagai berikut :

- (a) K-means
- (b) Hierarchical Clustering
- (c) DBSCAN
- (d) Fuzzy C-Means
- (e) Self-Organizing Map

3. Clustering

Clustering adalah metode pengelompokan objek sedemikian rupa sehingga objek dengan fitur serupa berkumpul, dan objek dengan fitur yang berbeda berpisah. Ini adalah teknik umum untuk analisis data statistik untuk pembelajaran mesin dan penggalian data. Analisis data eksplorasi dan generalisasi juga merupakan area yang menggunakan clustering.

2.3 Evaluasi dan Akurasi

Evaluasi merupakan kegiatan yang dilakukan untuk mengukur seberapa baik sebuah model dapat bekerja dengan menghitung akurasi. Akurasi merupakan ukuran atau persentase data yang diklasifikasikan dengan benar. Akurasi klasifikasi juga dapat berarti membagi jumlah prediksi benar terhadap total prediksi. Dalam model klasifikasi, dapat diprediksi nilai terbesar dan memberikan akurasi yang tinggi serta model yang dihasilkan dapat memprediksi nilai yang salah. Sehingga dalam hal ini dibutuhkan adanya metrik evaluasi yang dapat mengukur performa dari model klasifikasi yang sudah dibuat. Metrik yang digunakan adalah Precision, Recall dan Confusion Matrix.

2.4 Confusion Matrix

Confusion Matrix adalah pengukuran performa untuk masalah klasifikasi machine learning dimana keluaran dapat berupa dua kelas atau lebih. Confusion Matrix

adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual.

Berikut adalah contoh dari confusion matrix :

1. Accuracy
2. Precision (Positive Predictive Value)
3. Recall atau Sensitivity (True Positive Rate)

Cara membuat dan membaca confusion matrix

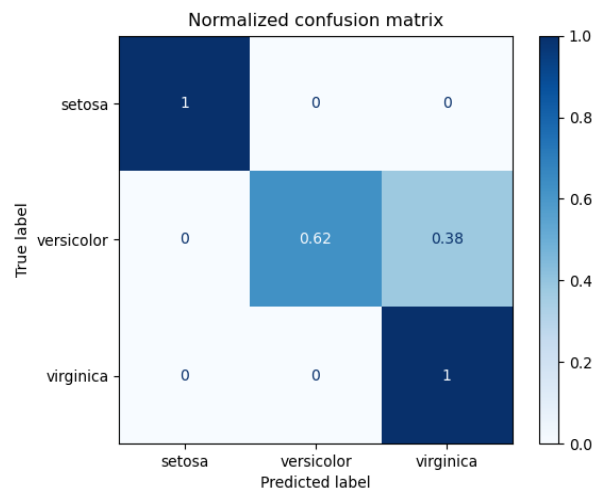


Figure 2.3: Confusion Matrix

2.5 K-fold cross validation

K-fold cross validation merupakan model evaluasi atau prosedur yang akan memisahkan antara data training dan data testing. K-fold cross validation dapat di definisikan sebagai pengujian cross validation yang digunakan untuk menilai kinerja dari sebuah metode algoritma dengan membagi sampel data secara acak dan mengelompokkan data tersebut sebanyak nilai K k-fold.

Cara kerja K-fold cross validation :

1. Tentukan instance, dibagi menjadi N bagian atau misalnya ada 10 data dan akan dilakukan K-fold cross validation pada data tersebut.
2. Data dibagi menjadi data testing untuk pengujian pada model dan data training untuk melatih model.

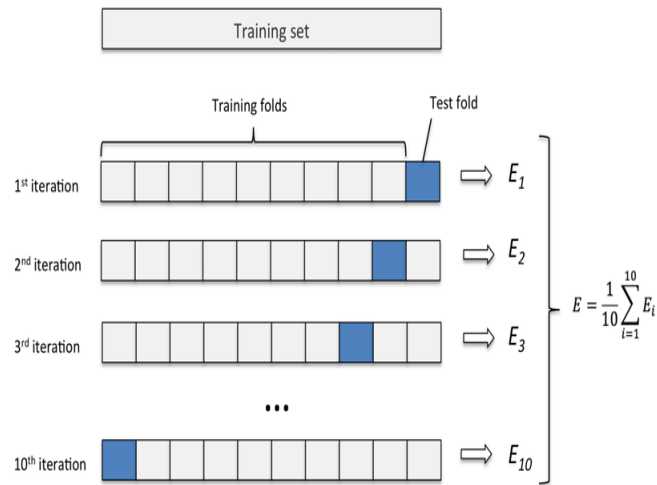


Figure 2.4: K-fold cross validation

3. Menentukan nilai K, misalnya dalam hal ini ditentukan nilai $K = 10$ dimana data tersebut nantinya akan ada 10 lipatan atau disebut dengan fold

2.6 Decision Tree

Decision Tree merupakan metode pembelajaran non parametrik yang digunakan untuk klasifikasi dan regresi. Tujuan dari decision tree adalah membuat model yang akan memprediksi nilai variable dengan mempelajari aturan keputusan sederhana yang disimpulkan dari sebuah fitur data. Decision tree merupakan struktur yang sama seperti diagram alur dimana simpul internal sebagai fitur atau atribut, cabang sebagai aturan dan keputusan, serta setiap simpul daun akan mewakili hasilnya.

2.7 Information Gain dan Entropi

1. Entropi

Entropy merupakan ukuran ketidakpastian. Semakin besar nilai informasi gain dari suatu atribut, maka semakin signifikan atribut tersebut untuk tugas prediksi. Sebuah objek yang akan di klasifikasikan ke dalam decision tree harus di uji nilai entropinya. Entropi merupakan ukuran dari informasi yang dapat mengetahui karakteristik dari dari impurity ,dan homogeneity dari sekumpulan data. Entropi juga merupakan jumlah bit yang diperkirakan akan dibutuhkan untuk mengekstrak suatu kelas dari data acak pada suatu ruang sampel. Dari ni-

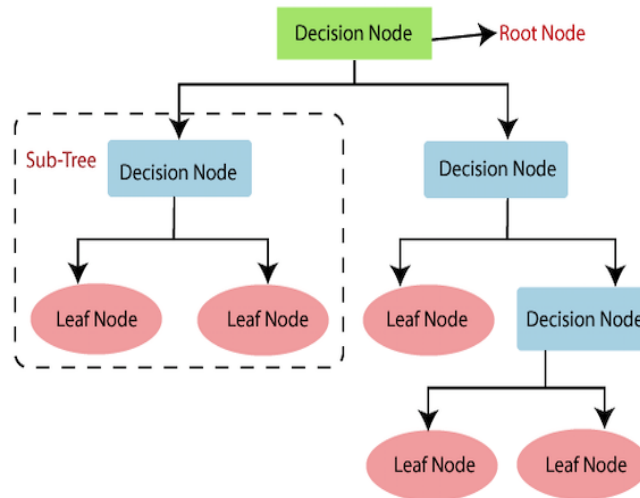


Figure 2.5: Decision Tree

lai entropi tersebut kemudian akan dihitung information gain masing-masing atribut.

2. Information Gain

Metode Information Gain adalah metode yang menggunakan teknik scoring untuk pembobotan sebuah fitur dengan menggunakan maksimal entropy. Fitur yang dipilih adalah fitur dengan nilai Information Gain yang lebih besar atau sama dengan nilai threshold tertentu. Kemudian setelah mendapatkan nilai entropi untuk suatu kumpulan data, maka akan diukur efektivitas nya atau disebut dengan information gain. Information gain digunakan untuk mengukur seberapa besar relevan atau pengaruh sebuah feature terhadap hasil pengukuran. Information Gain dikenal juga dengan sebutan Mutual Information dalam kasus untuk mengetahui dependency antara dua variable (x,y).

2.8 Praktikum

2.8.1 Scikit-Learn

```

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

```

2. #Generate binary label (pass/fail)
durian['pass'] = durian.apply(lambda row: 1 if (
    row['G1']+row['G2']+row['G3']) >= 35 else 0, axis=1)
durian = durian.drop(['G1', 'G2', 'G3'], axis=1)
durian.head()
#print(durian.head())

3. #Use one-hot encoding on categorical coloms

durian = pd.get_dummies(durian, columns=['sex', 'school', 'address', 'famsize',
    'reason', 'guardian', 'schoolsup', '
    'nursery', 'higher', 'internet', 'ro

durian.head()
#print(durian.head())

4. #shuffle rows
durian = durian.sample(frac=1)

durian_train = durian[:300]
durian_test = durian[300:]

durian_train_att = durian_train.drop(['pass'], axis=1)
durian_train_pass = durian_train['pass']

durian_test_att = durian_test.drop(['pass'], axis=1)
durian_test_pass = durian_test['pass']

durian_att = durian.drop(['pass'], axis=1)
durian_pass = durian['pass']

print("Passing: %d out of %d (%.2f%%)" % (np.sum(durian_pass), len(
durian_pass), 100*float(np.sum(durian_pass)) / len(durian_pass)))

```

```

5. #fit a decision tree
    timun = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
    timun = timun.fit(durian_train_att, durian_train_pass)

    #print(timun)

6. #Visualize tree
    delima_data = tree.export_graphviz(timun, out_file=None, label="all", impurity=
    feature_names=list(durian_train_att), class_names=["fail", "pass"],
    filled=True, rounded=True)

    graph = graphviz.Source(delima_data)
    #print(graph)

7. # save tree
    #Save tree
    tree.export_graphviz(timun, out_file='student-performance.dot', label="all",
                        feature_names=list(durian_train_att), class_names=['fail', 'pass'],
                        filled=True, rounded=True)

8. #t.score
    timun.score(durian_test_att, durian_test_pass)
    #print(timun.score(durian_test_att, durian_test_pass))

9. salak = cross_val_score(timun, durian_att, durian_pass, cv=5)
    # show average score and +/- two standard deviations away
    #(covering 95% of scores)
    print("Accuracy: %0.2f (+/- %0.2f)" % (salak.mean(), salak.std() * 2))

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

11. duku = np.empty((19, 3), float)
    ilwara = 0

```



```

for max_depth in range(1, 20):
    timun = tree.DecisionTreeClassifier(
        criterion="entropy", max_depth=max_depth)
    salak = cross_val_score(timun, durian_att, durian_pass, cv=5)
    duku[ilwara, 0] = max_depth
    duku[ilwara, 1] = salak.mean()
    duku[ilwara, 2] = salak.std() * 2
    ilwara += 1
print(duku)

```

12. `fig, ax = plt.subplots()`
`ax.errorbar(duku[:, 0], duku[:, 1], yerr=duku[:, 2])`
`plt.show()`

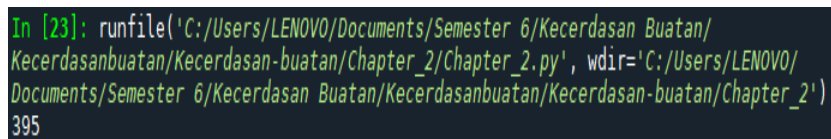
```

fig, ax = plt.subplots()
ax.errorbar(duku[:, 0], duku[:, 1], yerr=duku[:, 2])
plt.show()

```

13. Load Dataset student-mat.csv

Digunakan untuk import module pandas dan mendefinisikan suatu variable yang mana memiliki tugas untuk memanggil dataset yang diambil dari student-mat.csv.



```

In [23]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')
395

```

Figure 2.6: Load Dataset

14. Generate Binary Label

Selanjutnya pada bagian ini digunakan untuk mendeklarasikan pass/fail pada suatu data yang berdasarkan $G1+G2+G3$ dengan ketentuan nilai pass = 30 dan pada variable bandung dideklarasikan jika baris dengan $G1+G2+G3$ ditambahkan, dan hasilnya sama dengan 35 maka axis nya 1.

```
In [24]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')
  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]
```

Figure 2.7: Generate Binary Label

15. Use one-hot encoding on categorical columns

One-hot encoding merupakan proses dimana sebuah variable kategorikal dikonversikan menjadi bentuk yang dapat disediakan oleh algoritma Machine Learning untuk dapat melakukan pekerjaan yang lebih baik dalam memprediksi. Disini menggunakan fungsi panda `pd.get_dummies` untuk jenis kelamin, sekolah, alamat dan lainnya. Metode `head` ini digunakan untuk mengembalikan baris n atas 5 secara default dari frame atau seri datanya.

```
In [25]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')
  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]
```

Figure 2.8: Use one-hot encoding on categorical columns

16. Shuffle Rows

Pada `shuffle rows` ini ditujukan untuk mengembalikan sample secara tidak teratur dari objek. Terdapat `train` dan `test` yang digunakan untuk membagi `train`, `test` dan kemudian dibagi lagi `train` ke `validasi` dan `test`. kemudian di `import` sebuah modul `numpy` sebagai `np` yang digunakan untuk mengembalikan nilai `passing` dari pelajar dan dari keseluruhan dataset dengan menggunakan `print`.

17. Fit a decision tree

Import modul `tree` dari library `scikit-learn`. Dan selanjutnya nanti akan mendefin-

```
In [26]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')
Passing: 166 out of 395 (42.03%)
```

Figure 2.9: Shuffle Rows

isikan variable menggunakan decision classifier. Di dalam variable itulah terdapat criterion yang merupakan suatu fungsi untuk mengukur kualitas split. Agar decision tree classifier dapat di jalankan maka gunakan perintah fit

```
In [27]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')
DecisionTreeClassifier(criterion='entropy', max_depth=5)
```

Figure 2.10: Fit a decision tree

18. Visualize tree

Konsep dari decision tree adalah mengubah data menjadi aturan-aturan keputusan. Manfaat utama dari penggunaan decision tree adalah kemampuannya untuk mem-break down proses pengambilan keputusan yang kompleks menjadi lebih simple, sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan. Graphviz yaitu suatu perangkat lunak visualisasi grafik objek open source. Visualisasi grafik merupakan cara untuk mewakili informasi struktural sebagai diagram grafik dan jaringan abstrak. `treeexportgraphviz` merupakan sebuah fungsi yang akan menghasilkan representasi Graphviz dari decision tree, kemudian ditulis kedalam out file, sehingga akan ditampilkan sebuah diagram grafik bercabang.

```
In [28]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')
digraph Tree {
node [shape=box, style="filled, rounded", color="black", fontname=helvetica] ;
edge [fontname=helvetica] ;
0 [label="schoolsup <= 0.5\nsamples = 100.0%\nvalue = [0.58, 0.42]\nnclass = fail",
fillcolor="#f8dcc8"] ;
1 [label="failures <= 1.5\nsamples = 87.0%\nvalue = [0.536, 0.464]\nnclass = fail",
fillcolor="#fbf9e4"] ;
0 --> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
2 [label="Mjob_services <= 0.5\nsamples = 79.3%\nvalue = [0.5, 0.5]\nnclass = fail",
fillcolor="#ffffff"] ;
1 --> 2 ;
3 [label="Fjob_teacher <= 0.5\nsamples = 59.3%\nvalue = [0.562, 0.438]\nnclass = fail",
fillcolor="#f9e3d3"] ;
2 --> 3 ;
4 [label="famrel <= 1.5\nsamples = 55.3%\nvalue = [0.59, 0.41]\nnclass = fail",
fillcolor="#f7d8c2"] ;
3 --> 4 ;
5 [label="samples = 1.3%\nvalue = [0.0, 1.0]\nnclass = pass", fillcolor="#399de5"] ;
6 [label="samples = 54.0%\nvalue = [0.605, 0.395]\nnclass = fail", fillcolor="#f6d3ba"] ;
4 --> 5 ;
4 --> 6 ;
```

Figure 2.11: Visualize tree - graph

19. Save Tree

TREEEXPORTGRAPHVIZ merupakan sebuah fungsi yang akan menghasilkan representasi graphviz dari decision tree yang kemudian akan di tulis ke dalam sebuah outfile. Di dalam file tersebut akan disimpan classifier nya kemudian mengeksport file tersebut dengan namastudent performance kedalam folder tujuan kemudian jika salah maka akan mengembalikan nilai fail.

20. Score

Score biasa di kenal sebagai prediksi atau proses yang nantinya dapat menghasilkan nilai berdasarkan pada model pembelajaran mesin yang terlatih dan diberi beberapa data input baru. Nilai dibuat untuk mewakili prediksi nilai di masa depan atau juga dapat mewakili kategori serta hasil yang mungkin. Dalam hal ini variable solo akan memprediksi nilai bandung test att dan test pass.

```
In [29]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/  
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/  
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')  
0.6105263157894737
```

Figure 2.12: Score

21. Evaluasi Score - Cross Val Score

di dalam Evaluasi Score, di script ini akan mengevaluasi score dengan menggunakan validasi silang.

```
In [30]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/  
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/  
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')  
Accuracy: 0.59 (+/- 0.06)
```

Figure 2.13: Evaluasi Score - Cross Val Score

22. Max Depth

Script berikut dapat menunjukkan bahwa semakin banyak tree maka semakin banyak perpecahan yang dimiliki dan akan lebih banyak menangkap informasi dari data. Variable solo disini akan mendefinisikan tree kemudian variable jakarta akan mengevaluasi score nya dengan validasi silang. Kemudian akan di definisikan decision tree dengan kedalaman mulai dari 1 hingga 20 dan merencanakan pelatihan dan menguji skor auc.

```

In [32]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')
[[1.00000000e+00 5.79746835e-01 1.01265823e-02]
 [2.00000000e+00 6.15189873e-01 1.10467971e-01]
 [3.00000000e+00 5.34177215e-01 6.48417644e-02]
 [4.00000000e+00 5.62025316e-01 9.68664631e-02]
 [5.00000000e+00 5.56962025e-01 9.05749054e-02]
 [6.00000000e+00 5.44303797e-01 1.09769536e-01]
 [7.00000000e+00 5.72151899e-01 1.39033217e-01]
 [8.00000000e+00 5.77215190e-01 8.85714212e-02]
 [9.00000000e+00 5.72151899e-01 1.17004760e-01]
 [1.00000000e+01 5.54430380e-01 1.00503459e-01]
 [1.10000000e+01 5.74683544e-01 1.17223665e-01]
 [1.20000000e+01 5.77215190e-01 1.08122311e-01]
 [1.30000000e+01 5.77215190e-01 8.41177100e-02]
 [1.40000000e+01 6.07594937e-01 6.97926519e-02]
 [1.50000000e+01 5.62025316e-01 1.27691344e-01]
 [1.60000000e+01 5.94936709e-01 1.15460803e-01]
 [1.70000000e+01 6.05063291e-01 1.21307833e-01]
 [1.80000000e+01 5.84810127e-01 1.19175719e-01]
 [1.90000000e+01 5.97468354e-01 8.22687433e-02]]

```

Figure 2.14: Max Depth

23. Depth In Range

Depth acc membuat array kosong dengan mengembalikan array baru menggunakan bentuk dan tipe yang diberikan, tanpa menginisialisasi entri. Dengan 19 sebagai bentuk array kosong, 3 sebagai output data-type dan float urutan kolom utama (gaya Fortran) dalam memori. Variabel solo yang akan melakukan split score dan jakarta akan mengvalidasi score secara silang. Kemudian jakarta std yaitu menghitung standar deviasi dari data yang diberikan (elemen array) di sepanjang sumbu yang ditentukan (jika ada).

24. Matplotlib

Contoh gambar ini akan di import sebuah library dari matplotlib yaitu pyplot sebagai plt, fig dan ax yang menggunakan subplots untuk dapat membuat gambar serta satu set subplot. axerrorbar dalam script akan membuat error bar kemudian membuat sebuah grafik yang akan ditampilkan menggunakan perintah show.

```
In [31]: runfile('C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/
Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2/Chapter_2.py', wdir='C:/Users/LENOVO/
Documents/Semester 6/Kecerdasan Buatan/Kecerdasanbuatan/Kecerdasan-buatan/Chapter_2')
Max depth: 1, Accuracy: 0.57 (+/- 0.12)
Max depth: 2, Accuracy: 0.57 (+/- 0.12)
Max depth: 3, Accuracy: 0.57 (+/- 0.12)
Max depth: 4, Accuracy: 0.57 (+/- 0.12)
Max depth: 5, Accuracy: 0.57 (+/- 0.12)
Max depth: 6, Accuracy: 0.57 (+/- 0.12)
Max depth: 7, Accuracy: 0.57 (+/- 0.12)
Max depth: 8, Accuracy: 0.57 (+/- 0.12)
Max depth: 9, Accuracy: 0.57 (+/- 0.12)
Max depth: 10, Accuracy: 0.57 (+/- 0.12)
Max depth: 11, Accuracy: 0.57 (+/- 0.12)
Max depth: 12, Accuracy: 0.57 (+/- 0.12)
Max depth: 13, Accuracy: 0.57 (+/- 0.12)
Max depth: 14, Accuracy: 0.57 (+/- 0.12)
Max depth: 15, Accuracy: 0.57 (+/- 0.12)
Max depth: 16, Accuracy: 0.57 (+/- 0.12)
Max depth: 17, Accuracy: 0.57 (+/- 0.12)
Max depth: 18, Accuracy: 0.57 (+/- 0.12)
Max depth: 19, Accuracy: 0.57 (+/- 0.12)
```

Figure 2.15: Depth In Range

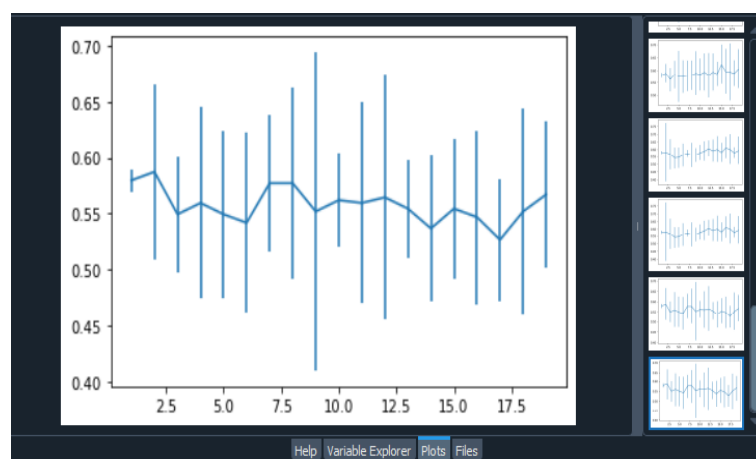


Figure 2.16: Matplotlib

Chapter 3

Prediksi dengan Random Forest

Random forest adalah suatu algoritma yang digunakan pada klasifikasi data dalam jumlah yang besar. Proses klasifikasi pada random forest berawal dari memecah data sampel yang ada kedalam decision tree secara acak. Setelah pohon terbentuk, maka akan dilakukan voting pada setiap kelas dari data sampel. Kemudian, mengkombinasikan vote dari setiap kelas kemudian diambil vote yang paling banyak. Random forest terdiri dari kumpulan decision tree. Random forest merupakan algoritma supervised learning yang digunakan untuk klasifikasi dan regresi. Random forest dapat menangani kumpulan data yang berisi variable kontinu seperti dalam kasus regresi dan variable kategoris dalam kasus klasifikasi. Random forest bekerja seperti ensemble dimana ensemble berarti menggabungkan beberapa model menjadi kumpulan model yang digunakan untuk membuat prediksi. Random forest akan memilih pengamatan secara acak kemudian membangun sebuah decision tree dan mengambil hasil rata-ratanya.

3.1 Dataset

Dataset adalah sekumpulan data yang disusun secara terstruktur. Biasanya, dataset dipresentasikan dalam bentuk tabel, alias baris dan kolom. Dataset merupakan sekumpulan data dimana data tersebut berasal dari informasi di masa lalu yang dikelola menjadi sebuah informasi untuk dapat melakukan data mining. Dataset berisi lebih dari satu variable yang digunakan untuk klasifikasi. Cara Membaca Dataset dan Arti Setiap File dan isi Field Masing-masing File :

1. Langkah awal yaitu dengan Menggunakan library Pandas pada python untuk membaca dataset dengan format text file.

2. Selanjutnya buat variable baru misalnya "dataset" yang berisi perintah untuk membaca file dataset.

3.2 Cross Validation

Cross Validation adalah metode yang paling biasa digunakan untuk evaluasi kinerja prediktif dari model. Data biasanya dibagi menjadi dua bagian dan berdasarkan pemisahan ini pada satu bagian, pelatihan dilakukan sementara prediktif diuji pada bagian lain. Cross validation merupakan metode untuk mengevaluasi dan membandingkan algoritma dengan membagi data menjadi dua yaitu data latih dan data uji. Cross validation (CV) adalah salah satu teknik yang digunakan untuk menguji keefektifan suatu model dan merupakan prosedur pengambilan sampel ulang yang digunakan untuk mengevaluasi suatu model jika memiliki data yang terbatas. Untuk dapat melakukan CV maka perlu menyisihkan sampel/sebagian data yang tidak digunakan untuk melatih model, kemudian menggunakan sampel ini untuk pengujian/validasi. Bentuk dasar dari cross-validation adalah k-fold cross-validation.

3.3 Arti Score 44% Pada Random Forest, 27% Pada Decission Tree Dan 29% Dari SVM

merupakan presentase keakurasian prediksi yang dilakukan pada saat testing menggunakan label pada dataset yang digunakan. Score akan mendefinisikan aturan evaluasi model, kemudian saat dijalankan akan muncul sebuah persentase yang menunjukkan keakurasian atau keberhasilan dari prediksi yang dilakukan. Jika menggunakan Random Forest maka hasilnya 40% dan jika menggunakan Decission Tree hasil prediksinya yaitu 27% dan pada SVM 29%.

3.4 Confusion Matrix

Confusion Matrix adalah pengukuran performa untuk masalah klasifikasi machine learning dimana keluaran dapat berupa dua kelas atau lebih. Confusion Matrix adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Confusion Matrix atau disebut juga dengan Error Matrix pada dasarnya akan memberikan informasi mengenai perbandingan hasil dari klasifikasi yang dilakukan oleh model atau sistem dengan hasil sebenarnya. Confusion Matrix berbentuk tabel matrix yang

akan menggambarkan kinerja dari model klasifikasi pada data uji yang nilai sebenarnya diketahui.

Berikut adalah contoh dari confusion matrix :

1. Precision (Positive Predictive Value)
2. Accuracy
3. Recall atau Sensitivity (True Positive Rate)

Langkah membuat dan membaca confusion matrix

1. Langkah awal yaitu Tentukan pokok permasalahan atau menggunakan dataset, misalnya data pasien covid
2. Langkah selanjutnya, Membagi dataset serta membuat model prediksi menggunakan Decision Tree. Dataset dibagi menjadi dua yaitu data latih dan data uji. Data latih digunakan untuk melatih model yang dibuat dan evaluasi akan dilakukan pada data uji.
3. lalu langkah terakhir, Evaluasi model menggunakan confusion matrix yaitu untuk mengetahui keakuratan model yang sudah dibuat menggunakan performance metrics seperti: accuracy, recall, dan precision.

3.5 Voting Pada Random Forest

Target prediksi dengan voting tertinggi digunakan sebagai prediksi akhir dari algoritma random forest. Voting merupakan suara untuk setiap target yang diprediksi pada saat melakukan Random Forest.

3.6 Praktikum

3.6.1 Aplikasi Sederhana Menggunakan Pandas

Penjelasan Code Aplikasi Sederhana pandas perbaris :

- Langkah awal yaitu di line 1 merupakan langkah mengimport library pandas kemudian di inisialisasi menjadi pd
- Untuk bagian Variable data di definisikan data data untuk kolom nama, kolom npm dan kolom angkatan

```
In [61]: runcell('buat aplikasi sederhana menggunakan pandas (ciri Khas per ibu kota)', 'C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/3/praktek_program.py')
Ibu Kota      Makanan      Tarian
0      Aceh      Mie Aceh      Tari Saman
1  Sumatera Utara  Bika ambon  Tari Tor Tor
2  Sumatera Barat  Rendang      Tari Piring
3      Jambi  Gulai patin  Tari Sekapur Sirih
4  Sumatera Selatan  Pempek, D.C.  Tari Tanggai

In [62]:
```

Figure 3.1: Aplikasi sederhana menggunakan pandas

- Pada bagian Variable frame itu akan berfungsi mengubah data pada variable data disejajarkan dengan baris dan kolom menggunakan pd dataframe
- Selanjutnya Perintah frame kemudian dijalankan untuk dapat menampilkan hasil dari dataframe

3.6.2 Aplikasi Sederhana Menggunakan Matplotlib

```
In [62]: runcell('Buat aplikasi sederhana menggunakan numpy', 'C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/3/praktek_program.py')
[1 2 3 4 5]
```

Figure 3.2: Aplikasi sederhana menggunakan matplotlib

Penjelasan Code Aplikasi Sederhana matplotlib perbaris :

- Baris Pertama Yaitu import library matplotlib kemudian di inisialisasi menjadi plt
- Variable prodi sebagai sumbu x untuk nama nama prodi
- Variable jumlah mhs sebagai sumbu y untuk jumlah atau banyaknya mahasiswa
- plt figure digunakan untuk mengatur size dan plt bar merupakan fungsi yang digunakan untuk memvisualisasikan bar
- plt title digunakan untuk memberikan judul dan plt ylabel untuk memberikan judul pada sumbu y
- plt yticks dan xticks digunakan untuk mengatur size tulisan pada sumbu y dan x

3.6.3 Aplikasi Sederhana Menggunakan Numpy

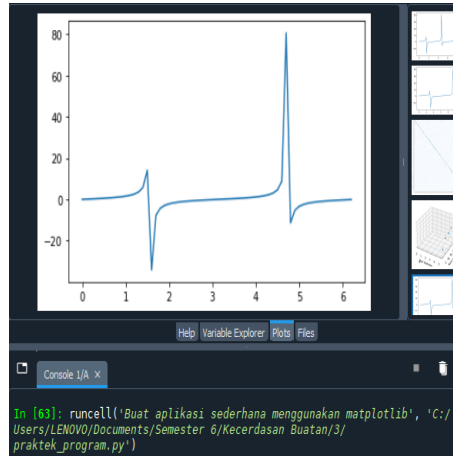


Figure 3.3: Aplikasi sederhana menggunakan numpy

Penjelasan Code Aplikasi Sederhana numpy perbaris :

- Baris Pertama Yaitu import library numpy kemudian di inisialisasi menjadi np
- Variable a digunakan sebagai fungsi array yang pertama
- Variable b digunakan sebagai fungsi array yang kedua
- Variable c digunakan sebagai fungsi array yang ketiga
- Perintah print digunakan untuk menampilkan hasil array
- Operasi pada array menggunakan perkalian pada a dan b dan penjumlahan pada b dan c

3.7 Program Klasifikasi Random Forest

Berikut merupakan output dari percobaan Random Forest yang telah dilakukan :

1. Kode berikut akan menampilkan output banyaknya jumlah baris dan kolom data frame imgatt
2. imgatt2 menggunakan function pivot untuk merubah kolom menjadi baris dan baris menjadi kolom dari data frame sebelumnya.

```

In [17]: runcell('melihat sebagian data awal dengan listing', 'C:/Users/
LENOVO/Documents/Semester 6/Kecerdasan Buatan/3/chapter3.py')
    imgid  attid  present
0      1      1      0
1      1      2      0
2      1      3      0
3      1      4      0
4      1      5      1

In [18]:

```

Figure 3.4: Membaca dataset file txt

```

In [18]: runcell('melihat jumlah data menggunakan listing', 'C:/Users/
LENOVO/Documents/Semester 6/Kecerdasan Buatan/3/chapter3.py')
(3677656, 3)

```

Figure 3.5: Mengetahui jumlah data

```

attid  1      2      3      4      5      6      7      ...  306  307  308  309  310
311  312
imgid
0      0      0      0      0      1      0      0      ...  0      0      1      0      0
1      0      0      0      0      0      0      0      ...  0      0      0      0      0
2      0      0      0      0      0      1      0      ...  0      0      1      0      0
3      0      0      0      0      0      1      0      ...  1      0      0      1      0
4      0      0      0      0      0      1      0      ...  0      0      0      0      0

[5 rows x 312 columns]
(11788, 312)

```

Figure 3.6: Pivot dataset

3. Pada kode berikut digunakan dataset label kemudian akan diberi label pada burung dan menentukan burung tersebut ke dalam spesies apa. di dalam data yang sebelumnya ada 312 kolom dimana 312 data tersebut memiliki kelompoknya masing-masing. Dalam hal ini akan dimunculkan dua kolom pada variable explorer yaitu `imgd` dan `label` yang terdiri dari 11788 baris dan 1 kolom

Figure 3.7: Membaca Dataset label

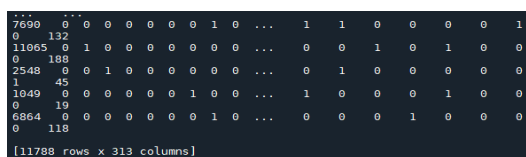


Figure 3.8: Menggabungkan field dari file yang terpisah



Figure 3.9: Memisahkan dan memilih label

```

8388 0 0 0 0 0 0 0 0 ... 0 0 0 1 0
0 0
3850 0 0 0 1 0 0 0 0 ... 0 0 0 0 0
0 1

[5 rows x 312 columns]
label
imgid
8807 150
758 14
9942 169
8388 143
3850 67

```

Figure 3.10: Pembagian data training dan tes

7. Melakukan klasifikasi, di dalam kelas `randomforestclassifier` setting parameter variable yaitu 50 dalam satu independen tree maksimal akan mengakomodir 50 atribut. Kemudian lakukan instansiasi dengan melakukan klasifikasi pada data train att beserta data label nya.

```

In [24]: runcell('instansiasi kelas Random Forest', 'C:/Users/LENOVO/
Documents/Semester 6/Kecerdasan Buatan/3/chapter3.py')
RandomForestClassifier(max_features=50, random_state=0)

```

Figure 3.11: Instansiasi kelas Random Forest

3.8 Program Confusion Matrix

Berikut merupakan output dari percobaan Confusion Matrix yang telah dilakukan :

1. Selanjutnya plotting confusion matrix menggunakan matplotlib.
2. Membaca File Classes
3. Plot hasil perubahan label

3.9 Program Klasifikasi SVM dan Decission Tree

Berikut merupakan output dari percobaan Klasifikasi SVM dan Decission Tree yang telah dilakukan :

1. Klasifikasi SVM dengan menggunakan dataset yang sama

```

195         196.House_Wren
196         197.Marsh_Wren
197         198.Rock_Wren
198         199.Winter_Wren
199         200.Common_Yellowthroat
Name: birdname, Length: 200, dtype: object
Normalized confusion matrix
[[0.13 0.04 0.35 ... 0.  0.  0. ]
 [0.  0.79 0.  ... 0.  0.  0. ]
 [0.  0.06 0.59 ... 0.  0.  0. ]
 ...
 [0.  0.  0.  ... 0.17 0.04 0. ]
 [0.  0.  0.  ... 0.  0.36 0. ]
 [0.  0.  0.  ... 0.  0.  0.76]]

```

Figure 3.12: Plotting Confusion Matrix

```

In [31]: runcell('membaca file classes txt', 'C:/Users/LENOVO/Documents/
Semester 6/Kecerdasan Buatan/3/chapter3.py')
0         001.Black_footed_Albatross
1         002.Laysan_Albatross
2         003.Sooty_Albatross
3         004.Groove_billed_Ani
4         005.Crested_Auklet
...
195        196.House_Wren
196        197.Marsh_Wren
197        198.Rock_Wren
198        199.Winter_Wren
199        200.Common_Yellowthroat
Name: birdname, Length: 200, dtype: object

```

Figure 3.13: Membaca file classes

```

In [31]: runcell('membaca file classes txt', 'C:/Users/LENOVO/Documents/
Semester 6/Kecerdasan Buatan/3/chapter3.py')
0      001.Black_footed_Albatross
1      002.Laysan_Albatross
2      003.Sooty_Albatross
3      004.Groove_billed_Ani
4      005.Crested_Auklet
...
195     196.House_Wren
196     197.Marsh_Wren
197     198.Rock_Wren
198     199.Winter_Wren
199     200.Common_Yellowthroat
Name: birdname, Length: 200, dtype: object

```

Figure 3.14: Plot hasil perubahan label

```

In [33]: runcell('Mencoba klasifikasi dengan decission tree', 'C:/Users/
LENOVO/Documents/Semester 6/Kecerdasan Buatan/3/chapter3.py')
0.27059134107708555

```

Figure 3.15: Klasifikasi Decission Tree

```

In [36]: clfsvm = svm.SVC()
...: clfsvm.fit(df_train_att, df_train_label)
...: clfsvm.score(df_test_att, df_test_label)
...:
...: print(clfsvm.score(df_test_att, df_test_label))

```

Figure 3.16: Klasifikasi SVM

3.10 Program Cross Validation

Berikut merupakan output dari percobaan Program Cross Validation yang telah dilakukan :

1. tugas minggu hari ini dan besok (maks 100). pada chapter ini
2. presentasi decision tree (maks 100). Mempraktekkan kode python dan menjelaskan cara kerjanya.
3. presentasi Random Forest (maks 100).Mempraktekkan kode python dan menjelaskan cara kerjanya.
4. Hasil Cross Validation untuk Random forest

```
In [39]: runcell('Hasil cross validation untuk random forest', 'C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/3/chapter3.py')
Accuracy: 0.44 (+/- 0.02)
```

Figure 3.17: Hasil Cross Validation Random Forest

5. Hasil Cross Validation untuk Decission Tree

```
In [40]: runcell('Hasil cross validation untuk decission tree', 'C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/3/chapter3.py')
Accuracy: 0.26 (+/- 0.03)
```

Figure 3.18: Hasil Cross Validation Decission Tree

6. Hasil Cross Validation untuk SVM

3.11 Program Pengamatan Komponen Informasi

Berikut merupakan output dari percobaan Program Pengamatan Komponen Informasi yang telah dilakukan :

```
In [43]: runcell('Hasil croos validation untuk SVM', 'C:/Users/LENOVO/Documents/Semester 6/Kecerdasan Buatan/3/chapter3.py')
Accuracy: 0.46 (+/- 0.02)
```

Figure 3.19: Hasil Cross Validation SVM

```
In [44]: runcell('melakukan pengamatan komponen informasi', 'C:/Users/
LENOVO/Documents/Semester 6/Kecerdasan Buatan/3/chapter3.py')
Max features: 5, num estimators: 10, accuracy: 0.26 (+/- 0.02)
Max features: 5, num estimators: 30, accuracy: 0.36 (+/- 0.01)
Max features: 5, num estimators: 50, accuracy: 0.39 (+/- 0.02)
Max features: 5, num estimators: 70, accuracy: 0.41 (+/- 0.03)
```

Figure 3.20: Hasil plotting komponen

1. Output berikut dapat mengetahui banyaknya tree yang dibuat, berapa atribut yang digunakan dan informasi lainnya.
2. Output berikut merupakan hasil plotting komponen informasi agar dapat dibaca

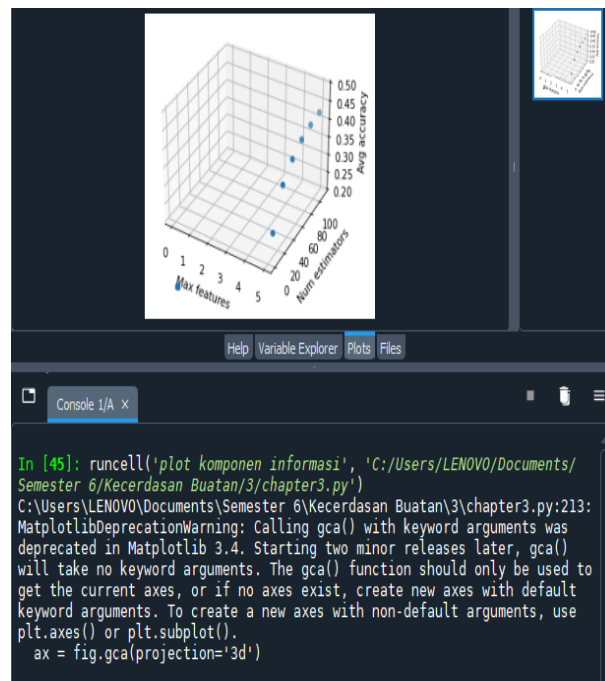


Figure 3.21: Hasil plotting komponen