Jane Doe and Max Power

Quarto CRC Book

To blah, blah, and blah.

Table of contents

References **43**

Preface

This is a Quarto book.

Software conventions

```
1 + 1
```

```
2
```

To learn more about Quarto books visit https://quarto.org/docs/books.

Acknowledgments

Blah, blah, blah...

1

TUGAS KLASIFIKASI DATA PROYEK SAINS DATA - B

Nama : Ananda Ramadana Ahmad Mulya
NIM : 210411100135
Kelas : B

2

— *BUSSINESS UNDERSTANDING* —

2.1 Tujuan

Melakukan klasifikasi untuk menentukan predikat kelulusan mahasiswa yang berasal dari berbagai jurusan berdasarkan ciri - ciri tertentu.

Untuk menentukan predikat kelulusan mahasiswa dapat ditentukan melalui ciri - ciri berikut :

2.2 Fitur Pada Dataset

1. `Target (label)`
2. `Marital Status (Status)`
3. `Application Mode (metode penerapan dari tugas akhir yang digunakan siswa)`
4. `Application Order (urutan siswa yang melamar)`
5. `Course (kursus yang diambil siswa)`
6. `Daytime/evening (waktu kursus)`
7. `Previous Qualification (kualifikasi yang telah diambil sebelumnya)`
8. `Nacionality (kewarganegaraan)`
9. `Mother's Qualification (kualifikasi dari ibu siswa)`
10. `Father's Qualification (kualifikasi dari ayah siswa)`
11. `Mother's Occupation (pekerjaan ibu)`

12. Father's Occupation (pekerjaan ayah)
13. Displaced (terlantar)
14. Educational Special Need (membutuhkan pendidikan tambahan)
15. Debtor (memiliki tanggungan)
16. Tuition fees up to date (biaya kuliah terbaru)
17. Gender
18. Scholarship Holder (memiliki sertifikat)
19. Age at enrollment (usia ketika terdaftar)
20. Internacional (termasuk murid internasional atau tidak)
21. Curricular units 1st sem (credited) (jumlah mata kuliah kurikuler yang dikreditkan mahasiswa pada semester pertama)
22. Curricular units 1st sem (enrolled) (jumlah mata kuliah kurikuler yang didaftarkan mahasiswa pada semester pertama)
23. Curricular units 1st sem (evaluations) (jumlah mata kuliah kurikuler yang dinilai mahasiswa pada semester pertama)
24. Curricular units 1st sem (approved) (jumlah mata kuliah kurikuler yang disetujui mahasiswa pada semester pertama)
25. Curricular units 1st sem (grade) (nilai rata - rata mata kuliah kurikuler mahasiswa pada semester pertama)
26. Curricular units 1st sem (without evaluations) (jumlah mata kuliah kurikuler yang tidak dinilai mahasiswa pada semester pertama)
27. Curricular units 2nd sem (credited) (jumlah mata kuliah kurikuler yang dikreditkan mahasiswa pada semester kedua)
28. Curricular units 2nd sem (enrolled) (jumlah mata kuliah kurikuler yang didaftarkan mahasiswa pada semester kedua)
29. Curricular units 2nd sem (evaluations) (jumlah mata kuliah kurikuler yang dinilai mahasiswa pada semester kedua)
30. Curricular units 2nd sem (approved) (jumlah mata kuliah kurikuler yang disetujui mahasiswa pada semester kedua)
31. Curricular units 2nd sem (grade) (nilai rata - rata mata kuliah kurikuler mahasiswa pada semester kedua)
32. Curricular units 2nd sem (without evaluations) (jumlah mata kuliah kurikuler yang tidak dinilai mahasiswa pada semester kedua)
33. Unemployment rate (persentase pengangguran)
34. Inflation rate (persentasi inflasi)

35. GDP

3

— DATA UNDERSTANDING —

3.1 Teknik Pengumpulan Data

Kumpulan data ini dibuat dari institusi pendidikan tinggi (diperoleh dari beberapa database terpisah) terkait mahasiswa yang terdaftar di berbagai gelar sarjana, seperti agronomi, desain, pendidikan, keperawatan, jurnalisme, manajemen, pelayanan sosial, dan teknologi. Dataset tersebut mencakup informasi yang diketahui pada saat pendaftaran mahasiswa (jalur akademik, demografi, dan faktor sosial ekonomi) dan prestasi akademik mahasiswa pada akhir semester pertama dan kedua. Masalah dirumuskan sebagai tugas klasifikasi tiga kategori, di mana terdapat ketidakseimbangan yang kuat pada salah satu kelas.

Data ini dibuat dalam sebuah proyek yang bertujuan untuk berkontribusi pada pengurangan angka putus sekolah dan kegagalan akademik di pendidikan tinggi, dengan menggunakan teknik pembelajaran mesin untuk mengidentifikasi siswa yang berisiko pada tahap awal jalur akademik mereka, sehingga strategi untuk mendukung mereka dapat diambil. Kumpulan data ini didukung oleh program SATDAP - Capacitação da Administração Pública berdasarkan hibah POCI-05-5762-FSE-000191, Portugal.

Jumlah Dataset sebanyak 4424 dengan rincian sebagai berikut: - Mahasiswa dengan predikat “Graduate”/“Lulus” (2) : 2209 - Mahasiswa dengan predikat “Dropout”/“Tidak Lulus” (0) : 1421 - Mahasiswa dengan predikat “Enrolled”/“Belum Lulus” (1) : 794

3.2 Mengidentifikasi Fitur

```
import pandas as pd

# membaca data dan ditampilkan
```

```
data = pd.read_csv('dataset.csv')
data.head(5)
```

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previou
0	1	8	5	2	1	1
1	1	6	1	11	1	1
2	1	1	5	5	1	1
3	1	8	2	15	1	1
4	2	12	1	3	0	1

```
# Rincian dataset (banyak data dan kolom)
print("Banyaknya data : ", data.shape[0])
print("Banyaknya kolom : ", data.shape[1])
```

```
Banyaknya data :  4424
Banyaknya kolom :  35
```

3.2.1 Deskripsi Fitur

1. `Target (label predikat mahasiswa setelah melalui identifikasi data setiap ciri - ciri)` > Graduate
   > Enrolled
   > Dropout
2. `Marital Status (Status pernikahan mahasiswa)`
   1 - single
   2 - married
   3 - widower
   4 - divorced
   5 - facto
   6 - legally
3. `Application Mode (metode penerapan dari tugas akhir pada semester 1 dan 2 yang digunakan siswa)`
   1 - 1st phase - general contingent
   2 - Ordinance No. 612/93
   3 - 1st phase - special contingent (Azores Island)
   4 - Holders of other higher courses
   5 - Ordinance No. 854-B/99
   6 - International student (bachelor)
   7 - 1st phase - special contingent (Madeira Island)
   8 - 2nd phase - general contingent
   9 - 3rd phase - general contingent

10 - Ordinance No. 533-A/99, item b2 (Different Plan)
11 - Ordinance No. 533-A/99, item b3 (Other Institution)
12 - Over 23 years old
13 - Transfer
14 - Change of course
15 - Technological specialization diploma holders
16 - Change of institution/course
17 - Short cycle diploma holders
18 - Change of institution/course (International)

4. `Application Order (urutan siswa yang melamar pada semester 1 dan 2)`

5. `Course (kursus yang pernah diambil siswa)`
   1 - Biofuel Production Technologies
   2 - Animation and Multimedia Design
   3 - Social Service (evening attendance)
   4 - Agronomy 9070 - Communication Design
   5 - Communication Design
   6 - Veterinary Nursing
   7 - Informatics Engineering
   8 - Equinculture
   9 - Management
   10 - Social Service
   11 - Tourism
   12 - Nursing
   13 - Oral Hygiene
   14 - Advertising and Marketing Management
   15 - Journalism and Communication
   16 - Basic Education
   17 - Management (evening attendance)

6. `Daytime/evening (waktu kursus yang pernah diambil siswa)` 1 - daytime
   0 - evening

7. `Previous Qualification (pendidikan terakhir dari siswa)`
   1 - Secondary education
   2 - Higher education - bachelor's degree
   3 - Higher education - degree
   4 - Higher education - master's
   5 - Higher education - doctorate
   6 - Frequency of higher education
   7 - 12th year of schooling - not completed
   8 - 11th year of schooling - not completed
   9 - Other - 11th year of schooling
   10 - 10th year of schooling

11 - 10th year of schooling - not completed
12 - Basic education 3rd cycle (9th/10th/11th year) or equiv.
13 - Basic education 2nd cycle (6th/7th/8th year) or equiv.
14 - Technological specialization course
15 - Higher education - degree (1st cycle)
16 - Professional higher technical course
17 - Higher education - master (2nd cycle)

8. `Nacionality (kewarganegaraan siswa)`
   1 - Portuguese
   2 - German
   3 - Spanish
   4 - Italian
   5 - Dutch
   6 - English
   7 - Lithuanian
   8 - Angolan
   9 - Cape Verdean
   10 - Guinean
   11 - Mozambican
   12 - Santomean
   13 - Turkish
   14 - Brazilian
   15 - Romanian
   16 - Moldova (Republic of)
   17 - Mexican
   18 - Ukrainian
   19 - Russian
   20 - Cuban
   21 - Colombian

9. `Mother's Qualification (pendidikan terakhir dari ibu siswa)`
   1 - Secondary Education - 12th Year of Schooling or Eq.
   2 - Higher Education - Bachelor's Degree
   3 - Higher Education - Degree
   4 - Higher Education - Master's
   5 - Higher Education - Doctorate
   6 - Frequency of Higher Education
   7 - 12th Year of Schooling - Not Completed
   8 - 11th Year of Schooling - Not Completed
   9 - 7th Year (Old)
   10 - Other - 11th Year of Schooling
   11 - 2nd year complementary high school course
   12 - 10th Year of Schooling
   13 - General commerce course
   14 - Basic Education 3rd Cycle (9th/10th/11th Year) or Equiv.

15 - Complementary High School Course
16 - Technical-professional course
17 - Complementary High School Course - not concluded
18 - 7th year of schooling
19 - 2nd cycle of the general high school course
20 - 9th Year of Schooling - Not Completed
21 - 8th year of schooling
22 - General Course of Administration and Commerce
23 - Supplementary Accounting and Administration
24 - Unknown
25 - Can't read or write
26 - Can read without having a 4th year of schooling
27 - Basic education 1st cycle (4th/5th year) or equiv.
28 - Basic Education 2nd Cycle (6th/7th/8th Year) or Equiv.
29 - Technological specialization course
30 - Higher education - degree (1st cycle)
31 - Specialized higher studies course
32 - Professional higher technical course
33 - Higher Education - Master (2nd cycle)
34 - Higher Education - Doctorate (3rd cycle)

10. `Father's Qualification (pendidikan terakhir dari ayah siswa)`
1 - Secondary Education - 12th Year of Schooling or Eq.
2 - Higher Education - Bachelor's Degree
3 - Higher Education - Degree
4 - Higher Education - Master's
5 - Higher Education - Doctorate
6 - Frequency of Higher Education
7 - 12th Year of Schooling - Not Completed
8 - 11th Year of Schooling - Not Completed
9 - 7th Year (Old)
10 - Other - 11th Year of Schooling
11 - 2nd year complementary high school course
12 - 10th Year of Schooling
13 - General commerce course
14 - Basic Education 3rd Cycle (9th/10th/11th Year) or Equiv.
15 - Complementary High School Course
16 - Technical-professional course
17 - Complementary High School Course - not concluded
18 - 7th year of schooling
19 - 2nd cycle of the general high school course
20 - 9th Year of Schooling - Not Completed
21 - 8th year of schooling
22 - General Course of Administration and Commerce
23 - Supplementary Accounting and Administration

24 - Unknown
25 - Can't read or write
26 - Can read without having a 4th year of schooling
27 - Basic education 1st cycle (4th/5th year) or equiv.
28 - Basic Education 2nd Cycle (6th/7th/8th Year) or Equiv.
29 - Technological specialization course
30 - Higher education - degree (1st cycle)
31 - Specialized higher studies course
32 - Professional higher technical course
33 - Higher Education - Master (2nd cycle)
34 - Higher Education - Doctorate (3rd cycle)

11. `Mother's Occupation (pekerjaan ibu siswa)`
0 - Student
1 - Representatives of the Legislative Power and Executive Bodies, Directors, Directors and Executive Managers
2 - Specialists in Intellectual and Scientific Activities
3 - Intermediate Level Technicians and Professions
4 - Administrative staff
5 - Personal Services, Security and Safety Workers and Sellers
6 - Farmers and Skilled Workers in Agriculture, Fisheries and Forestry
7 - Skilled Workers in Industry, Construction and Craftsmen
8 - Installation and Machine Operators and Assembly Workers
9 - Unskilled Workers
10 - Armed Forces Professions
11 - Other Situation
12 - (blank)
13 - Health professionals
14 - teachers
15 - Specialists in information and communication technologies (ICT)
16 - Intermediate level science and engineering technicians and professions
17 - Technicians and professionals, of intermediate level of health
18 - Intermediate level technicians from legal, social, sports, cultural and similar services
19 - Office workers, secretaries in general and data processing operators
20 - Data, accounting, statistical, financial services and registry-related operators
21 - Other administrative support staff
22 - personal service workers
23 - sellers
24 - Personal care workers and the like

25 - Skilled construction workers and the like, except electricians
26 - Skilled workers in printing, precision instrument manufacturing, jewelers, artisans and the like
27 - Workers in food processing, woodworking, clothing and other industries and crafts
28 - cleaning workers
29 - Unskilled workers in agriculture, animal production, fisheries and forestry
30 - Unskilled workers in extractive industry, construction, manufacturing and transport
31 - Meal preparation assistants

12. `Father's Occupation (pekerjaan ayah siswa)`
0 - Student
1 - Representatives of the Legislative Power and Executive Bodies, Directors, Directors and Executive Managers
2 - Specialists in Intellectual and Scientific Activities
3 - Intermediate Level Technicians and Professions
4 - Administrative staff
5 - Personal Services, Security and Safety Workers and Sellers
6 - Farmers and Skilled Workers in Agriculture, Fisheries and Forestry
7 - Skilled Workers in Industry, Construction and Craftsmen
8 - Installation and Machine Operators and Assembly Workers
9 - Unskilled Workers
10 - Armed Forces Professions
11 - Other Situation
12 - (blank)
13 - Health professionals
14 - teachers
15 - Specialists in information and communication technologies (ICT)
16 - Intermediate level science and engineering technicians and professions
17 - Technicians and professionals, of intermediate level of health
18 - Intermediate level technicians from legal, social, sports, cultural and similar services
19 - Office workers, secretaries in general and data processing operators
20 - Data, accounting, statistical, financial services and registry-related operators
21 - Other administrative support staff
22 - personal service workers
23 - sellers
24 - Personal care workers and the like

25 - Skilled construction workers and the like, except electricians
26 - Skilled workers in printing, precision instrument manufacturing, jewelers, artisans and the like
27 - Workers in food processing, woodworking, clothing and other industries and crafts
28 - cleaning workers
29 - Unskilled workers in agriculture, animal production, fisheries and forestry
30 - Unskilled workers in extractive industry, construction, manufacturing and transport
31 - Meal preparation assistants

13. `Displaced (keadaan keluarga siswa apakah termasuk kelaurga cukup atau tidak)`
    1 - yes
    0 - no

14. `Educational Special Need (apakah siswa membutuhkan pendidikan tambahan)`
    1 - yes
    0 - no

15. `Debtor (apakah siswa memiliki tanggungan berupa hutang)`
    1 - yes
    0 - no

16. `Tuition fees up to date (apakah siswa telah membayar biaya kuliah terakhir)`
    1 - yes
    0 - no

17. `Gender`
    1 - male
    0 - female

18. `Scholarship Holder (sertifikat yang dimiliki siswa)`
    1 - yes
    0 - no

19. `Age at enrollment (usia siswa ketika terdaftar di kampus)`

20. `Internacional (termasuk murid internasional atau tidak)`
    1 - yes
    0 - no

21. `Curricular units 1st sem (credited) (jumlah mata kuliah kurikuler yang dikreditkan mahasiswa pada semester pertama)`

22. `Curricular units 1st sem (enrolled) (jumlah mata kuliah kurikuler yang didaftarkan mahasiswa pada semester pertama)`

23. Curricular units 1st sem (evaluations) (jumlah mata kuliah kurikuler yang dinilai mahasiswa pada semester pertama)
24. Curricular units 1st sem (approved) (jumlah mata kuliah kurikuler yang disetujui mahasiswa pada semester pertama)
25. Curricular units 1st sem (grade) (nilai rata - rata mata kuliah kurikuler mahasiswa pada semester pertama)
26. Curricular units 1st sem (without evaluations) (jumlah mata kuliah kurikuler yang tidak dinilai mahasiswa pada semester pertama)
27. Curricular units 2nd sem (credited) (jumlah mata kuliah kurikuler yang dikreditkan mahasiswa pada semester kedua)
28. Curricular units 2nd sem (enrolled) (jumlah mata kuliah kurikuler yang didaftarkan mahasiswa pada semester kedua)
29. Curricular units 2nd sem (evaluations) (jumlah mata kuliah kurikuler yang dinilai mahasiswa pada semester kedua)
30. Curricular units 2nd sem (approved) (jumlah mata kuliah kurikuler yang disetujui mahasiswa pada semester kedua)
31. Curricular units 2nd sem (grade) (nilai rata - rata mata kuliah kurikuler mahasiswa pada semester kedua)
32. Curricular units 2nd sem (without evaluations) (jumlah mata kuliah kurikuler yang tidak dinilai mahasiswa pada semester kedua)
33. Unemployment rate (persentase pengangguran)
34. Inflation rate (persentasi inflasi)
35. GDP

```
data.columns
```

```
Index(['Marital status', 'Application mode', 'Application order', 'Course',
    'Daytime/evening attendance', 'Previous qualification', 'Nacionality',
       'Mother's qualification', 'Father's qualification',
       'Mother's occupation', 'Father's occupation', 'Displaced',
      'Educational special needs', 'Debtor', 'Tuition fees up to date',
    'Gender', 'Scholarship holder', 'Age at enrollment', 'International',
       'Curricular units 1st sem (credited)',
       'Curricular units 1st sem (enrolled)',
       'Curricular units 1st sem (evaluations)',
       'Curricular units 1st sem (approved)',
```

```
    'Curricular units 1st sem (grade)',
    'Curricular units 1st sem (without evaluations)',
    'Curricular units 2nd sem (credited)',
    'Curricular units 2nd sem (enrolled)',
    'Curricular units 2nd sem (evaluations)',
    'Curricular units 2nd sem (approved)',
    'Curricular units 2nd sem (grade)',
'Curricular units 2nd sem (without evaluations)', 'Unemployment rate',
    'Inflation rate', 'GDP', 'Target'],
   dtype='object')
```

3.2.2 Tipe data

Berikut Macam - Macam Data yang ada pada data ini.

1. Tipe nominal
   - memiliki value 1 yang melambangkan ya dan 0 yang melambangkan tidak. > Pada data ini mencakup fitur : *'Displaced', 'Educational Special Need', 'Debtor', 'Tuition feed up to date', 'Scholarship Holder', 'Internacional'.*
   - memiliki value perempuan dan laki laki. > yakni pada fitur *'Gender'.*
   - memiliki value siang dan sore hari. > yakni pada fitur *'Daytime/evening'.*
   - mencakup tipe data numeric. > yakni pada fitur *'Application Order','Curricular units 1st sem (credited)','Curricular units 1st sem (enrolled)','Curricular units 1st sem (evaluations)','Curricular units 1st sem (approved)','Curricular units 1st sem (grade)',dan 'Curricular units 1st sem (without evaluations)'.*
2. Tipe rentang > yakni pada fitur *'Age at enrollment'.*
3. Tipe ordinal > yakni pada fitur *'Inflation rate','Unemployment rate','Previous Qualification', dan 'GDP'.*

3.3 Mengidentifikasi Data

3.3.1 Missing Value

Missing Values sesuai dengan namanya yaitu keberadaan nilai yang kosong atau hilang pada data. Pada proses analsisis data hilangnya banyak data dapat menyebabkan akurasi yang dihasilkan semakin menurun

```
# Menghitung apakah ada nilai yang hilang dalam setiap kolom
missing_values = data.isna().any()

# Menampilkan hasil
print("Hasil Deteksi Missing Values")
print(missing_values)
print("Total Missing Values :", missing_values.sum())
```

```
Hasil Deteksi Missing Values
Marital status                                  False
Application mode                                False
Application order                               False
Course                                          False
Daytime/evening attendance                      False
Previous qualification                          False
Nacionality                                     False
Mother's qualification                          False
Father's qualification                          False
Mother's occupation                             False
Father's occupation                             False
Displaced                                       False
Educational special needs                       False
Debtor                                          False
Tuition fees up to date                         False
Gender                                          False
Scholarship holder                              False
Age at enrollment                               False
International                                   False
Curricular units 1st sem (credited)             False
Curricular units 1st sem (enrolled)             False
Curricular units 1st sem (evaluations)          False
Curricular units 1st sem (approved)             False
Curricular units 1st sem (grade)                False
Curricular units 1st sem (without evaluations)  False
Curricular units 2nd sem (credited)             False
Curricular units 2nd sem (enrolled)             False
Curricular units 2nd sem (evaluations)          False
Curricular units 2nd sem (approved)             False
Curricular units 2nd sem (grade)                False
Curricular units 2nd sem (without evaluations)  False
Unemployment rate                               False
Inflation rate                                  False
GDP                                             False
Target                                          False
```

```
dtype: bool
Total Missing Values : 0
```

3.3.2 Duplikat Data

Mengecek data duplikat adalah proses untuk menemukan dan mengidentifikasi apakah ada entri yang sama atau serupa dalam suatu set data. Fungsi ini umumnya digunakan dalam pengolahan data dan analisis untuk memastikan keakuratan dan konsistensi data.

```
# menghitung jumlah data redundan
jumlah_duplikat = data.duplicated().sum()

# Menampilkan jumlah data yang duplikat
print("Jumlah data yang duplikat:", jumlah_duplikat)
```

```
Jumlah data yang duplikat: 0
```

3.3.3 Visualisasi Data

Visualisasi data menggunakan grafik memiliki beberapa fungsi penting dalam analisis data. Berikut adalah beberapa manfaat utama dari visualisasi data menggunakan grafik:

1. **Meringkas Informasi:** Grafik membantu dalam merangkum informasi yang kompleks menjadi bentuk yang lebih mudah dipahami. Dengan melihat grafik, pengguna dapat dengan cepat mendapatkan gambaran umum tentang pola atau tren dalam data.
2. **Mengidentifikasi Pola dan Tren:** Grafik memungkinkan pengguna untuk mengidentifikasi pola atau tren dalam data dengan lebih mudah daripada hanya melihat angka-angka mentah. Ini membantu dalam pemahaman yang lebih baik tentang hubungan antar variabel dan perubahan sepanjang waktu.
3. **Membandingkan Data:** Grafik memudahkan perbandingan antara berbagai set data atau variabel. Dengan membandingkan grafik, pengguna dapat melihat perbedaan, kesamaan, atau tren di antara berbagai kategori atau kelompok.
4. **Mengungkap Outlier:** Grafik membantu dalam mengidentifikasi outlier atau data yang tidak biasa. Outlier dapat memberikan wawasan tambahan tentang data atau mencerminkan anomali yang perlu diselidiki lebih lanjut.
5. **Membantu Pengambilan Keputusan:** Visualisasi data dapat

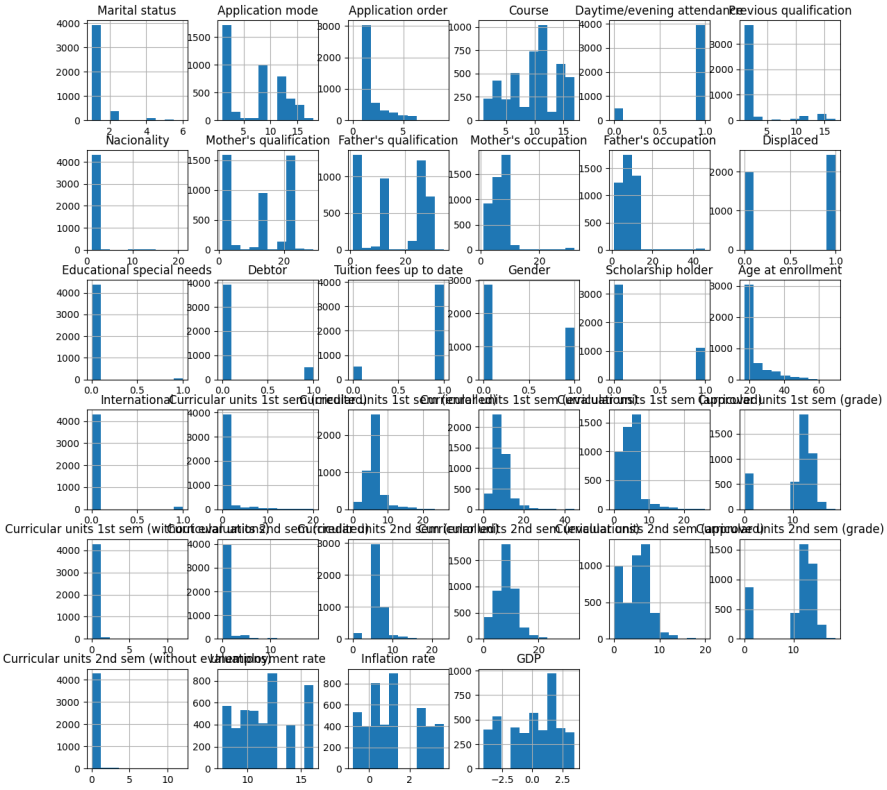
membantu pengambilan keputusan dengan menyediakan pemahaman yang lebih baik tentang situasi atau masalah. Grafik yang jelas dan informatif dapat mendukung proses pengambilan keputusan yang lebih baik.

6. **Meningkatkan Komunikasi:** Grafik memberikan cara yang lebih efektif untuk berkomunikasi hasil analisis data kepada orang lain yang mungkin tidak memiliki pengetahuan teknis yang sama. Visualisasi dapat membantu mempermudah pemahaman dan memudahkan diskusi.
7. **Memahami Distribusi Data:** Grafik memungkinkan pemahaman yang lebih baik tentang distribusi data, seperti apakah data terdistribusi normal, atau apakah ada skewness atau kurtosis yang signifikan.
8. **Meningkatkan Daya Ingat:** Manusia cenderung lebih baik mengingat informasi visual daripada informasi teks atau numerik. Oleh karena itu, penggunaan grafik dapat meningkatkan daya ingat terhadap informasi yang disajikan.
9. **Menyoroti Perbedaan dan Kesamaan:** Grafik dapat dengan mudah menyoroti perbedaan dan kesamaan antara berbagai kategori atau variabel, membantu dalam menarik kesimpulan yang lebih cepat dan efektif.
10. **Mendorong Eksplorasi Data:** Visualisasi data merangsang eksplorasi lebih lanjut terhadap data. Pengguna dapat dengan mudah mengidentifikasi area yang menarik perhatian dan mengeksplorasi lebih dalam untuk pemahaman yang lebih mendalam.

Dengan menggunakan grafik, informasi yang terkandung dalam data dapat diungkapkan secara lebih jelas dan dapat dimengerti oleh berbagai pemangku kepentingan, baik yang memiliki latar belakang teknis maupun non-teknis.

```
import matplotlib.pyplot as plt

# menampilkan distribusi data di dalam bentuk grafik
data.hist(figsize=(14, 14))
plt.show()
```



3.3.4 Proporsi Data

Untuk mencapai hasil maksimal, perlu dilakukan identifikasi proporsi jumlah data dari masing-masing kelas. Dengan begitu ketidakseimbangan data disetiap kelas pada data red wine ini dapat ditangani dengan menyeimbangkan jumlah data disetiap kelasnya. Dengan ketentuan

Graduate : 2
Dropout : 0
Enrolled : 1

```
data['Target'].value_counts()
```

```
Target
Graduate    2209
Dropout     1421
Enrolled     794
Name: count, dtype: int64
```

3.4 Mengidentifikasi Outlier

Outlier Pada Data adalah nilai yang berbeda dari yang lain dimana perbedaannya sangat jauh dengan sekumpulan data yang lain dalam satu kolom.Keberadaan Outlier sendiri dinilai dapat mengganggu analisis statistik dan kesimpulan yang diambil dari data karena mereka bisa menyebabkan pergeseran rata-rata atau mengganggu distribusi data secara keseluruhan. Maka dari itu, pada data red wine ini perlu dilakukan identifikasi outlier pada data. Untuk menentukan outluer pada data dapat dengan menggunakan metode Local Outlier Factor.

3.4.0.1 LOCAL OUTLIER FACTOR

Adalah metode yang digunakan untuk mendeteksi outlier dalam data dengan memperhatikan konteks lokal dari setiap data poin. LOF menghitung seberapa "aneh" atau tidak biasa suatu poin data jika dibandingkan dengan tetangga-tetangganya. Poin yang memiliki LOF tinggi dibandingkan dengan tetangganya dapat dianggap sebagai outlier.

Adapun tahap-tahp untuk mengidentifikasi outlier pada data dengan menggunakan Local Outlier Factor :

1. Hitung Jarak Antar Data dimana jarak yang dihitung adalah jarak titik yang akan dievaluasi dengan semua titik didalam satu baris. Perhitungan Jarak dilakukan menggunakan perhitungan jarak euclidean.
$$\text{distance}(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$
dimana : p = titik yang akan dievaluasi q = titik selain titik p

2. Hitung Kepadatan Lokal Setelah jarak diketahui, maka selanjutnya kepadatan lokal dari titik data tersebut perlu dihitung. Kepadatan lokal dapat dihitung dengan membandingkan jumlah titik-titik tetangga dalam jarak tertentu (radius) terhadap titik data yang sedang dievaluasi.
$$\text{Local Density}(p) = \frac{\text{jumlah tetangga dalam radius}}{\text{jumlah total data}}$$

3. Hitung Local Reachability Density(LRD) Hitung kepadatan jarak (reachability distance) dari titik data (p) terhadap tetangganya (q). Local Reachability Density dari titik p terhadap tetangga q dihitung sebagai rata-rata dari jarak antara q dan p terhadap tetangga q:
$$\text{reachdist}(p, q) = \max(\text{distance}(p, q), \text{radius})$$

$$\text{Local Reachability Density}(p) = \frac{1}{\text{jumlah tetangga}} \sum_{q \in N_{\text{radius}}(p)} \frac{\text{reachdist}(p,q)}{\text{density}(q)}$$

dimana: - N radius(p) adalah himpunan tetangga dalam radius tertentu radius dari titik p. - density(q) adalah kepadatan lokal dari tetangga q.

4. Hitung Nilai LOF LOF dari suatu titik data (p) dihitung sebagai rasio dari rata-rata Local Reachability Density dari tetangganya terhadap kepadatan lokalnya sendiri:
$$\text{LOF}(p) = \frac{1}{\text{jumlah tetangga}} \sum_{q \in N_{\text{radius}}(p)} \frac{\text{Local Reachability Density}(q)}{\text{Local Reachability Density}(p)}$$

dimana : LOF yang tinggi menunjukkan bahwa titik tersebut memiliki kepadatan lokal yang lebih rendah dibandingkan dengan tetangganya, sehingga cenderung menjadi outlier.

Contoh Kasus, untuk mencari outlier pada data misalkan terdapat tabel seperti dibawah ini:

X	Y
1	4
2	3
3	8
7	2
5	9

- Langkah 1 : Hitung Jarak Antar Data lalu nilai radius yang diambil adalah 5

X	Y	Jarak
1	4	1,41 ; 4,47 ; 4,47
2	3	1,41 ; 3,16
3	8	4,47
5	2	4,47 ; 3,16 ; 4,24
8	5	4,24

- Langkah 2 : Hitung jumlah tetangga dalam radius 5

X	Y	Jumlah Tetangga
1	4	3
2	3	2
3	8	1

X	Y	Jumlah Tetangga
5	2	3
8	5	1

- Langkah 3 : Hitung Local Reachability Density

X	Y	Jarak
1	4	(1,41 + 4,47 + 4,47) / 3 = 3,45
2	3	(1,41 + 3,16) /2 = 2,285
3	8	4,47
5	2	(4,47 + 3,16 + 4,24) / 3 = 3,95
8	5	4,24

- Langkah 4 : Menghitung nilai LOF data

X	Y	LOF
1	4	(1/3,45) x ((2,285 + 4,47 + 3,95) / 3) = 1,03
2	3	(1/2,285) x ((3,45 + 3,95) / 2) = 1,61
3	8	(1/4,47) x ((3,45)) = 0,77
5	2	(1/3,95) x ((3,45 + 2,285 + 4,24) / 3) = 0,83
8	5	(1/4,24) x ((3,95)) = 0.936

Dengan begitu, nilai yang berkemungkinan menjadi outlier adalah baris 2 dan baris 1

Inter Pretasi Local Outlier Factor :

- Jika LOF $>$ 1, itu menunjukkan bahwa kepadatan lokal dari titik p lebih rendah daripada rata-rata kepadatan lokal dari tetangganya. Artinya, titik tersebut memiliki sifat yang “aneh” atau berbeda dari lingkungan sekitarnya dan cenderung menjadi outlier.
- Jika LOF 1, itu menunjukkan bahwa kepadatan lokal dari titik p mirip dengan rata-rata kepadatan lokal tetangganya.
- Jika LOF $<$ 1, itu menunjukkan bahwa kepadatan lokal dari titik p lebih tinggi daripada rata-rata kepadatan lokal dari tetangganya, sehingga cenderung menjadi titik yang tidak aneh.

```
import numpy as np
from scipy import stats
from sklearn.neighbors import LocalOutlierFactor
from sklearn.preprocessing import LabelEncoder
```

```
# menampilkan perbandingan jumlah data berdasarkan target
data['Target'] = LabelEncoder().fit_transform(data['Target'])
data.loc[:,'Target'].value_counts()

# menghitung zscore untuk mendeteksi outliers
clf = LocalOutlierFactor(n_neighbors=20)  # Jumlah tetangga yang digunakan
outlier_scores = clf.fit_predict(data)

# mencari data dengan nilai zscore lebih dari 3
outliers = np.where(outlier_scores == -1)[0]

# menampilkan hasil deteksi outliers
print(f"Total outlier : ", len(outliers))
```

```
Total outlier :  82
```

4

— DATA PREPROCESSING —

Dari tahap data understanding dapat disimpulkan bahwa

1. Tidak terdapat missing value di dalam data
2. Tidak terdapat duplikat data di dalam data
3. Proporsi kolom target pada data telah seimbang sehingga tidak perlu terjadi balancing data

Setelah memahami data, akan dilakukan tahap preprocessing untuk menangani masalah pada data yang sudah didefinisikan pada data understanding, yakni. 1. Menghapus Outlier

Setelah data siap, akan dilakukan : 1. Normalisasi Data 2. Eksplorasi Model

4.1 Menghapus Outlier

Langkah yang dapat diambil untuk menangani terdapatnya outlier pada data adalah dapat dengan menghapus baris-baris data yang di beberapa kolomnya mengandung oulier.

```
# menghapus baris yang termasuk outliers
list_from_outlier_indices = outliers.tolist()
data_cleaned = data[outlier_scores != -1]  # Mengambil baris yang bukan outlier

# memroses ulang data frame dengan tanpa outliers
data_nonoutliers = data_cleaned.reset_index(drop=True)
data_nonoutliers.shape
data_nonoutliers
```

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Prev
0	1	8	5	2	1	1
1	1	6	1	11	1	1
2	1	1	5	5	1	1

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Prev
3	1	8	2	15	1	1
4	2	12	1	3	0	1
...	...	...	...	...	...	...
4337	1	1	6	15	1	1
4338	1	1	2	15	1	1
4339	1	1	1	12	1	1
4340	1	1	1	9	1	1
4341	1	5	1	15	1	1

```
# memisahkan antara kolom fitur dan kolom target
fitur = data_nonoutliers.drop(columns=['Target'])
target = data_nonoutliers['Target']

# menghitung proporsi setiap data berdasarkan target
target.value_counts()
```

```
Target
2    2169
0    1401
1     772
Name: count, dtype: int64
```

```
import pandas as pd

# Membuat DataFrame dari fitur dan target yang telah seimbang
data_fix = pd.concat([fitur, target], axis=1)

# Menyimpan DataFrame ke dalam file CSV
data_fix.to_csv('data_fix.csv', index=False)
```

```
from sklearn.model_selection import train_test_split

# membagi kolom fitur dan kolom target dari file data terbaru
fitur = data_fix.drop(columns=['Target'])
target = data_fix['Target']

fitur_train, fitur_test, target_train, target_test = train_test_split(fitur, target, test_size=
```

4.2 Normalisasi Data

Normalisasi Data adalah salah satu proses dalam pre-processing data untuk mengatur dataset agar memenuhi standar tertentu. Data perlu dilakukan agar dapat mengurangi kemungkinan terjadinya redundansi data. Selain itu, normalisasi dagunakan untuk membantu menghindari anomali dalam pengolahan data dan memungkinkan desain basis data yang lebih efisien.

4.2.1 Menggunakan Standarscaler (zscore)

Metode StandardScaler adalah salah satu teknik normalisasi yang umum digunakan dalam pengolahan data. Tujuannya adalah untuk menyesuaikan distribusi data agar memiliki mean (rata-rata) nol dan standar deviasi satu. Ini berguna saat bekerja dengan algoritma yang sensitif terhadap skala dan asumsi dasar bahwa data terdistribusi normal atau mendekati distribusi normal.

Proses normalisasi menggunakan StandardScaler melibatkan dua langkah utama: 1. Menghitung Mean dan Standar Deviasi: Pertama, perhitungan rata-rata (mean) dan standar deviasi dari setiap fitur (kolom) dalam data dilakukan. 2. Transformasi Data: Setelah mendapatkan mean dan standar deviasi, nilai dari setiap fitur dikurangi dengan mean dari fitur tersebut, kemudian hasilnya dibagi dengan standar deviasi fitur tersebut. Proses ini dilakukan untuk setiap nilai dalam setiap fitur.

```
import pickle
from sklearn.preprocessing import StandardScaler

# menentukan lokasi file pickle akan disimpan
path = 'zscore_scaler.pkl'

# membuat dan melatih objek StandardScaler
zscore_scaler = StandardScaler()
zscore_scaler.fit(fitur_train)

# menyimpan model ke dalam file pickle
with open(path, 'wb') as file:
    pickle.dump(zscore_scaler, file)

# memanggil kembali model normalisasi zscore dari file pickle
with open(path, 'rb') as file:
    zscore_scaler = pickle.load(file)
```

```
# menerapkan normalisasi zscore pada data training
zscore_training = zscore_scaler.transform(fitur_train)

# menerapkan normalisasi zscore pada data testing
zscore_testing = zscore_scaler.transform(fitur_test)
```

4.2.2 Menggunakan Minmaxscaler

MinMax Scaling adalah salah satu teknik untuk melakukan normalisasi pada data dengan merubah nilai-nilai dalam kumpulan data ke dalam rentang tertentu, biasanya antara 0 dan 1. Tujuan utamanya adalah untuk menjaga skala relatif antarfitur agar tidak mendominasi satu sama lain.

Proses Min-Max Scaling dilakukan dengan langkah-langkah berikut: 1. Identifikasi Rentang: Tentukan rentang nilai yang ingin Anda gunakan. Biasanya, dalam Min-Max Scaling, rentang nilai yang dipilih adalah 0 hingga 1, tetapi ini bisa disesuaikan tergantung pada kasus penggunaan. 2. Hitung Nilai Minimum dan Maksimum: Tentukan nilai minimum (min) dan nilai maksimum (max) dari setiap fitur dalam kumpulan data yang akan dinormalisasi. 3. Normalisasi: Gunakan formula rumusn Min-Max Scaling untuk mengubah nilai-nilai dalam rentang yang ditentukan.

```
import pickle
from sklearn.preprocessing import MinMaxScaler

# menentukan lokasi file pickle akan disimpan
path = 'minmaxscaler.pkl'

# membuat dan melatih objek MinMaxScaler
minmaxscaler = MinMaxScaler()
minmaxscaler.fit(fitur_train)

# menyimpan model ke dalam file pickle
with open(path, 'wb') as file:
    pickle.dump(minmaxscaler, file)

# memanggil kembali model normalisasi minmaxscaler dari file pickle
with open(path, 'rb') as file:
    minmaxscaler = pickle.load(file)

# menerapkan normalisasi zscore pada data training
minmaxtraining = minmaxscaler.transform(fitur_train)
```

```
# menerapkan normalisasi zscore pada data testing
minmaxtesting = minmaxscaler.transform(fitur_test)
```

4.3 Eksplorasi Model

4.3.1 Random Forest

Random Forest adalah algoritma pembelajaran terawasi yang digunakan untuk tugas klasifikasi dan regresi dalam machine learning. Ini merupakan bagian dari keluarga algoritma yang dikenal sebagai ensemble learning, yang menggabungkan hasil beberapa model untuk meningkatkan kinerja dan ketepatan prediksi.

Konsep inti dari Random Forest adalah membuat sejumlah besar pohon keputusan saat melakukan prediksi. Setiap pohon keputusan dibuat berdasarkan sampel acak dari data pelatihan dan fitur yang dipilih secara acak. Proses ini mengurangi risiko overfitting (memfitting data pelatihan secara berlebihan) yang sering terjadi pada pohon keputusan tunggal.

Selama proses pelatihan, setiap pohon keputusan dalam hutan acak memilih subset data yang diambil secara acak dan subset fitur untuk membuat keputusan. Ketika melakukan prediksi, setiap pohon memberikan hasilnya, dan hasil akhir dari Random Forest diperoleh dengan mengambil mayoritas suara dari semua pohon keputusan (untuk klasifikasi) atau rerata hasil (untuk regresi).

Kelebihan dari Random Forest termasuk kemampuannya dalam menangani data yang besar dengan fitur yang banyak, serta kemampuan untuk mengatasi overfitting. Namun, seperti halnya dengan banyak algoritma machine learning, pengaturan parameter yang tidak tepat atau kekurangan pemrosesan data yang tepat dapat mempengaruhi kinerja Random Forest.

Random Forest terdiri dari beberapa pohon keputusan yang dibuat secara acak. Untuk setiap pohon keputusan:

- Sampling Data: Lakukan bootstrap sampling pada dataset (ambil sampel acak dengan penggantian).

  Bootstrap Sampling: Proses pengambilan sampel acak dengan penggantian dari dataset yang sama ukuran dengan dataset asli.

- Pemilihan Jumlah Pohon (n_estimators): Tentukan jumlah pohon keputusan yang akan dibuat dalam Random Forest.

- Pemilihan Fitur: Ambil subset acak dari fitur-fitur yang tersedia untuk membangun pohon.
- Pembentukan Pohon: Gunakan algoritma pembentukan pohon (seperti CART atau ID3) untuk membagi data berdasarkan aturan yang paling informatif.

```
from sklearn.feature_selection import SelectKBest, mutual_info_classif
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

best_accuracy_rf_zscore = 0
best_k_zscore = 0
best_accuracy_rf_minmax = 0
best_k_minmax = 0

for k in range(1, fitur_train.shape[1] + 1):
    # Buat objek SelectKBest dengan mutual_info_classif sebagai fungsi skor
    k_best = SelectKBest(score_func=mutual_info_classif, k=k)

    # Fiturkan objek SelectKBest ke data training dan testing untuk kedua normalisasi (zscore d
    zscore_training_terbaik = k_best.fit_transform(zscore_training, target_train)
    zscore_testing_terbaik = k_best.transform(zscore_testing)
    minmaxtraining_terbaik = k_best.fit_transform(minmaxtraining, target_train)
    minmaxtesting_terbaik = k_best.transform(minmaxtesting)

    # Buat dan latih model dengan normalisasi zscore
    model_zscore = RandomForestClassifier(random_state=42)
    model_zscore.fit(zscore_training_terbaik, target_train)

    # Lakukan prediksi pada data uji dengan normalisasi zscore
    y_pred_rf_zscore = model_zscore.predict(zscore_testing_terbaik)

    # Hitung akurasi dengan normalisasi zscore
    accuracy_rf_zscore = accuracy_score(target_test, y_pred_rf_zscore)
    # print(k, "fitur menghasilkan akurasi : ", accuracy_rf_zscore)

    # Buat dan latih model dengan normalisasi minmax
    model_minmax = RandomForestClassifier(random_state=42)
    model_minmax.fit(minmaxtraining_terbaik, target_train)

    # Lakukan prediksi pada data uji dengan normalisasi minmax
    y_pred_rf_minmax = model_minmax.predict(minmaxtesting_terbaik)

    # Hitung akurasi dengan normalisasi minmax
```

```
        accuracy_rf_minmax = accuracy_score(target_test, y_pred_rf_minmax)
        # print(k, "fitur menghasilkan akurasi : ", accuracy_rf_minmax)

        # Memeriksa apakah akurasi dengan normalisasi zscore lebih baik dari yang sebelumnya
        if accuracy_rf_zscore > best_accuracy_rf_zscore:
            best_accuracy_rf_zscore = accuracy_rf_zscore
            best_k_zscore = k

        # Memeriksa apakah akurasi dengan normalisasi minmax lebih baik dari yang sebelumnya
        if accuracy_rf_minmax > best_accuracy_rf_minmax:
            best_accuracy_rf_minmax = accuracy_rf_minmax
            best_k_minmax = k

    print("Dengan Normalisasi Zscore:")
    print("Fitur terbaik yang bisa digunakan", best_k_zscore, "dengan akurasi : ", best_accuracy_rf

    print("Dengan Normalisasi Minmax:")
    print("Fitur terbaik yang bisa digunakan", best_k_minmax, "dengan akurasi : ", best_accuracy_rf

    # Ambil indeks fitur terbaik dari SelectKBest
    best_feature_indices_zscore = SelectKBest(score_func=mutual_info_classif, k=best_k_zscore).fit(
    best_feature_indices_minmax = SelectKBest(score_func=mutual_info_classif, k=best_k_minmax).fit(

    # Dapatkan nama fitur terbaik dari indeksnya
    best_features_zscore = [fitur.columns[i] for i in best_feature_indices_zscore]
    best_features_minmax = [fitur.columns[i] for i in best_feature_indices_minmax]


    print("Fitur terbaik yang digunakan (dengan normalisasi Zscore):")
    print(best_features_zscore)

    print("Fitur terbaik yang digunakan (dengan normalisasi Minmax):")
    print(best_features_minmax)
```

```
Dengan Normalisasi Zscore:
Fitur terbaik yang bisa digunakan 24 dengan akurasi :  0.7753222836095764
Dengan Normalisasi Minmax:
Fitur terbaik yang bisa digunakan 18 dengan akurasi :  0.7780847145488029
Fitur terbaik yang digunakan (dengan normalisasi Zscore):
['Application mode', 'Course', 'Previous qualification', "Father's qualification", "Mother's occupati
Fitur terbaik yang digunakan (dengan normalisasi Minmax):
['Application mode', 'Course', "Mother's occupation", 'Debtor', 'Tuition fees up to date', 'Scholarshi
```

5

— *MODELLING* —

Setelah dilakukan skenario perulangan untuk menghasilkan model terbaik, dapat diketahui bahwasannya model klasifikasi yang terbaik untuk data anggur merah ini adalah dengan menggunakan : - Metode Random Forest - Metode normalisasi nya adalah MINMAX Scaler - Banyak Fitur yang digunakan dalam data sebanyak 23 fitur - Parameter dalam metode yang akan digunakan, sebagai berikut:

- jumlah estimator : 50, 100, 200
- maksimal kedalaman : none, 10, 20
- minimal pembagian sampel : 2, 5, 10
- minimal sampel daun : 1, 2, 4

```
import pandas as pd

# kolom-kolom yang ingin Anda pertahankan
kolom_pilihan = ['Application mode', 'Course', 'Previous qualification', "Father's qualificatio
                 'Tuition fees up to date', 'Scholarship holder', 'Age at enrollment', 'Curricu
                 'Curricular units 1st sem (evaluations)', 'Curricular units 1st sem (approved)
                 'Curricular units 2nd sem (enrolled)', 'Curricular units 2nd sem (evaluations)
                 'Curricular units 2nd sem (grade)','Target']

# Buat dataset baru hanya dengan kolom yang dipilih
dataset_baru = data_nonoutliers[kolom_pilihan]

# Simpan dataset baru dalam file CSV
dataset_baru.to_csv('dataset_baru.csv', index=False)  # Simpan ke file CSV tanpa menyertakan in
```

```
from sklearn.model_selection import train_test_split
```

```
# memisahkan kolom fitur dan target
fitur = dataset_baru.drop(columns=['Target'], axis =1)
target = dataset_baru['Target']

# melakukan pembagian dataset, dataset dibagi menjadi 80% data training dan 20% data testing
fitur_train, fitur_test, target_train, target_test = train_test_split(fitur, target, test_size =
```

```
import pickle
from sklearn.preprocessing import MinMaxScaler

# menentukan lokasi file pickle akan disimpan
path = 'minmaxscaler_baru.pkl'

# membuat dan melatih objek MinMaxScaler
minmaxscaler = MinMaxScaler()
minmaxscaler.fit(fitur_train)

# menyimpan model ke dalam file pickle
with open(path, 'wb') as file:
    pickle.dump(minmaxscaler, file)

# memanggil kembali model normalisasi minmaxscaler dari file pickle
with open(path, 'rb') as file:
    minmaxscaler = pickle.load(file)

# menerapkan normalisasi zscore pada data training
minmaxtraining = minmaxscaler.transform(fitur_train)

# menerapkan normalisasi zscore pada data testing
minmaxtesting = minmaxscaler.transform(fitur_test)
```

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier

# mendefinisikan ruang parameter
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# membuat model
```

```
model = RandomForestClassifier()

# menggunakan modul gridsearch untuk mencari parameter terbaik
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5, scoring='accuracy', ve
grid_search.fit(fitur_train, target_train)

best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

# menampilkan parameter terbaik
print("Best Parameters:", best_params)
print("Best Model:", best_model)
```

```
Fitting 5 folds for each of 81 candidates, totalling 405 fits
```

```
Best Parameters: {'max_depth': 20, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200
Best Model: RandomForestClassifier(max_depth=20, min_samples_leaf=2, n_estimators=200)
```

```
import pickle

# menentukan lokasi file pickle akan disimpan
path_rf = 'best_model.pkl'
with open(path_rf, 'wb') as model_file:
    pickle.dump(best_model, model_file)
```

6

— EVALUATION —

Pada tahap ini model terbaik yang diperoleh pada tahap modeling dilakukan validasi dengan menampilkan nilai confusion matrix nya atau laporan klasifikasinya dengan menggunakan grafik ROC-AUC

6.1 ## CONFUSION MATRIX

Confusion matrix adalah sebuah tabel yang digunakan dalam evaluasi kinerja model klasifikasi untuk memahami performa model dalam memprediksi kelas-kelas target. Matrix ini memiliki empat sel yang mewakili:

1. True Positive (TP): Prediksi yang benar ketika kelas sebenarnya adalah positif.
2. True Negative (TN): Prediksi yang benar ketika kelas sebenarnya adalah negatif.
3. False Positive (FP): Prediksi yang salah ketika model memprediksi positif tetapi kelas sebenarnya negatif (juga dikenal sebagai Type I error).
4. False Negative (FN): Prediksi yang salah ketika model memprediksi negatif tetapi kelas sebenarnya positif (juga dikenal sebagai Type II error).

Bentuk dari tabel Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

Dari Confusion Matriks, kta dapat menghitung metrik evaluasi seperti akurasi, presisi, recall, F1-score, dan lainnya yang membantu dalam mengevaluasi performa model klasifikasi.

6.1.0.1 Metrik Evaluasi

Metrik evaluasi adalah ukuran atau parameter yang digunakan untuk mengevaluasi kinerja suatu model atau sistem dalam melakukan tugas tertentu, seperti klasifikasi, regresi, atau tugas lainnya dalam bidang machine learning dan statistika. Metrik-metrik ini membantu dalam memahami seberapa baik atau buruk model tersebut dalam melakukan prediksi atau tugas yang ditetapkan.

Beberapa metrik evaluasi umum dalam machine learning termasuk: > - Akurasi (Accuracy): Seberapa sering model memberikan prediksi yang benar secara keseluruhan. Rumus Akurasi : $$Accuracy = \frac{TN + TP}{TN + FP + FN + TP}$$

> - Presisi (Precision): Proporsi dari prediksi positif yang benar dibandingkan dengan semua prediksi positif yang dibuat oleh model Rumus Precision : $$Precision = \frac{TP}{TP + FP}$$

> - Recall (Sensitivity atau True Positive Rate): Proporsi dari kelas positif yang diprediksi dengan benar oleh model. Rumus Recall : $$Recall = \frac{TP}{TP + FN}$$

> - F1-Score: Nilai rata-rata harmonik antara presisi dan recall. Berguna ketika perlu menyeimbangkan antara presisi dan recall. Rumus F1-Score : $$F1 - Score = 2x\frac{PresisixRecall}{PresisixRecall}$$

> - Specificity (Specificity atau True Negative Rate): Proporsi dari kelas negatif yang diprediksi dengan benar oleh model. Rumus Specificity : $$Specificity = \frac{TN}{TN + FP}$$

```
with open(path_rf, 'rb') as file:
    model_rf = pickle.load(file)

from sklearn.metrics import accuracy_score

model_rf.fit(minmaxtraining, target_train)
y_pred_rf = model_rf.predict(minmaxtesting)

# akurasi
akurasi_rf = accuracy_score(target_test, y_pred_rf)
print('Akurasi Random Forest : ', akurasi_rf)
```

```
Akurasi Random Forest :  0.7744533947065593
```

```
from sklearn.metrics import confusion_matrix, classification_report

# Mengukur akurasi
accuracy = accuracy_score(target_test, y_pred_rf)
print(f'Akurasi: {accuracy}')

# Menghasilkan dan menampilkan confusion matrix
cm = confusion_matrix(target_test, y_pred_rf)
print("Confusion Matrix:")
print(cm)

# Mendapatkan nilai TP, TN, FP, FN dari confusion matrix
TN = cm[0, 0]
FP = cm[0, 1]
FN = cm[1, 0]
TP = cm[1, 1]

print("\nTrue Positive (TP):", TP)
print("True Negative (TN):", TN)
print("False Positive (FP):", FP)
print("False Negative (FN):", FN)

# Menampilkan classification report
print("Classification Report:")
print(classification_report(target_test, y_pred_rf))
```

```
Akurasi: 0.7744533947065593
Confusion Matrix:
[[205  23  38]
 [ 35  50  69]
 [ 12  19 418]]

True Positive (TP): 50
True Negative (TN): 205
False Positive (FP): 23
False Negative (FN): 35
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.77      0.79       266
           1       0.54      0.32      0.41       154
           2       0.80      0.93      0.86       449

    accuracy                           0.77       869
```

```
   macro avg       0.72      0.68      0.69       869
weighted avg       0.76      0.77      0.76       869
```

7

Summary

In summary, this book has no content whatsoever.

References