Jane Doe and Max Power

Quarto CRC Book

To blah, blah, and blah.

Table of contents

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 PROYEK SAINS DATA

- Nama : Soni Indra Maulana
- NIM : 210411100136
- Kelas : B

1.1 BUSSINESS UNDERSTANDING

1.1.1 TUJUAN BISNIS

Meningkatkan nilai jual salah satu produk wine yaitu produk red wine Vinho Verde dengan meningkatkan kualitas produk red wine milik perusahaan Vinho Verde

1.1.2 TUJUAN PENAMBANGAN DATA

Menilai kualitas pada anggur merah (red wine) yang diproduksi oleh perusahaan vinho verde dengan menggunakan model klasifikasi, kualitas anggur dapat dinyatakan dengan menentukan nilai regressi seperti berikut:

- Kualitas Anggur lebih besar sama dengan 7 maka dinyatakan “good” atau bernilai 1
- Kualitas Anggur di bawah angka 7 maka dinyatakan “not good” atau bernilai 0

Untuk menentukan kualitas dari anggur dapat dilihat dari ciri-ciri sebagai berikut:

1.1.3 DESKRIPSI FITUR

- **Fixed acidity (Kadar asam tartarat pada anggur) (g/dm3)**
- **Volatile acidity (Kadar asam asetat pada anggur)(g/dm3)**
- **Citric acid (Kadar asam sitrat dalam anggur) (g/dm3)**
- **Residual Sugar(Kadar sisa gula pada wine) (g/dm3)**
- **Chlorides (Kadar natrium klorida dalam anggur) (g/dm3)**
- **Free sulfur dioxide (Tingkat sulfur dioksida bebas dalam anggur) (mg/dm3)**
- **Total sulfur dioxide (SO2 Total = SO2 bebas + SO2 bereaksi) (mg/dm3)**
- **Density(Kepadatan anggur) (g/cm3)**
- **pH(tingkat keasaman anggur)(Range: 0 - 14)**
- **Sulphates(Kadar kalium sulfat dalam anggur)(g/dm3)**
- **Alcohol(Konsentrasi alkohol dalam anggur)(% vol.)**

1.1.4 RENCANA PROYEK

Untuk mencapai Tujuan Penambangan Data serta Tujuan Bisnis diatas. Maka, terdapat serangkaian langkah-langkah yang perlu dilakukan diantaranya: - Mengumpulkan Sumber Daya : Sumber Daya yang akan diolah sebagai masukan adalah data penilaian serta kandungan dari salah satu produk anggur milik perusahaan vinho verde yaitu anggur merah . Data yang dimasukkan adalah data yang telah di berikan label berdasarkan penilaian dari anggur merah serta beberapa ciri didalam anggur yang mendukung pengelompokkan anggur berdasarkan label nya

- Mengelompokkan Sumber Daya : Sesui dengan tujuan penambangan data nya. Maka dari itu, diasumsikan terdapat dua jenis kategori dari kualitas anggur merah ini yaitu anggur merah dengan kualitas baik dan buruk sehingga banyak kelas dari sebelumnya berupa rentang angka dari 1 sampai 10 menjadi data tipe binary yang hanya memiliki dua jenis, yaitu 1 (baik) dan 0 (buruk)

- Analisis Sumber Daya : Analisis perlu dilakukan dikarenkan untuk mengetahui tahap-tahap pre-processing apa saja yang diperlukan untuk mengatasi berkurangnya kualitas data yang akan diolah, terdapat beberapa analisis yang perlu dilakukan seperti : analisis tipe data apa saja didalm data, analisis missing values pada data, analsis oulier pada data serta analsis proporsi jumlah kelas didalm data.

- Pre-processing Sumber Daya : sebelum memasuki tahap modeling, data terlebih dahulu harus dalam kualitas yang sangat baik dengan menangani permasalahan pada data yang telah diketahui dan dianalisis pada tahap analisis sumber daya. Dengan begitu, pada tahap ini telah diketahui apa saja langkah-langkah yang harus dilakukan untuk meningkatkan kualitas dari data, seperti penyeimbangan jumlah kelas dalam data, penanganan

missing values dan outlier pada data. Selain penanganan terhadap kesalahan dalam data, pada tahap ini dilakukan juga split atau pembagian data menjadi 2 bagian yatu data latih(training) dan data uji(testing), pembagian perlu dilakukan untuk melatih model untuk menentukan model yang terbaik untuk data nantinya di tahap selanjutnya. Pada tahap ini dapat ditambahkan pemilihan fitur terbaik dengan me-ranking setiap fitur dalam data sehingga dapat diketahui fitur-fitur apa saja yang paling berpengaruh dan paling tidak berpengaruh pada penentuan kelas pada data.

- Skenario Percobaan : tahap ini digunakan untuk melakukan pencarian terhadap metode dan parameter nya yang cocok pada data. Dikatakan cocok dengan data jika akurasi yang didapatkan dari metode dan paramter tersebut dinilai besar dan lebih baik dari metode dan parameter nya yang lain. Untuk mengetahui nya maka perlu dilakukan percobaan untuk mengetahui metode dan parameter nya yang dapat menghasilkan nilai akurasi yang besar.
- Modelling : setalah data dalam keadaan berkualitas yang baik, maka untuk melakukan prediksi kualitas anggur merah sebelumnya perlu dilakukan pemilihan metode serta parameter terbaik nya. Pada tahap ini, metode serta parameter yang dinilai terbaik untuk data di terapkan pada data setelah itu ekstraksi model dilakukan dengan berisikan metode serta parameter terbaik nya yang disimpan dalam sebuah file, yang dimana nanti untuk melakukan prediksi pada data uji model ini lah yang akan di load dan diterapkan.
- Evaluasi Model : Pada tahap ini model terbaik yang telah diciptakan dilakukan evaluasi dengan melihat akurasi yang didapatkannya serta dapat dengan membuat tabel kebingungan serta grafik kurva ROC-AUC dari hasil modelling data dengan metode serta parameter terbaik nya. Disini, hasil modelling dapat dinilai dan evaluasi tingkat keakuratan nya
- Deployment : Disini prediski pada data uji dapat diterapkan dengan mengimpementasikan model terbaik yang telah di ekspor tadi kedalam data baru yang belum diketahui kelas nya apakah termasukka kedalm kelas kualitas baik atau buruk. Dimulai dengan input data yang akan diuji lalu model terbaik yang telah di load lalu diterapkan dalam data uji yang barusan diinputkan sehingga dapat melakukan prediksi kualitas anggur merah terhadap data uji yang baru tersebut.

1.2 DATA UNDERSTANDING

1.2.1 TEKNIK PENGUMPULAN DATA

Data yang digunakan pada proyek ini adalah data dari salah satu produk unik berupa red wine yang diproduksi oleh perusahaan vinho verde dari wilayah Minho (barat laut) Portugal.Data dikumpulkan dari Mei 2004 hingga Februari 2007 hanya dengan menggunakan proteksi penunjukan sampel asal yang diuji di badan sertifikasi resmi (CVRVV).CVRVV adalah organisasi antarprofesional dengan tujuan meningkatkan kualitas dan pemasaran vinho verde. Datanya adalah dicatat oleh sistem komputerisasi (iLab), yang secara otomatis mengelola proses pengujian sampel anggur mulai dari permintaan produsen hingga analisis laboratorium dan sensorik. Setiap entri menunjukkan tes tertentu (analitis atau sensorik) dan database akhir diekspor ke dalam satu lembar (.csv). Tentang preferensi, setiap sampel dievaluasi oleh minimal tiga penilai sensorik (menggunakan pengecap buta),yang menilai anggur dalam skala yang berkisar dari 0 (sangat buruk) hingga 10 (sangat baik).

```
# Berikut Library apa saja yang perlu di import untuk membantu pemrosesan pada data
import numpy as np
import seaborn as sns
import pandas as pd
from sklearn.neighbors import LocalOutlierFactor
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV
from imblearn.over_sampling import RandomOverSampler
from collections import Counter
import joblib
from sklearn.model_selection import train_test_split
import pickle
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score,roc_curve
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import mutual_info_classif
import matplotlib.pyplot as plt
```

```
# Me-load file red-wine dengan ekstensi csv
data = pd.read_csv('winequality-red.csv')
```

```
# Untuk Mengetahui Keunikan Kelas Dalam Data
data['quality'].unique()
```

```
array([5, 6, 7, 4, 8, 3], dtype=int64)
```

1.2.2 JUMLAH DATASET

Jumlah Dataset sebanyak 1599 record. Dikarenakan pada data red wine pada kolom 'quality' berisikan rentang nilai dari 0 sampai 10. Maka, untuk menentukan kualitas baik atau buruk nya red wine dapat dengan memasukkan data dengan rentang 0 sampai 6 termasuk kedalam golongan kualitas anggur yang buruk (0) dan memasukkan data dengan rentang 7 sampai 10 kedalam golongan kualitas anggur yang baik (1)

Berikut Hasil pembagian kelas didalm data : - Berkualitas buruk (0) = 1382 data - Berkualitas baik (1) = 217 data

```
# untuk membagi data menjadi fitur dan target
fitur = data.drop('quality', axis= 1)
target = data['quality'].apply(lambda y_value: 1 if y_value >= 7 else 0)
```

```
#untuk menghitung berapa banyak kelas dan jumlah nya dalam sekumpulan data
target.value_counts()
```

```
quality
0    1382
1     217
Name: count, dtype: int64
```

1.2.3 ANALISIS STATISTIK

Analisis statistik pada data adalah proses mengumpulkan, menyusun, meringkas, menginterpretasi, dan membuat kesimpulan dari data menggunakan metode statistik. Tujuan utamanya adalah untuk memahami pola, tren, atau hubungan di dalam data yang telah dikumpulkan.

Deskripsi Statistik: Ini mencakup teknik untuk merangkum dan menggambarkan data, seperti mean (rata-rata), median (nilai tengah), modus (nilai yang paling sering muncul), kuartil, dan deviasi standar.

Deskripsi Statistik digunakan untuk memberikan nilai statistik deskriptif dari suatu dataframe dengan hasil sebagai berikut- :

- Count : jumlah data yang bernilai didalam kolom.
- Mean : Nilai rata-rata nilai setiap kolom
- std (standar deviasi) : Sebaran data, standar deviasi yang tinggi menunjukka bahwa data cenderung lebih tersebar.
- Min : menunjukkan nilai terkecil pada setiap kolom
- 25% atau Q1 : Menunjukkan kuartil pertama, nilai yang membagi 25% data terandah.
- 50% : Median atau Q2, nilai tengah dari data .

- 75% atau Q3 :Nilai yang membagi 75% data terendah atau batas atas .
- max : Nilai terbesar dalams etiap kolom.

```
data.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	tota
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	159
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.4
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.8
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.00
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.0
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.0
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.0
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289

1.2.4 DESKRIPSI FITUR

- **Fixed acidity (Kadar asam tartarat pada anggur) (g/dm3)**

  Asam tetap (Fixed acidity) yang dominan ditemukan dalam anggur mengacu pada total asam-asam yang terdapat dalam anggur yang tidak teruapkan selama fermentasi atau proses lainnya seperti tartarat, malat, sitrat, dan suksinat. Tingkat masing-masing yang ditemukan dalam anggur bisa sangat bervariasi tetapi dalam secara umum orang akan mengharapkan untuk melihat 1.000 hingga 4.000 mg/L asam tartarat, 0 hingga 8.000 mg/L asam malat, 0 hingga 500 mg/L asam sitrat, dan 500 hingga 2.000 mg/L asam suksinat. Secara umum, kadar fixed acidity yang ideal dalam anggur merah berkisar antara 5 hingga 6 gram per liter.

- **Volatile acidity (Kadar asam asetat pada anggur)(g/dm3)**

  Volatil acidity atau keasaman yang mudah menguap dalam anggur mengacu pada kandungan asam-asam yang bisa menguap pada suhu ruangan. Asam-asam ini, seperti asam asetat, terutama berkontribusi pada karakteristik rasa dan aroma anggur. Asam asetat adalah salah satu dari beberapa komponen yang memberikan rasa pada anggur. Kadar volatile acidity yang dianggap “terbaik” dalam anggur bervariasi tergantung pada jenis anggur, gaya pembuatan anggur, dan preferensi individu. Secara umum, kandungan VA yang diinginkan dalam anggur merah, batasnya cenderung sedikit lebih tinggi, mungkin sekitar 0,5 gram per liter.

- **Citric acid (Kadar asam sitrat dalam anggur)(g/dm3)**

Asam Sitrat adalah asam yang memberikan kontribusi pada rasa asam yang lembut pada anggur. Asam sitrat memiliki banyak kegunaan dalam produksi anggur. Asam sitrat adalah bahan organik lemah asam, yang sering digunakan sebagai pengawet alami atau bahan tambahan pada makanan atau minuman untuk menambah rasa asam pada makanan. Kadar asam sitrat dalam anggur merah bisa bervariasi, namun secara umum, anggur merah cenderung memiliki kadar asam sitrat yang lebih rendah dibandingkan dengan jenis asam lainnya seperti asam tartarat. Kandungan asam sitrat dalam anggur merah biasanya berkisar antara 0,1 hingga 0,7 gram per liter.

- **Residual Sugar(Kadar sisa gula pada wine)(g/dm3)**

  Gula Residu berasal dari gula anggur alami yang tersisa dalam anggur setelahnya fermentasi alkohol selesai. Itu diukur dalam gram per liter. Sisa kadar gula bervariasi pada berbagai jenis anggur. Anggur kering memiliki kadar residual sugar yang rendah, biasanya kurang dari 10 gram per liter (g/L). Anggur kering memiliki sedikit rasa manis.

- **Chlorides (Kadar natrium klorida dalam anggur)(g/dm3)**

  Klorida adalah salah satu komponen kimia yang bisa ditemukan dalam anggur. Kadar klorida dalam anggur dapat berasal dari air irigasi, tanah tempat anggur tumbuh, atau dari pupuk yang digunakan selama pertumbuhan tanaman. Kandungan klorida ini bisa mempengaruhi karakteristik dan kualitas anggur. Maksimal konsentrasi klorida dalam anggur sekitar 0,20 - 0,60 g/L

- **Free sulfur dioxide (Tingkat sulfur dioksida bebas dalam anggur)(mg/dm3)**

  Sulfur dioksida (SO2) adalah keadaan dimana aat sulfur dioksida larut dalam anggur, sebagian dari sulfurnya bereaksi menjadi bentuk bebas. Kadar SO2 bebas yang dianggap ideal dalam anggur merah dapat bervariasi tergantung pada beberapa faktor, termasuk jenis anggur, kondisi penyimpanan, dan preferensi pembuat anggur.Secara umum, kadar SO2 bebas dalam anggur merah biasanya berada dalam kisaran 10 hingga 40 mg/L

- **Total sulfur dioxide (SO2 Total = SO2 bebas + SO2 bereaksi) (mg/dm3)**

  Total sulfur dioxide (SO2) adalah ukuran untuk jumlah keseluruhan sulfur dioksida yang ada dalam anggur. Sulfur dioksida digunakan secara umum sebagai bahan tambahan dalam produksi anggur untuk melindungi anggur dari oksidasi dan infeksi mikroba. Dapat dinyatakan berupa Bagian dari sulfur dioksida bebas ditambah bagian yang terikat dengan bahan kimia lain di dalamnya anggur seperti aldehida, pigmen, atau gula. Untuk anggur merah, biasanya kadar SO2 yang dianjurkan berkisar antara 30 hingga 50 mg/L (miligram per liter), tetapi angka ini dapat bervariasi berdasarkan

regulasi lokal, jenis anggur, dan praktik pembuatan anggur yang digunakan oleh produsen.

- **Density(Kepadatan anggur) (g/cm3)**

  Density (kepadatan) dalam konteks anggur merujuk pada kepekatan relatif dari komponen-komponen dalam anggur, terutama gula dan alkohol yang terlarut dalam jus anggur, biasanya diukur dalam satuan Brix. Brix adalah unit yang digunakan untuk mengukur kadar padatan terlarut dalam cairan. Kadar Brix yang dianggap baik untuk anggur merah bisa berkisar antara 22 hingga 25 Brix.

- **pH(tingkat keasaman anggur)(Range: 0 - 14)**

  Tingkat keasaman dalam anggur merah adalah salah satu komponen penting yang mempengaruhi rasa dan karakteristik anggur. Keasaman dalam anggur biasanya diukur dengan pH, yang mengindikasikan tingkat keasaman relatif dalam cairan tersebut. Rentang pH yang umum untuk anggur merah adalah sekitar 3 hingga 4.

- **Sulphates(Kadar kalium sulfat dalam anggur)(g/dm3)**

  Sulfit, juga biasa disebut sulfur dioksida, adalah senyawa kimia yang mengandung ion sulfit.Sulfit adalah zat yang sering digunakan sebagai pengawet dan antioksidan dalam anggur. Ini digunakan untuk mencegah oksidasi anggur dan pertumbuhan mikroorganisme yang tidak diinginkan. Kadar sulfat yang direkomendasikan dalam anggur merah berkisar antara 10 hingga 70 miligram per liter (mg/L). Namun, peraturan tentang batas maksimum penggunaan sulfat dalam anggur dapat berbeda di berbagai negara.

- **Alcohol(Konsentrasi alkohol dalam anggur)(% vol.)**

  Alkohol dalam anggur adalah etanol yang dihasilkan melalui proses fermentasi gula dalam anggur oleh ragi atau mikroorganisme lainnya. Kadar alkohol dalam anggur dapat bervariasi tergantung pada beberapa faktor, seperti jenis anggur, kondisi iklim tempat tumbuhnya anggur, serta teknik pembuatan dan proses fermentasi yang digunakan. Kadar alkohol yang dianggap ideal dalam anggur merah biasanya berkisar antara 12% hingga 15%.

```
# untuk mengetahui fitur apa saja dalam data
data.columns
```

```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
    'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
      'pH', 'sulphates', 'alcohol', 'quality'],
     dtype='object')
```

1.2.5 TIPE DATA

Tipe Data didalam data terdapat dua jenis yaitu Tipe data Rasio dan Tipe Data Numerik

TIPE DATA RASIO

adalah tipe data yang mengacu pada jenis data yang memiliki sifat kuantitatif dan memiliki nol absolut yang bermakna.Contoh tipe data rasio adalah variabel yang melibatkan pengukuran yang memiliki nol yang bermakna, seperti panjang, berat, suhu, dan waktu.Operasi aritmatika seperti penambahan, pengurangan, perkalian, dan pembagian dapat dilakukan pada tipe data rasio ini. Nilai-nilai pada tipe data rasio dapat dibandingkan dan digunakan untuk perbandingan rasio atau perbandingan persentase

TIPE DATA NUMERIK

adalah tipe data yang mencakup sejumlah besar nilai numerik, yang dapat berupa bilangan bulat (integer) atau bilangan pecahan (floating-point).Bilangan bulat adalah bilangan utuh tanpa bagian desimal seperti -3, 0, 7, dan sebagainya.Bilangan pecahan atau floating-point adalah bilangan dengan bagian desimal seperti 3.14, 0.5, -8.75, dan sejenisnya.Data numerik dapat digunakan untuk operasi matematika seperti penambahan, pengurangan, perkalian, pembagian, serta operasi lainnya.

Tipe data dari disetiap kolom pada data anggur merah vinho verde sebagai berikut

1. Tipe Rasio :

Tipe data Rasio adalah

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol

---

2. Tipe Numerik

Tipe data Nominal adalah

- quality

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

1.2.6 IDENTIFIKASI DUPLIKAT PADA DATA

Duplikat data adalah terdapat record atau baris yang identik atau sangat mirip dalam hal nilai-nilai di beberapa kolom. Ini bisa menjadi masalah karena dapat mengganggu analisis data dan mempengaruhi keakuratan hasil yang diperoleh dari query atau pemrosesan data.

Hasil Identifikasi Duplikat Data : - Didalam data red wine setelah di identifikasi di temukan duplikat data dalam tabel red wine yaitu sebanyak : 240 data

```
jumlah_duplikat = data.duplicated().sum()
print("Jumlah data duplikat:", jumlah_duplikat)
```

```
Jumlah data duplikat: 240
```

1.2.7 IDENTIFIKASI MISSING VALUES

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.

Hasil Identifikasi Missing Values :

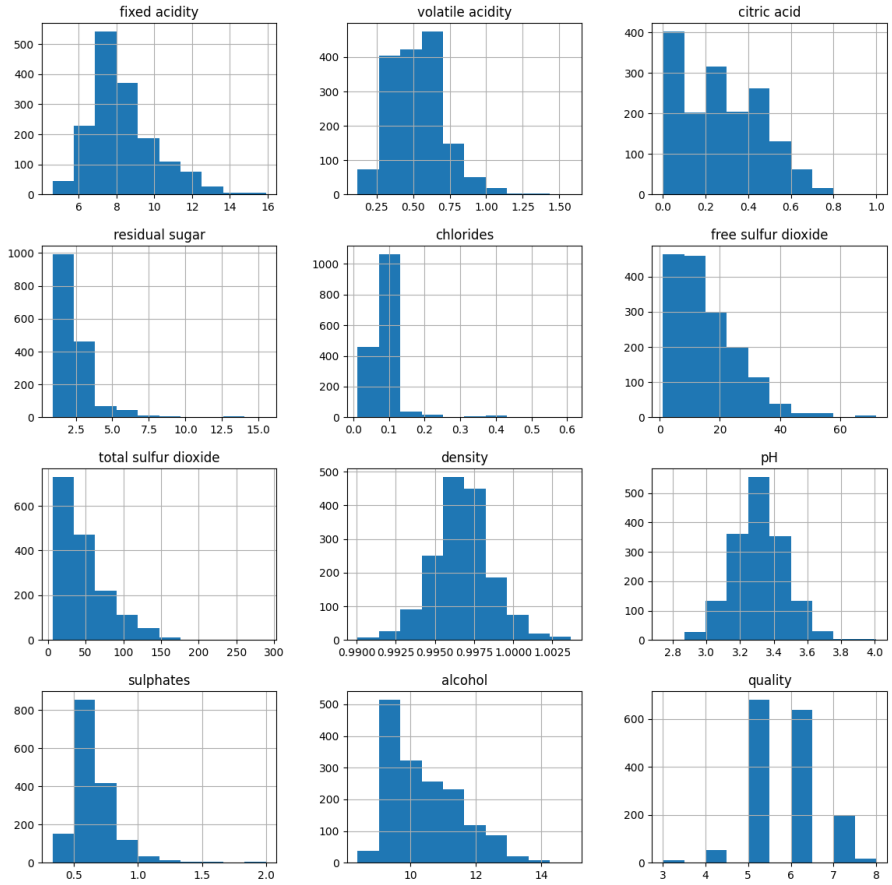
- Di dalam data sampel anggur merah setelah dilakukan pengecekan nilai tidak ditemukan data dengan nilai yang hilang (Missing Values).

```
# untuk mengecek apakah terdapat missing values pada data
data.isnull().sum()
```

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

1.2.8 VISUALISASI DATA

```
# menampilkan diagram batang setiap kolom dalam data
data.hist(figsize=(14, 14))
plt.show()
```



1.2.9 IDENTIFIKASI OUTLIER PDA DATA

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.

Hasil Identifikasi Outlier pada Data :

- Setelah dilakukan pengecekan outlier dengan menggunakan teknik Local Outlier Factor diketahui terdapat 31 data yang terdeteksi mengandung outlier.

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.

```
# Membuat model LOF
clf = LocalOutlierFactor(n_neighbors=20)  # Jumlah tetangga yang digunakan
outlier_scores = clf.fit_predict(data) # untuk menghitung nilai LOF

# Menampilkan indeks outlier
```

```
outlier_indices = np.where(outlier_scores == -1)[0] # menampilkan indeks/baris apa saja yang me
print("Indeks outlier:",outlier_indices)
print("Indeks outlier:",len( outlier_indices))
```

```
Indeks outlier: [ 324  325  354  374  396  400  442  444  480  530  535  538  559  564
  652  773  774  911  917  923  982 1043 1079 1081 1131 1186 1235 1244
 1478 1558 1574]
Indeks outlier: 31
```

1.2.10 IDENTIFIKASI PROPORSI JUMLAH MASING-MASING KELAS DALAM 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.

Hasil Identifikasi Proporsi Jumlah Masing-Masing Kelas dalam data :

Berdasarkan jumlah data pada setiap kelas yang telah di definisikan sebelumnya, dapat diketahui bahwasan nya pada data jumlah masing-masing kelas tidak seimbang dengan perbandingan :

- 86.4% dari jumlah seluruh data untuk data yang diidentifikasi memiliki kualitas yang buruk
- 13.5% dari jumlah seluruh data untuk data yang diidentifikasi termasuk memiliki kualitas yang bagus.

```
#untuk menghitung berapa banyak kelas dan jumlah nya dalam sekumpulan data
target.value_counts()
```

```
quality
0    1382
1     217
Name: count, dtype: int64
```

1.3 PREPROCESSING DATA

Dari Data Understanding diatas dapat disimpulkan bahwasannya:

- Didalam data tidak memiliki Missing Values

- Didalam data terdapat duplikat data
- Didalam data juga terdapat outlier yang terdeteksi
- Terjadi Inbalancing pada jumlah masing-masing kelas dalam data

Maka dari itu perlu dilakukan beberapa langkah preprocessing pada data seperti:

1.3.1 PENANGANAN DUPLIKAT DATA

Setelah dilakukan Identifikasi duplikat data pada tabel red wine. Maka, untuk melakukan penanganan pada masalah ini maka dapat dilakukan penghapusan data duplikat yang telah diidentifikasi sebelumnya di tahap data understanding

```
# variabel data yang menyimpan data dilakukan drop atau penghapusan data yang  duplikat
data.drop_duplicates(inplace=True)
```

1.3.2 PENANGANAN OUTLIER PADA DATA

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

```
clf = LocalOutlierFactor(n_neighbors=20)  # Jumlah tetangga yang digunakan
outlier_scores = clf.fit_predict(data) # untuk menghitung nilai LOF

# Menampilkan indeks outlier
outlier_indices = np.where(outlier_scores == -1)[0]
list_from_outlier_indices = outlier_indices.tolist() # untuk mengubah tiped data pada variabel
data_cleaned = data[outlier_scores != -1]  # Mengambil baris yang bukan outlier

print("Data sebelum penghapusan outlier:",len(data))  # Menampilkan panjang dari data sebelum m
print("Data setelah penghapusan outlier:", len(data_cleaned)) # Menampilkan panjang dari data s
```

```
Data sebelum penghapusan outlier: 1359
Data setelah penghapusan outlier: 1334
```

1.3.3 PENANGANAN PROPORSI JUMLAH MASING-MASING KELAS PADA DATA

Dikarenakan terdapat ketidakseimbangan pada data disetiap kelasnya maka diperlukan sebuah proses balancing data agar jumlah data pada setiap kelas-

nya dapat seimbang untuk menanganin ketidakseimbangan data maka diperlukan suatu metode yaitu metode Random Over Sampling yang akan membuat jumlah data pada masing-masing kelas menjadi sama sehingga seimbang

```
fitur_cleaned = data_cleaned.drop('quality', axis= 1) # memasukkan fitur selain kolom quality k
target_cleaned = data_cleaned['quality'].apply(lambda y_value: 1 if y_value >= 7 else 0) # mema
```

BALANCING DATA

Balancing data adalah proses memastikan bahwa dataset yang digunakan untuk melakukan analisis atau pelatihan model memiliki distribusi yang seimbang di antara kelas-kelas yang berbeda. Hal ini sering kali terjadi dalam konteks klasifikasi di mana terdapat lebih sedikit contoh atau sampel untuk satu kelas dibandingkan dengan kelas lainnya. Ketika kelas-kelas dalam dataset tidak seimbang, model yang dibangun cenderung cenderung memprediksi kelas mayoritas dengan lebih baik daripada kelas minoritas, sehingga mengurangi performa model untuk kelas minoritas.

RANDOM OVER SAMPLING

RandomOverSampling adalah salah satu teknik dalam pengolahan data yang digunakan untuk menangani ketidakseimbangan dalam dataset kelas-kelas yang berbeda. Ketidakseimbangan terjadi ketika ada perbedaan signifikan dalam jumlah sampel antara kelas mayoritas dan kelas minoritas dalam dataset. Dalam RandomOverSampling, sampel dari kelas minoritas secara acak ditingkatkan untuk mencapai keseimbangan dengan kelas mayoritas. Ini dapat dilakukan dengan menduplikasi data dari kelas minoritas secara acak hingga jumlah sampelnya sebanding dengan kelas mayoritas.

Berikut adalah langkah-langkah umum dalam melakukan RandomOverSampling:

1. Identifikasi Ketidakseimbangan

Periksa jumlah sampel untuk setiap kelas dalam dataset Anda untuk menentukan kelas mana yang merupakan kelas mayoritas dan kelas mana yang merupakan kelas minoritas.

2. Penerapan RandomOverSampling

Salah satu cara untuk menerapkan RandomOverSampling adalah dengan menggunakan pustaka atau modul dalam bahasa pemrograman seperti Python. Contohnya, modul imbalanced-learn menyediakan fungsi RandomOverSampler untuk melakukan oversampling secara acak.

```
# inisialisasi RandomOverSampler
balancer = RandomOverSampler(random_state=42) # menyimpan metode randomoversampler untuk menyei

# Menerapkan Random Over-Sampling untuk menyeimbangkan data
fitur_balance, target_balance = balancer.fit_resample(fitur_cleaned, target_cleaned) # balancin

# Menampilkan data yang telah diseimbangkan
print('Hasil Penyeimbangan:', Counter(target_balance))
```

```
Hasil Penyeimbangan: Counter({0: 1156, 1: 1156})
```

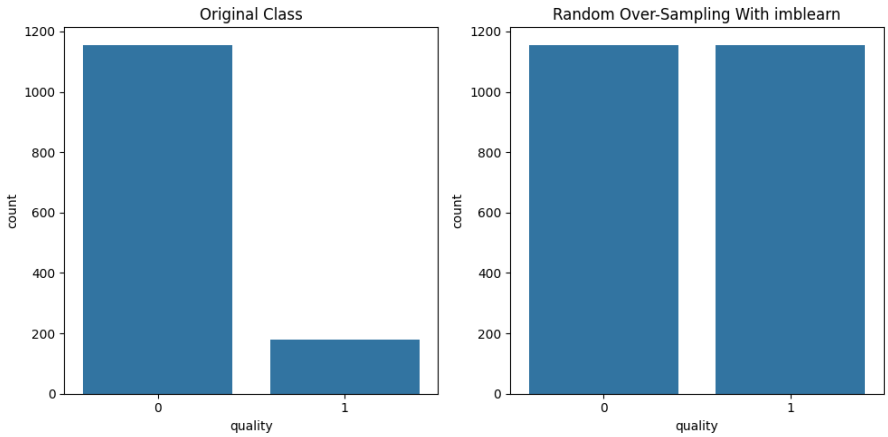
Setelah dilakukan balancing data maka data akan disimpan didalam sebuah file berekstensi pickle dengan tujuan jika ingin digunakan lagi untuk predksi maka data yang digunakan adalah data yang telah dinyatakan bersih dan memiliki kualitas data yang baik dikarenakn telah melewati pembersihand ata seperti penghapusan outlier dan penyeimbangan jumlah proporsi masing-masing kelas dalam data

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

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

```
# menghitung jumlah data setelah balancing data
count_data = data_clean.shape[0]
print("Jumlah Data Setelah Proses Balancing :", count_data)
```

```
Jumlah Data Setelah Proses Balancing : 2312
```

```
# Menampilkan kelas dalam data sebelum dilakukan balancing data dengan menggunakan plot
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
sns.countplot(data=pd.DataFrame({'quality': target_cleaned}), x='quality')
plt.title('Original Class')

# Menampilkan kelas dalam data sesudah dilakukan balancing data dengan menggunakan plot
plt.subplot(1, 2, 2)
sns.countplot(data=pd.DataFrame({'quality': target_balance}), x='quality')
plt.title('Random Over-Sampling With imblearn')

plt.tight_layout()
plt.show()
```



1.3.4 SPLIT DATA

Split Data adalah membagi data menjadi dset data uji dan set data training(latih) sehingga data fitur dan target masing-masing dibagi menjadi set data fitur training dan set data fitur testing serta set data target training dan set data target uji

1. set data pelatihan (training) Subset data yang digunakan untuk melatih model mesin. Model menggunakan data ini untuk belajar pola dan hubungan antara fitur (features) dan label (target).
2. set data uji(test) Subset yang tidak terlibat dalam pelatihan model atau penyetelan parameter. Data ini digunakan untuk menguji kinerja model yang telah dilatih dan divalidasi. Ini memberikan perkiraan seberapa baik model akan berperforma pada data yang belum pernah dilihat sebelumnya.

Pada Proyek ini data testing yang digunakan adalah 1/4 data (25%) dari jumlah keseluruhan data lalu untuk data tarining yang digunakan adalah sebanyak 3/4 (75%) data dar jumlah keseluruhan data dalam dataset fungsi train_test_split yang digunakan untuk membagi dataset menjadi set pelatihan (train set) dan set pengujian (test set). Ini membantu dalam mengevaluasi kinerja model pada data yang tidak digunakan selama pelatihan.

- fitur_balance: Matriks fitur dari dataset yang telah di-resampled dengan menggunakan Random Over-Sampling (X_ros).
- target_balance: Ini adalah vektor target yang sesuai dengan fitur_balance.
- test_size=0.25: Ini menentukan proporsi data yang akan dialokasikan untuk test set. Dalam hal ini, 25% dari total data akan digunakan sebagai test set.

- random_state=42: Ini digunakan untuk memastikan reproduktibilitas hasil. Jika kita memberikan nilai tertentu (dalam hal ini 42), maka setiap kali kita menjalankan fungsi ini, kita akan mendapatkan pembagian yang sama untuk set pelatihan dan set pengujian. Ini berguna ketika kita ingin hasil yang konsisten setiap kali kode dijalankan.

Hasil dari fungsi train_test_split adalah empat kumpulan data:

- fitur_train: Matriks fitur dari set pelatihan.
- fitur_test: Matriks fitur dari set pengujian.
- target_train: Vektor target dari set pelatihan.
- target_test: Vektor target dari set pengujian.

```
# membagi data dengan perbandingan data latih dan uji 4:1
fitur_train, fitur_test, target_train, target_test = train_test_split(fitur_balance, target_bal
```

1.3.5 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.

MINMAX

Pengertian

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.

Rumus Minmax Scaler

$$X' = \frac{X - min}{max - min}$$

Dimana :

- X adalah nilai asli dari suatu kolom/fitur
- min adalah nilai minimum dari suatu kolom/fitur dalam dataset
- max adalah nilai maximum dari suatu kolom/fitur dalam dataset
- X' adalah nilai X yang telah dinormalisasi.

Berikut Contoh Penggunaan minmax pada data, sebagai berikut:

1. Terdapat sebuah tabel dengan kolom X, seperti berikut:

X	X‘
2	0
4	0
3	0
10	0

2. Untuk melakukan normalisasi dengan Min-Max Scaling, maka perlu diidentifikasi nilai tertingggi dan terendah pada kolom. Deketahui :

- Nilai terendah pada kolom X (min) = 2
- Nilai tertinggi pada kolom X (max) = 10

3. Lalu untuk menentukan nilai X yang telah dinormalisasi maka dengan menghitung setiap baris data yang dimasukkan kedalam rumus Min-Max Scaling seperti berikut:

X	X‘
2	(2 - 2) / (10 - 2)
4	(4 - 2) / (10 - 2)
3	(3 - 2) / (10 - 2)
10	(10 - 2) / (10 - 2)

4. Sehingga nilai X' hasil normalisasi dapat diketahui seperti berikut :

X	X‘
2	0
4	0.25
3	0.125
10	1

```
mm = MinMaxScaler() # meyimpan normalisasi MinMax Scaler() dalam variabel
mm.fit(fitur_train)
minmax_training = mm.fit_transform(fitur_train) # melakukan normalisasi pada kolom fitur latih
minmax_testing = mm.transform(fitur_test) # melakukan normalisasi pada kolom fitur uji
with open('minmax.pkl', 'wb') as file:
    pickle.dump(mm, file)
```

1.3.6 RANKING SETIAP FITUR DALAM DATA

Untuk memilih fitur terbaik dapat dengan me-renking fitur-fitur didalm dataset dengan mencari nilai mutual information setiap fitur agar dapat dibandingkan diurutkan dengan fitur lainnya. Padaproyek ini digunakan metode SelectKBest dengan mencari nilai mutual information dari setiap fitur.

1.3.6.1 MUTUAL INFORMATION

Mutual information (MI) adalah metrik yang berguna dalam pemilihan fitur karena mengukur seberapa banyak informasi yang saling terkait antara fitur (variabel independen) dengan variabel target (variabel dependen). Dalam konteks pemilihan fitur, kita ingin mempertahankan fitur-fitur yang memiliki hubungan yang kuat atau tinggi dalam menjelaskan variabel target.

Rumus Mutual Information (MI) between X and Y:

$$\mathrm{MI}(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x) \cdot p(y)} \right)$$

Dimana:

- MI(X;Y) adalah mutual information antara variabel X dan Y.
- p(x,y) adalah probabilitas bersama dari X=x dan Y=y.
- p(x) adalah probabilitas margina X=x.
- p(y) adalah probabilitas margina Y=y.

Dari tahap ini dihasilkan rangking dari setiap kolom atau fitur sehingga dapat mengetahui fitur-fitur yang memiliki ciri khas penting untuk data red wine itu sendiri. Pada tahap skenario percobaan nantinya akan dipilih fitur-fitur yang memiliki nilai mutual information tertinggi. Untuk memilih fitur-fitur yang

terbaik dapat dimulai dengan menghapus fitur yang memiliki nilai mutual information paling rendah lalu skenario bergeser ke kiri dimana fitur yang dihapus yaitu fitur dengan mutual information terendah nomer 2 dan paling rendah. pemilihan fitur terus dilakukan pada skenario percobaan hingga mencapai akurasi tertinggi.

Contoh Kasus untuk mengurutkan kolom dengan mutual information paling tinggi hingga terendah, di berikan tabel berikut :

ID	Variabel_1	Variabel_2	Variabel_Target
1	2	3	b
2	1	2	a
3	3	1	a
4	2	2	b
5	1	3	a

- Langkah 1 : Hitung distribusi probabilitas

Untuk Variabel_Target: > - P(Target=a) = 3/5 > - P(Target=b) = 2/5

Untuk Variabel_1: > - P(Variabel_1=1) = 2/5 > - P(Variabel_1=2) = 2/5 > - P(Variabel_1=3) = 1/5

Untuk Variabel_2:

- P(Variabel_2=1) = 1/5
- P(Variabel_2=2) = 2/5
- P(Variabel_2=3) = 2/5

- Langkah 2 : Hitung Entropi

Entropi(variabel_Target) = - (3/5) * log2(3/5) - (2/5) * log2(2/5) 0.971

- Entropi(Variabel_1) = - (2/5) * log2(2/5) - (2/5) * log2(2/5) - (1/5) * log2(1/5) 1.571

- Entropi(Variabel_2) = - (1/5) * log2(1/5) - (2/5) * log2(2/5) - (2/5) * log2(2/5) 1.571

- Langkah 3 : Hitung Conditional Entropi

Untuk setiap nilai Variabel_1, hitung Entropi(Target|Variabel_1).

Variabel_1

Conditional_Entropy(Target | Variabel_1) = P(Variabel_1=1) * Entropi(Target|Variabel_1=1) + P(Variabel_1=2) * Entropi(Target|Variabel_1=2) + P(Variabel_1=3) * Entropi(Target|Variabel_1=3)

- Entropi(Target|Variabel_1=1) = - (1/2) * log2(1/2) - (1/2) * log2(1/2) = 1.0

- Entropi(Target|Variabel_1=2) = - (1/1) * log2(1/1) - (1/1) * log2(1/1) = 0.0

- Entropi(Target|Variabel_1=3) = - (1/1) * log2(1/1) = 0.0

Conditional_Entropy(Target | Variabel_1) = (2/5)×1.0+(2/5)×0.0+(1/5)×0.0 = 0.4

Variabel_2

Conditional_Entropy(Target | Variabel_2) = P(Variabel_2=1) * Entropi(Target|Variabel_2=1) + P(Variabel_2=2) * Entropi(Target|Variabel_2=2) + P(Variabel_2=3) * Entropi(Target|Variabel_2=3)

- Entropi(Target | Variabel_2=1) = - (1) * log2(1) = 0

- Entropi(Target | Variabel_2=2) = - (0.5) * log2(0.5) - (0.5) * log2(0.5) = 1.0

- Entropi(Target | Variabel_2=3) = - (0.5) * log2(0.5) - (0.5) * log2(0.5) = 1.0

Conditional_Entropy(Target | Variabel_2) = (1/5) * 0 + (2/5) * 1.0 + (2/5) * 1.0 = 0.8

- Langkah 4 : Hitung Mutual Information

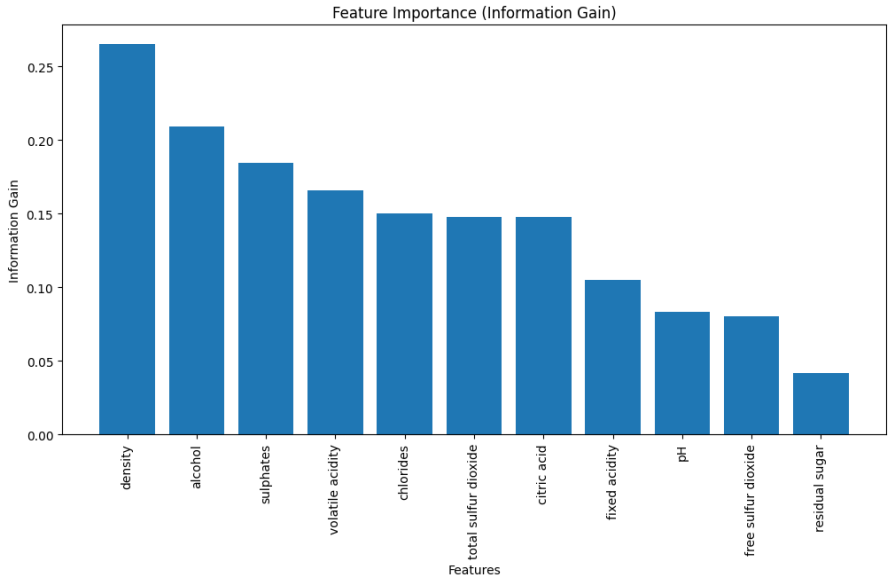
- Mutual_Information(Variabel_1;Target) = 0.971 - 0.4 = 0.571

- Mutual_Information(Variabel_2;Target) = 0.971 - 0.8 = 0.171

Dengan begitu variabel yang lebih berpengaruh adalah Variabel_1 dari Variabel_2

```
# menyimpan metode selectkbest dengan menggunakan mutual information lalu fitur yang dideleksi
selector = SelectKBest(score_func=mutual_info_classif, k='all')
# untuk mengubah dataset fitur menjadi X_new, yang hanya berisi fitur terbaik yang dipilih oleh
X_new = selector.fit_transform(fitur_balance,target_balance)
# Mengambil skor dari fitur-fitur dalam dataset. Skor-skor ini merepresentasikan informasi gain
scores = selector.scores_

# Menyimpan skor fitur dan nama fitur dalam variabel terpisah untuk digunakan dalam visualisasi
feature_scores = selector.scores_
feature_names = ['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol']

# Mengurutkan indeks fitur berdasarkan skor fitur dari yang tertinggi ke terendah.
sorted_indices = feature_scores.argsort()[::-1]

# Mengurutkan skor fitur dan nama fitur sesuai dengan indeks yang telah diurutkan sebelumnya.
sorted_scores = feature_scores[sorted_indices]
sorted_feature_names = [feature_names[i] for i in sorted_indices]

# Plot the feature importances
# digunakan untuk membuat diagram batang yang merepresentasikan skor fitur.
plt.figure(figsize=(12, 6))
plt.bar(range(len(sorted_feature_names)), sorted_scores)
# digunakan untuk menambahkan label pada sumbu x (fitur) dengan rotasi 90 derajat agar lebih mu
plt.xticks(range(len(sorted_feature_names)), sorted_feature_names, rotation=90)
```

```
# Label sumbu x adalah nama fitur, sedangkan label sumbu y adalah skor fitur (information gain)
plt.xlabel('Features')
plt.ylabel('Information Gain')
plt.title('Feature Importance (Information Gain)') # memberikan judul plot.
plt.show() # menampilkan plot.
```



Setelah itu dilakukan skenario percobaan , diketahui bahwa fitur yang terbaik untuk digunakan dalam klasifikasi data red wine, diantaranya

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol

1.4 MODELING

Tahap modeling dalam analisis data adalah tahap dimana digunakannya berbagai teknik dan algoritma untuk membangun model prediktif atau deskriptif dari data yang telah siapkan. Ini merupakan salah satu langkah penting dalam proses analisis data dan umumnya terdiri dari beberapa langkah, tergantung pada jenis analisis yang dilakukan.

1.4.1 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.

Prinsip kerja utama algoritma Random Forest

1. Pemilihan Sampel Acak (Random Sampling):

Dari dataset yang ada, Random Forest melakukan sampling acak dengan penggantian (bootstrap sampling) untuk membuat dataset yang lebih kecil namun representatif. Hal ini membantu dalam membangun berbagai pohon keputusan yang berbeda.

2. Pembangunan Pohon Keputusan (Decision Tree):

Setiap pohon keputusan dibangun menggunakan subset dari data yang diambil secara acak pada langkah pertama. Pohon-pohon ini dibuat dengan membagi data berulang-ulang berdasarkan fitur-fitur yang tersedia, untuk membuat keputusan yang optimal pada setiap simpulnya.

3. Voting atau Penggabungan (Voting or Aggregation):

Setelah semua pohon keputusan selesai dibuat, hasil prediksi dari setiap pohon dikumpulkan. Pada tahap ini, jika masalahnya adalah klasifikasi, hasilnya diambil dengan cara mayoritas (melalui voting), sedangkan untuk regresi, hasilnya bisa diambil dengan rata-rata dari prediksi semua pohon.

4. Pengukuran Keakuratan (Accuracy Measurement):

Untuk menentukan hasil akhir, model menghasilkan prediksi berdasarkan mayoritas suara dari semua pohon keputusan (untuk klasifikasi) atau nilai rata-rata (untuk regresi). Keakuratan model kemudian diukur dengan membandingkan prediksi tersebut dengan nilai sebenarnya dari data yang tidak terlibat dalam pelatihan.

Parameter Algoritma Random Forest

Jika ingin mengimplementasikan algoritma Random Forest dalam kasus klasifikasi, maka terdapat parameter-parameter yang perlu diperhatikan diantaranya : Parameter-parameter yang perlu diperhatikan saat membangun algoritma Random Forest adalah sebagai berikut:

1. Jumlah Pohon (n_estimators)

Ini adalah jumlah pohon keputusan yang ingin Anda bangun dalam Random Forest. Semakin banyak pohon, biasanya hasilnya akan lebih baik, namun akan memakan lebih banyak waktu komputasi.

2. Kedalaman Pohon (max_depth)

Menentukan kedalaman maksimum dari setiap pohon keputusan dalam ensemble. Kedalaman yang terlalu dalam dapat menyebabkan overfitting.

3. Min_samples_split dan min_samples_leaf

Menentukan jumlah sampel minimum yang diperlukan untuk membagi node internal atau menjadi daun (leaf node) dalam pohon.

Untuk mencari parameter terbaik dalam klasifikasi dengan menggunakan

metode Random Forest dapat menggunakan metode bantuan untuk mencari parameter terbaik yang akan digunakan dalam kasus klasifikasi data red wine. Pada proyek ini digunakan metode GridSearch CV untuk menemukan parameter yang optimal dan terbaik untuk metode Random Forest seperti mencari jumlah pohon,kedalaman pohon, minmal pembagian sampel dan pembagian daun.

1.4.2 GridSearchCV

---

adalah metode yang digunakan untuk menemukan parameter yang optimal bagi suatu model dalam machine learning dengan menggunakan validasi silang (cross-validation).GridSearchCV bekerja dengan melakukan pencarian parameter terbaik yang diberikan dalam rentang nilai yang telah ditentukan sebelumnya. Ini dilakukan dengan cara menguji semua kombinasi yang mungkin dari parameter-parameter yang diinginkan dan mengevaluasi performa model menggunakan teknik validasi silang untuk setiap kombinasi parameter.

Langkah-langkah umum yang dilakukan oleh GridSearchCV adalah sebagai berikut:

- Spesifikasi Model dan Parameter: Tentukan model yang ingin Anda gunakan serta parameter apa yang ingin Anda uji. Misalnya, jika Anda menggunakan model Random Forest, parameter seperti jumlah pohon, kedalaman maksimum, dan fitur yang harus dipertimbangkan.
- Definisikan Grid Parameter: Buat daftar parameter yang ingin diuji dengan rentang nilai-nilai yang berbeda untuk setiap parameter. Ini membentuk sebuah grid atau kisi dari kombinasi parameter-parameter yang mungkin.
- Cross-Validation: Gunakan teknik validasi silang (biasanya K-Fold Cross-Validation) untuk membagi data menjadi subset pelatihan dan validasi. Setiap kombinasi parameter akan dievaluasi pada setiap lipatan untuk mendapatkan skor evaluasi.
- Pemilihan Parameter Terbaik: Setelah proses validasi selesai, GridSearchCV akan menentukan kombinasi parameter terbaik yang memberikan performa terbaik sesuai dengan metrik evaluasi yang telah ditentukan sebelumnya (misalnya, akurasi, F1-score, dll.).
- Penggunaan Model Terbaik: Setelah menemukan parameter terbaik, Anda dapat menggunakan model dengan parameter tersebut untuk membuat prediksi pada data baru

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 11 fitur

```
# Define the parameter grid for Random Forest
param_grid = {
    'n_estimators': [100, 200, 300],  # You can adjust the number of trees
    'max_depth': [None, 10, 20, 30],  # You can adjust the maximum depth of each tree
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Create a Random Forest model
random_forest = RandomForestClassifier()

# MINMAX
grid_search2 = GridSearchCV(estimator=random_forest, param_grid=param_grid, cv=5, scoring='accu
grid_search2.fit(minmax_training, target_train)
print("Best Parameters MINMAX:", grid_search2.best_params_)
best_n_estimators_minmax = grid_search2.best_params_['n_estimators']
best_max_depth_minmax = grid_search2.best_params_['max_depth']
best_min_samples_split_minmax = grid_search2.best_params_['min_samples_split']
best_min_samples_leaf_minmax = grid_search2.best_params_['min_samples_leaf']

print("Parameter dalam metode yang digunakan, sebagai berikut \n")
print("jumlah best estimator  :",best_n_estimators_minmax,"\n")
print("best maksimal kedalaman  :",best_max_depth_minmax,"\n")
print("best minimal pembagian sampel  :",best_min_samples_split_minmax,"\n")
print("best minimal sampel daun  :",best_min_samples_leaf_minmax,"\n")
```

```
Best Parameters MINMAX: {'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimator
Parameter dalam metode yang digunakan, sebagai berikut

jumlah best estimator  : 300

best maksimal kedalaman  : 30

best minimal pembagian sampel  : 2

best minimal sampel daun  : 1
```

```
# MINMAX
#membuat model klasifikasi Random Forest dari parameter terbaik yang telah diketahui di proses
model_rf_minmax = RandomForestClassifier( max_depth= best_max_depth_minmax,min_samples_leaf= bes
```

```
# Melatih model RandomForestClassifier dengan dataset latihan (minmax_training) yang telah dino
model_rf_minmax.fit(minmax_training, target_train)
# Menggunakan model yang telah dilatih untuk melakukan prediksi pada dataset pengujian (minmax_
y_pred_minmax = model_rf_minmax.predict(minmax_testing)
# Menghitung akurasi dari prediksi yang dilakukan oleh model RandomForestClassifier
accuracy_rf_minmax = accuracy_score(target_test, y_pred_minmax)

print("AKURASI RANDOM FOREST")
print("AKURASI minmax :",accuracy_rf_minmax)
```

```
AKURASI RANDOM FOREST
AKURASI minmax : 0.9377162629757786
```

1.4.3 SIMPAN MODEL KLASIFIKASI

Untuk keperluan Prediksi nanti pada tahap Deployment, maka model terbaik yang telah diterapkan saat ini disimpan untuk kasus prediksi kualitas red wine kedepannya sehingga tidak perlu lagi mencari-cari model terbaik untuk melakukan prediksi pada data uji baru nantinya

```
with open('best_model_rf_minmax.pkl', 'wb') as model_file:
    pickle.dump(model_rf_minmax,model_file)
```

1.5 EVALUASI MODEL

Tahap evaluasi model adalah langkah penting dalam proses pembangunan model prediktif atau deskriptif untuk memastikan bahwa model yang dibuat dapat memberikan hasil yang berguna dan akurat. Evaluasi model dilakukan setelah proses pelatihan model untuk memahami seberapa baik model tersebut dapat melakukan prediksi atau generalisasi pada data yang belum pernah dilihat sebelumnya.

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

HASIL TABEL KEBINGUNGAN DAN MATRIKS EVALUASI

Pada tahap ini dilakukan evaluasi terhadap model terbaik yang telah dibuat

pada tahap modelling sebelumnya, sehingga menghasilkan beberapa evaluasi yang di representasikan dalam bentuk Tabel Kebingungan maupun Matriks Evaluasi:

Tabel Kebingungan

Didalam Tabel Kebingungan dihasilkan nilai dari TP,FP,FN dan TN seperti berikut:

- True Positive (TP): 342
- True Negative (TN): 317
- False Positive (FP): 17
- False Negative (FN): 3

Matriks Evaluasi

Didalam Matriks Evaluasi dapat diketahui nilai-nilai dari beberapa bagian di matriks Evaluasi diantaranya :

- Akurasi Model : 0.97
- Score ROC-AUC : 0.9702
- Nilai Presisi, Recall, F1-Score dan Support setiap kelas yang terdapat didalam data:

	Pesisi	Recall	f1-score
Kualitas Baik	0.95	0.95	0.97
Kualitas Buruk	0.99	0.99	0.97

HASIL INTERPRETASI ROC-AUC

Dari kurva ROC yang dihasilkan dapat disimpulkan bahwasannya nilai AUC lebih mendekati nilai 1, hal ini menandakan bahwasannya model yang diterapkan mampu membedakan antara kelas porsitif dan kelas negatif dengan baik.

1.5.1 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.

METRIKS 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 :

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi \times Recall}$$

- Specificity (Specificity atau True Negative Rate): Proporsi dari kelas negatif yang diprediksi dengan benar oleh model.

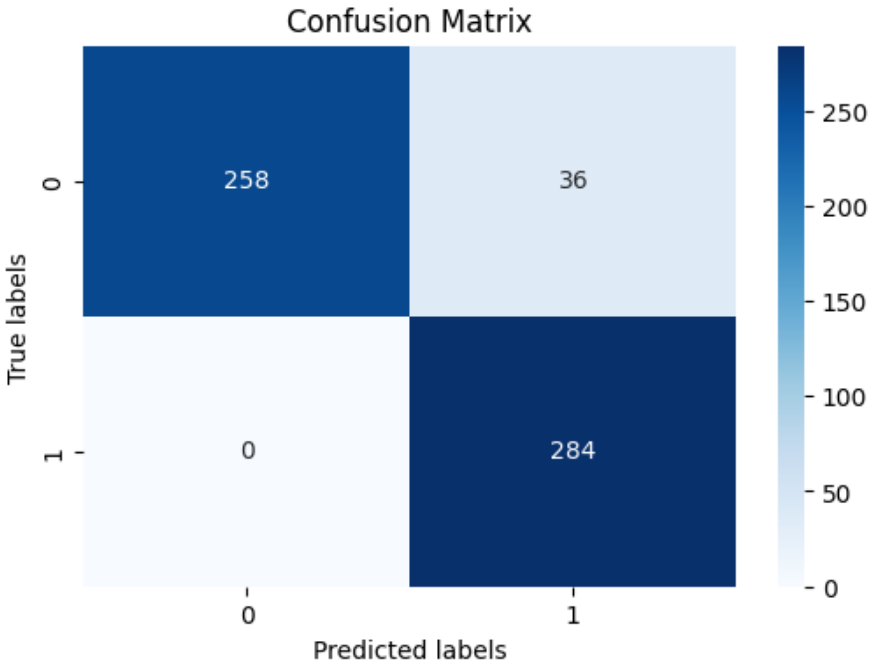
Rumus Specificity :

$$Specificity = \frac{TN}{TN + FP}$$

```
# Evaluasi model dengan data uji MINMAX
print("\nEVALUASI MODEL DENGAN DATA UJI ZSCORE")
print("Confusion Matrix ZSCORE:")
# menghiung confusion matriks antara target aktual hasil prediksi
conf_matrix = confusion_matrix(target_test, y_pred_minmax)
print(conf_matrix)

# Mendapatkan nilai TP, TN, FP, FN dari confusion matrix
TN = conf_matrix[0, 0] # mengambil nilai TN dalam matriks
FP = conf_matrix[0, 1] # mengambil nilai FP dalam matriks
FN = conf_matrix[1, 0] # mengambil nilai FN dalam matriks
TP = conf_matrix[1, 1] # mengambil nilai TP dalam matriks

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

print("\nClassification Report ZSCORE:")
print(classification_report(target_test, y_pred_minmax)) # mencetak laporan klasifikasi yang le
# Ini akan mencetak skor ROC-AUC (Area Under the Receiver Operating Characteristic Curve) dari
print("ROC-AUC Score MINMAX:", roc_auc_score(target_test, y_pred_minmax))
```

```
EVALUASI MODEL DENGAN DATA UJI ZSCORE
Confusion Matrix ZSCORE:
[[258  36]
 [  0 284]]

True Positive (TP): 284
True Negative (TN): 258
False Positive (FP): 36
False Negative (FN): 0

Classification Report ZSCORE:
              precision    recall  f1-score   support

           0       1.00      0.88      0.93       294
           1       0.89      1.00      0.94       284

    accuracy                           0.94       578
   macro avg       0.94      0.94      0.94       578
weighted avg       0.94      0.94      0.94       578

ROC-AUC Score MINMAX: 0.9387755102040816
```

```
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import seaborn as sns

conf_matrix = confusion_matrix(target_test, y_pred_minmax)

plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='g')
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.show()
```



1.5.2 GRAFIK ROC-AUC

Metrik evaluasi ROC (Receiver Operating Characteristic) dan AUC (Area Under the ROC Curve) adalah alat evaluasi yang digunakan untuk mengukur kinerja model klasifikasi, terutama ketika model harus mengklasifikasikan antara dua kelas.

Receiver Operating Characteristic (ROC) Curve

ROC Curve adalah adalah kurva grafik yang menampilkan kinerja model klasifikasi pada berbagai tingkat cutoff (threshold) untuk membedakan antara kelas positif dan negatif. Didalam ROC kurva dapat diketahui sensitivity (True Positive Rate) dan False Positive Rate (1-Specificity), untuk menunjukkan seberapa baik model klasifikasi sehingga dapat membedakan antara kelas positif dan negatif.

Area Under the ROC Curve (AUC-ROC)

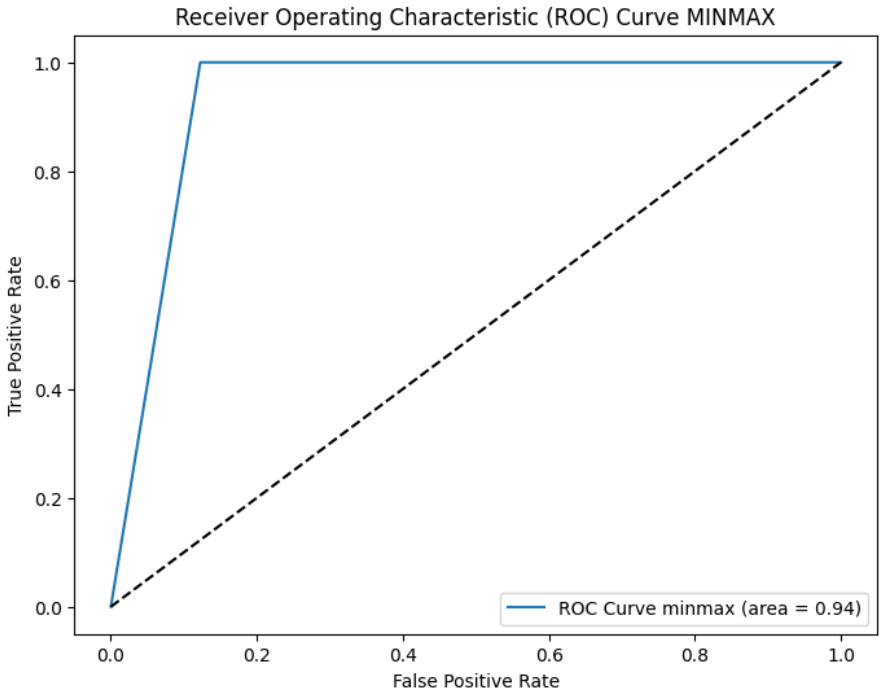
AUC-ROC adalah ukuran dari luas area di bawah kurva ROC.

- Interpretasi :

Nilai AUC berkisar antara 0 hingga 1. Semakin dekat nilainya ke 1, semakin

baik model dalam membedakan antara kelas positif dan negatif. Jika nilainya 0.5, itu menunjukkan klasifikasi acak.

```
# Kurva ROC-AUC untuk model dengan data uji minmax
# Menghitung False Positive Rate (FPR), True Positive Rate (TPR), dan threshold untuk kurva ROC
# target_test adalah label yang sebenarnya dari data pengujian, sedangkan y_pred_minmax adalah
fpr_minmax, tpr_minmax, thresholds_minmax = roc_curve(target_test, y_pred_minmax)
# Membuat figur (plot) dengan ukuran 8x6 inci.
plt.figure(figsize=(8, 6))
# Membuat plot untuk kurva ROC dengan sumbu x sebagai FPR dan sumbu y sebagai TPR.
plt.plot(fpr_minmax, tpr_minmax, label='ROC Curve minmax (area = %0.2f)' % roc_auc_score(target_
# Membuat garis putus-putus yang mewakili kurva ROC jika model tidak memiliki kemampuan predikt
plt.plot([0, 1], [0, 1], 'k--')
# Menambahkan label sumbu x dan y pada plot untuk menunjukkan apa yang direpresentasikan oleh su
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Memberikan judul pada plot, menunjukkan bahwa ini adalah kurva ROC untuk model dengan data pe
plt.title('Receiver Operating Characteristic (ROC) Curve MINMAX')
# Menampilkan legenda di pojok kanan bawah plot, menunjukkan informasi tambahan tentang kurva y
plt.legend(loc='lower right')
plt.show() #  Menampilkan plot kurva ROC yang telah dibuat.
```



1.6 DEPLOYMENT

Tahap deployment dalam konteks analisis data atau pengembangan model mengacu pada proses menerapkan atau menempatkan model yang telah dibuat ke dalam lingkungan produksi atau ke dalam penggunaan praktis di dunia nyata.

Penerapan model akan diimplementasikan untuk melakukan prediksi kualitas red wine. Setelah ini tahap berpindah ke aplikasi streamlit.py

–>

2

Summary

In summary, this book has no content whatsoever.

References