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 KLASIFIKASI DATA PROYEK SAINS DATA

Nama : Enjel Putri Sabrina
NIM : 210411100126
Kelas : B

1.1 – BUSSINESS UNDERSTANDING –

Identifikasi kasus :

Liver merupakan salah satu penyakit kronis yang paling umum di Timur Laut Andhra Pradesh, India. Kematian akibat liver ini terus meningkat, seiring dengan meningkatnya angka konsumsi alkohol, infeksi hepatitis kronis, dan penyakit hati terkait obesitas

Tujuan :

Membangun model klasifikasi liver sebagai upaya deteksi terhadap penyakit liver dengan menggunakan model Random Forest. Dengan melakukan analisa pada data liver, model ini dapat membangun prediksi yang akurat sehingga memungkinkan tenaga kesehatan untuk mengidentifikasi pasien liver.

1.2 – DATA UNDERSTANDING –

Tahapan Pada Data Understanding merupakan tahapan dimana kita perlu memahami data yang akan diolah. Adapun hal - hal yang perlu dilakukan nantinya untuk memahami dataset ini, yakni

1. Tentang data, mencakup :
    - Pengumpulan dataset
    - Pengenalan singkat mengenai data yang akan diolah

2. Mendeskripsikan data, mencakup :
   - analisa tipe data
   - deskripsi fitur
3. Eksplorasi data, mencakup :
   - Visualisasi data
4. Identifikasi kualitas data :
   - Identifikasi missing value setiap fitur atau kolom
   - Identifikasi data duplikat
   - Identifikasi outlier (data aneh)
   - Identifikasi jumlah data (proporsi data perkelas yang digunakan untuk mengetahui balancing dataset atau keseimbangan data per kelas)

1.2.1 1. Tentang Data

1.2.2 - Teknik Pengumpulan Data

Dalam proyek ini menggunakan kumpulan data yang tersedia di kaggle yang berupa csv. Fitur-fitur yang ada berupa catatan pasien liver dan pasien non liver yang dikumpulkan dari India dengan jumlah dataset sebanya 583 data.

```
import pandas as pd

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

	Selector	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	A/G Ratio
0	1	65	Female	0.7	0.1	187.0	16.0	18.0	6.8	3.3	0.9
1	1	62	Male	10.9	5.5	699.0	64.0	100.0	7.5	3.2	0.7
2	1	62	Male	7.3	4.1	490.0	60.0	68.0	7.0	3.3	0.9
3	1	58	Male	1.0	0.4	182.0	14.0	20.0	6.8	3.4	1.0
4	1	72	Male	3.9	2.0	195.0	27.0	59.0	7.3	2.4	0.4

```
# rincian dataset (jumlah data dan kolom)

print("Jumlah data : ", data.shape[0])
print("Jumlah kolom : ", data.shape[1])
```

```
Jumlah data :  583
Jumlah kolom :  11
```

Dataset ini sebanyak 583 data dengan rincian pelabelan sebagai berikut ini :

- Pasien liver (1) = 416 data
- Pasien Nonliver (2) = 167 data *Perbedaan yang signifikan pada jumlah data antar label sehingga perlu diseimbangkan terlebih dahulu*

Untuk mendeteksi penyakit liver di India, dapat dilihat dari ciri-ciri sebagai berikut :

1. **Age**: Mewakili usia pasien, mengindikasikan faktor demografis yang mungkin relevan dengan kesehatan hati.
2. **Gender**: Menunjukkan jenis kelamin pasien, memberikan informasi tentang distribusi penyakit hati antar jenis kelamin.
3. **TB (Total Bilirubin)**: Mengukur total bilirubin dalam darah, yang dapat menjadi indikator fungsi dan kesehatan hati. Bilirubin merupakan pigmen kuning yang dihasilkan sebagai hasil pemecahan hemoglobin dari sel darah merah yang tua di hati.
4. **DB (Direct Bilirubin)**: Secara khusus mengukur bilirubin langsung atau tidak terkonjugasi, memberikan informasi lebih rinci tentang tingkat bilirubin terkait dengan kesehatan hati.
5. **Alkphos (Alkaline Phosphatase)**: Mengukur level alkaline phosphatase, enzim yang dapat terkait dengan kesehatan hati dan tulang. Tingkat yang meningkat dapat menunjukkan masalah hati.
6. **SGPT (Alamine Aminotransferase)**: Mengukur level SGPT, sebuah enzim yang dapat menunjukkan kerusakan atau peradangan pada hati. Tingkat SGPT yang tinggi dalam tes darah dapat menjadi indikasi kerusakan hati, seperti hepatitis atau kerusakan hati akibat alkohol atau obat-obatan.
7. **SGOT (Aspartate Aminotransferase)**: Mengukur level SGOT, enzim yang terdapat di berbagai jaringan, dengan level yang tinggi mungkin mengindikasikan masalah pada hati atau jantung.
8. **TP (Total Protein)**: Menunjukkan total protein dalam darah, yang dapat terkait dengan fungsi hati dan kesehatan secara umum.
9. **Alb (Albumin)**: Mengukur tingkat albumin dalam darah, protein penting yang diproduksi oleh hati, dan dapat menjadi indikator fungsi hati. Penurunan kadar albumin dalam darah dapat terjadi pada pasien dengan penyakit hati yang merusak kemampuan hati untuk memproduksi protein ini.
10. **A/G Ratio (Rasio Albumin/Globulin)**: Mewakili rasio albumin terhadap globulin dalam darah, memberikan wawasan tentang fungsi hati dan ginjal.

11. **Selector**: kolom yang digunakan untuk pelabelan diagnosa liver dan non liver

1.2.3 2. Mendeskripsikan Data

```
data.columns
```

```
Index(['Selector', 'Age', 'Gender', 'TB', 'DB', 'Alkphos', 'Sgpt', 'Sgot',
       'TP', 'ALB', 'A/G Ratio'],
      dtype='object')
```

1.2.4 - Analisa Tipe Data

Dalam analisa data ini, terdapat tipe data yang ditemukan yaitu tipe data nominal.

- Tipe nominal
  - memiliki value 1 yang melambangkan ya dan 0 yang melambangkan tidak. > Pada data ini mencakup fitur *'Selector'*
  - memiliki value perempuan dan laki laki. > yakni pada fitur *'Gender'*
  - mencakup tipe data numeric. > yakni pada fitur *'Age', 'TB', 'DB', 'Alkphos', 'SGPT', 'SGOT', 'TP', 'ALB', 'A/G Ratio'*

1.2.5 - Deskripsi Fitur

Terdapat beberap fitur yang ada pada dataset ini yaitu sebagai berikut:

1. **Age**: Mewakili usia pasien, mengindikasikan faktor demografis yang mungkin relevan dengan kesehatan hati.
2. **Gender**: Menunjukkan jenis kelamin pasien, memberikan informasi tentang distribusi penyakit hati antar jenis kelamin.
3. **TB (Total Bilirubin)**: Mengukur total bilirubin dalam darah, yang dapat menjadi indikator fungsi dan kesehatan hati.Bilirubin merupakan pigmen kuning yang dihasilkan sebagai hasil pemecahan hemoglobin dari sel darah merah yang tua di hati.
4. **DB (Direct Bilirubin)**: Secara khusus mengukur bilirubin langsung atau tidak terkonjugasi, memberikan informasi lebih rinci tentang tingkat bilirubin terkait dengan kesehatan hati.
5. **Alkphos (Alkaline Phosphatase)**: Mengukur level alkaline phosphatase, enzim yang dapat terkait dengan kesehatan hati dan tulang. Tingkat yang meningkat dapat menunjukkan masalah hati.

6. **SGPT (Alamine Aminotransferase)**: Mengukur level SGPT, sebuah enzim yang dapat menunjukkan kerusakan atau peradangan pada hati. Tingkat SGPT yang tinggi dalam tes darah dapat menjadi indikasi kerusakan hati, seperti hepatitis atau kerusakan hati akibat alkohol atau obat-obatan.
7. **SGOT (Aspartate Aminotransferase)**: Mengukur level SGOT, enzim yang terdapat di berbagai jaringan, dengan level yang tinggi mungkin mengindikasikan masalah pada hati atau jantung.
8. **TP (Total Protein)**: Menunjukkan total protein dalam darah, yang dapat terkait dengan fungsi hati dan kesehatan secara umum.
9. **Alb (Albumin)**: Mengukur tingkat albumin dalam darah, protein penting yang diproduksi oleh hati, dan dapat menjadi indikator fungsi hati. Penurunan kadar albumin dalam darah dapat terjadi pada pasien dengan penyakit hati yang merusak kemampuan hati untuk memproduksi protein ini.
10. **A/G Ratio (Rasio Albumin/Globulin)**: Mewakili rasio albumin terhadap globulin dalam darah, memberikan wawasan tentang fungsi hati dan ginjal.
11. **Selector**: kolom yang digunakan untuk pelabelan diagnosa liver dan non liver.

- 1: Liver
- 2: NonLiver

1.2.6 3. Eksplorasi Data

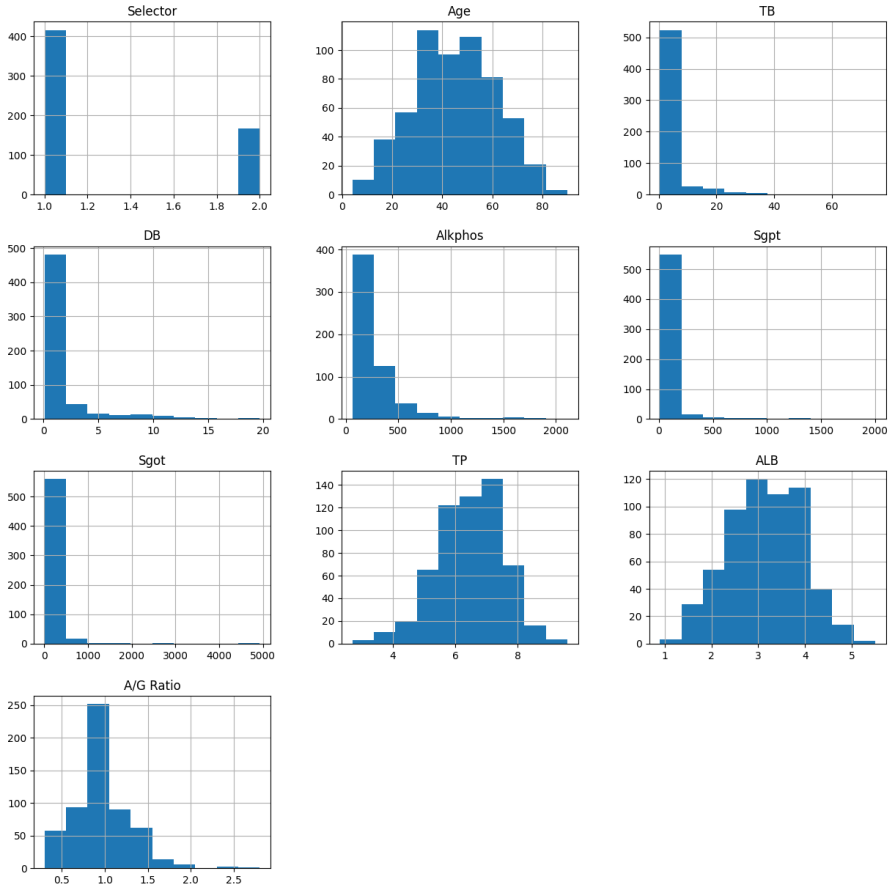
1.2.7 - Visualisasi Data

Visualisasi data dilakukan untuk memudahkan dalam memahami data dengan memperoleh informasi sebaran nilai dari dataset.

```
import matplotlib.pyplot as plt

data.hist(figsize=(15,15))
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



1.2.8 4. Identifikasi Kualitas Data

1.2.9 - Identifikasi Missing value

Missing value yaitu hilangnya satu atau beberapa data dalam suatu atribut. Penyebab dari adanya missing value yaitu kesalahan pada saat memasukkan data, variabel yang tidak relevan, dan ketidakmampuan merekam informasi, serta penghapusan.

Penanganan Missing value : Jika atribut tersebut memiliki banyak missing value, maka atribut tersebut perlu dihapus dari dataset. Namun jika hanya terdapat beberapa data yang missing value bisa dilakukan drop dari baris yang memiliki missing value atau mengisinya dengan rata-rata nilai pada atribut yang bersangkutan.

Contoh adanya missing value:

fitur1

fitur2

22

20

4

14

5

1

Dari tabel di atas, fitur 1 memiliki banyak sekali missing value sehingga atribut fitur 1 perlu dihilangkan.

Untuk mengidentifikasi ada tidaknya missing value, menggunakan fungsi *isna( )* yang digunakan untuk mengidentifikasi apakah setiap fitur dalam dataset memiliki nilai yang hilang (NaN atau NULL). Selain itu, juga menggunakan fungsi *any( )* yang digunakan untuk menentukan apakah ada setidaknya satu nilai yang hilang dalam setiap fitur.

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

# menampilkan hasil
print("Apakah ada nilai yang hilang dalam setiap kolom:")
print(missing_values)
```

```
Apakah ada nilai yang hilang dalam setiap kolom:
Selector     False
Age          False
Gender       False
TB           False
DB           False
Alkphos      False
Sgpt         False
Sgot         False
TP           False
ALB          False
A/G Ratio     True
dtype: bool
```

Note : terdapat missing value pada fitur *A/G Ratio*

Jumlah missing value pada data

```
missing_values = data.isnull().sum()
print("Jumlah Missing Values dalam Setiap Kolom:")
print(missing_values)
```

```
Jumlah Missing Values dalam Setiap Kolom:
Selector     0
Age          0
Gender       0
TB           0
DB           0
Alkphos      0
Sgpt         0
Sgot         0
TP           0
ALB          0
A/G Ratio    4
dtype: int64
```

1.2.10 - Identifikasi Duplikat data

Untuk mengidentifikasi baris-baris dalam dataset yang merupakan duplikat dari baris-baris sebelumnya yaitu dengan menggunakan fungsi *dupicate( )*

```
jumlah_duplikat = data.duplicated().sum()

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

```
Jumlah data yang duplikat: 13
```

Note : terdapat beberapa baris data yang sama, sehingga data tersebut harus dihilangkan untuk menghindari adanya data yang redundan

1.2.11 - Identifikasi Outlier

Outlier merupakan nilai yang signifikan atau ekstrem yang berbeda secara signifikan dari sebagian besar dalam satu dataset.

Contoh :

Nama

Nilai Ujian

siswa 1

79

siswa 2

9000

siswa 3

87

siswa 4

78

siswa 5

84

Dari data tersebut ditemukan satu outlier yakni pada data siswa 2 yang mana, nilai 9000 merupakan contoh outlier karena nilainya jauh lebih besar daripada nilai-nilai ujian lainnya.

Untuk mengidentifikasi outlier dengan mengimport kelas *LocalOutlierfactor* yang menyediakan implementasi algoritma LOF untuk deteksi outlier.Pada model LOF menggunakan parameter n_neighbors yaitu jumlah tetangga yang akan digunakan dalam perhitungan LOF. Pada model LOF menggunakan fungsi *fit_predict* yang digunakan untuk menilai apakah setiap baris dalam dataset merupakan outlier atau bukan, hasilnya yaitu dengan nilai -1 untuk outlier dan 1 untuk data yang bukan outlier.

```
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import LocalOutlierFactor
import numpy as np
import pandas as pd


# Membagi data menjadi numerik dan kategorikal
numerical_cols = data.select_dtypes(include=['int', 'float']).columns
categorical_cols = data.select_dtypes(include='object').columns

# Menggunakan SimpleImputer untuk mengisi nilai yang hilang dengan mean (dapat disesuaikan)
imputer = SimpleImputer(strategy='mean')
data[numerical_cols] = imputer.fit_transform(data[numerical_cols])

# Label Encoding untuk kolom kategorikal
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
data[categorical_cols] = data[categorical_cols].apply(lambda col: label_encoder.fit_transform(c
```

```
# Menggabungkan kolom numerik dan kategorikal
selected_data = data[numerical_cols.union(categorical_cols)]

# Normalisasi data numerik
scaler = StandardScaler()
selected_data[numerical_cols] = scaler.fit_transform(selected_data[numerical_cols])

# Membuat model LOF
clf = LocalOutlierFactor(n_neighbors=20)  # Jumlah tetangga yang digunakan
outlier_scores = clf.fit_predict(selected_data)

# Menampilkan indeks outlier
outlier_indices = np.where(outlier_scores == -1)[0]
print("Jumlah outlier:", len(outlier_indices))
```

```
Jumlah outlier: 38
```

Note : terdapat data yang memiliki outlier, sehingga data tersebut harus dihilangkan

1.2.12 - Identifikasi Jumlah Data

Dengan mengidentifikasi jumlah data, dapat mengetahui seberapa berbeda pada jumlah data di tiap-tiap label. Jika jumlah data antar label memiliki perbedaan yang sangat jauh maka akan mempengaruhi akurasi dan hasil klasifikasi sehingga perlu dilakukan penyeimbangan jumlah data di tiap labelnya.

```
# menghitung jumlah target pada data tanpa outlier
tiap_label = data['Selector'].value_counts()

print("Jumlah data pada tiap target :")
print(tiap_label)
```

```
Jumlah data pada tiap target :
Selector
1.0    416
2.0    167
Name: count, dtype: int64
```

Note : terjadi ketimpangan data yang signifikan sehingga dilakukan balancing data agar proporsi data tidak condong pada satu kelas saja.

Hasil Analisa Pada Data Understanding :

1. Data memiliki *missing values*

2. Data memiliki 13 data duplikat atau redundan
3. Data memiliki 1 outlier
4. Perbandingan proporsi data tiap target sangat jauh

1.3 – DATA PREPROCESSING –

Setelah memahami data, akan dilakukan tahap preprocessing untuk menangani masalah pada data yang sudah didefinisikan pada data understanding, yaitu:

1. Menghapus Missing Value
2. Menghapus Data Duplikat
3. Menghapus Outlier
4. Menyeimbangkan proporsi data tiap target

Setelah data siap, akan dilakukan skoring fitur kembali.

2

Summary

In summary, this book has no content whatsoever.

References