

Laporan Machine Learning Praktikum Data Processing

Nama : Surya Dwi Satria

Kelas : C7

NIM : 434231048

1. import pustaka & load data, set tampilan.

```
# 1) Import pustaka
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Opsional tampilan (biar tabel rapi di notebook)
pd.set_option('display.max_columns', None)

# 2) Load data (ganti path jika file bukan di folder yang sama)
df = pd.read_csv("movies_100.csv", low_memory=False)
print("Shape:", df.shape)
df.head()
```

Shape: (100, 11)

	title	year	genre	duration_min	rating	votes	revenue_musd	release_date	language	production_country	director
0	Popular Film #1	2018	Thriller	138.0	NaN	22083	6.72	2018-10-26	English	Germany	Director A.A
1	Popular Film #2	2008	Action	132.0	7.8	25949	18.19	2008-10-09	Hindi	Germany	Director B.H
2	Popular Film #3	1994	Comedy	132.0	NaN	56861	NaN	1994-03-17	Hindi	Canada	Director C.O
3	Popular Film #4	2022	NaN	119.0	8.2	52376	NaN	2022-08-09	Spanish	UK	Director D.V
4	Popular Film #5	1987	Animation	132.0	6.1	35862	21.11	1987-03-07	Korean	Spain	Director E.C

Penjelasan :

import pandas as pd (+ kemungkinan numpy, matplotlib) untuk olah data & visual.

pd.read_csv(...) untuk **membaca dataset**.

pd.set_option('display.max_rows/columns', ...) agar output tabel tidak terpotong.

df.head() (bila ada) untuk **lihat sampel baris atas**.

beberapa baris pertama dataset tampil rapi.

memastikan file dibaca benar & struktur kolom seperti yang kamu kira.

2. df.info()

```
# Info tipe data + jumlah non-null per kolom
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 11 columns):
#   Column              Non-Null Count  Dtype  
---  --
0   title                100 non-null   object  
1   year                 100 non-null   int64   
2   genre                93 non-null    object  
3   duration_min         92 non-null    float64  
4   rating               88 non-null    float64  
5   votes                100 non-null   object  
6   revenue_musd         86 non-null    object  
7   release_date         90 non-null    object  
8   language              94 non-null    object  
9   production_country   100 non-null   object  
10  director             100 non-null   object  
dtypes: float64(2), int64(1), object(8)
memory usage: 8.7+ KB
```

Penjelasan :

tampilkan **tipe data** dan **jumlah non-null** per kolom.

3. cek missing value per kolom (mis. df.isna().sum() dan persentasenya).

```
# Cek missing value per kolom
missing_count = df.isna().sum().sort_values(ascending=False)
missing_ratio = (df.isna().mean().sort_values(ascending=False) * 100).round(2)
print("Jumlah missing per kolom:\n", missing_count)
print("\nPersentase missing (%):\n", missing_ratio)
```

```

Jumlah missing per kolom:
revenue_musd      14
rating            12
release_date      10
duration_min       8
genre             7
language          6
title             0
year              0
votes             0
production_country 0
director          0
dtype: int64

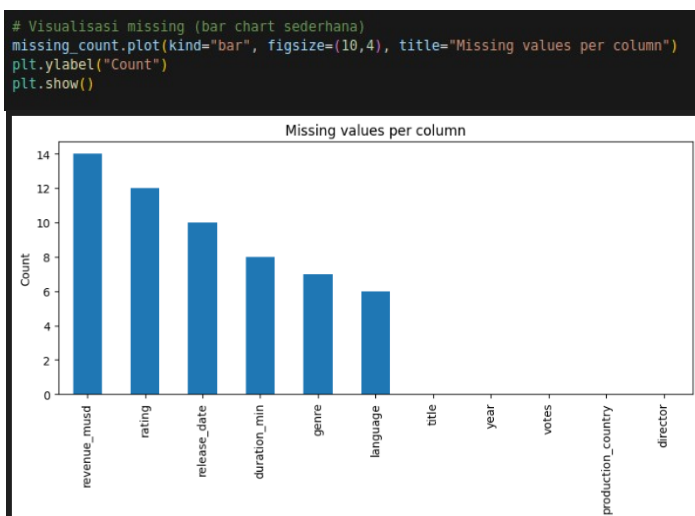
Persentase missing (%):
revenue_musd      14.0
rating            12.0
release_date      10.0
duration_min       8.0
genre             7.0
language          6.0
title             0.0
year              0.0
votes             0.0
production_country 0.0
director          0.0
dtype: float64

```

Penjelasan :

menghitung **jumlah & persentase NaN** tiap kolom (sering diurutkan menurun).

4. imputasi missing value untuk kolom **numerik** (mean/median) & **kategori** (mode).



Penjelasan :

cek ulang `df.isna().sum()` harus **0** (atau tinggal kolom yang sengaja dibiarkan).

5. deteksi outlier (Umum: Z-score atau IQR).

```

#S ---- Coerce angka yang tersimpan sebagai string (mis: '12,345' -> 12345) ----
def coerce_numeric_like(s: pd.Series, threshold=0.6):
    """Jika >= threshold% nilai bisa dikonversi ke angka, kembalikan seri numeric."""
    as_str = s.astype(str).str.replace(",", "").str.strip()
    numeric_try = pd.to_numeric(as_str, errors="coerce")
    if numeric_try.notna().mean() >= threshold:
        return numeric_try
    return s

df_fixed = df.copy()

for col in df_fixed.select_dtypes(include="object").columns:
    df_fixed[col] = coerce_numeric_like(df_fixed[col])

# ---- Parse tanggal bila ada kolom release_date ----
if "release_date" in df_fixed.columns:
    df_fixed["release_date"] = pd.to_datetime(df_fixed["release_date"], errors="coerce")

df_fixed.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 11 columns):
#   Column              Non-Null Count  Dtype
---  -
0   title                100 non-null   object
1   year                100 non-null   int64
2   genre               93 non-null    object
3   duration_min        92 non-null    float64
4   rating              88 non-null    float64
5   votes              100 non-null   int64
6   revenue_musd        85 non-null    float64
7   release_date        90 non-null    datetime64[ns]
8   language            94 non-null    object
9   production_country  100 non-null   object
10  director            100 non-null   object
dtypes: datetime64[ns](1), float64(3), int64(2), object(5)
memory usage: 8.7+ KB

```

Penjelasan :

daftar baris outlier atau jumlahnya, tahu fitur mana yang ekstrem sehingga bisa ditangani.

6. penanganan outlier

```
df_clean = df_filled.copy()

outlier_count = {}
for c in num_cols:
    q1, q3 = df_clean[c].quantile(0.25), df_clean[c].quantile(0.75)
    iqr = q3 - q1
    lo, hi = q1 - 1.5*iqr, q3 + 1.5*iqr
    # simpan jumlah outlier (sebelum capping)
    outlier_count[c] = int(((df_clean[c] < lo) | (df_clean[c] > hi)).sum())
    # winsorize = batasi nilai ekstrem ke batas lo/hi
    df_clean[c] = df_clean[c].clip(lower=lo, upper=hi)

pd.Series(outlier_count, name="outliers_before_capping")
```

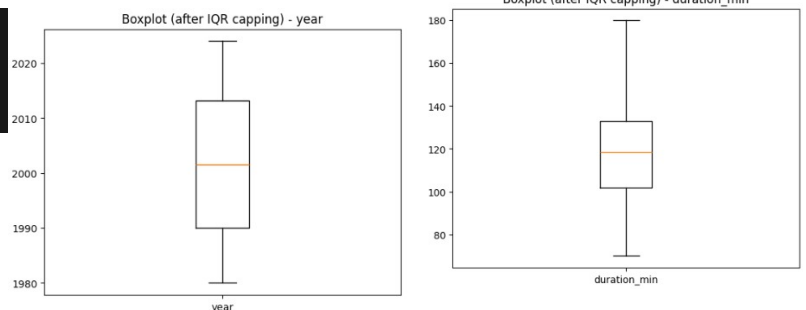
```
year          0
duration_min  2
rating        2
votes         8
revenue_musd 11
Name: outliers_before_capping, dtype: int64
```

Penjelasan :

jumlah baris mungkin berubah (kalau remove), atau distribusi fitur jadi lebih wajar (kalau capping/replacement).

7. visualisasi distribusi / outlier (mis. boxplot/histogram).

```
cols_to_plot = [c for c in num_cols[:2]] # ambil 2 kolom pertama
for c in cols_to_plot:
    plt.figure()
    plt.boxplot(df_clean[c].dropna(), vert=True, labels=[c])
    plt.title(f"Boxplot (after IQR capping) - {c}")
    plt.show()
```



Penjelasan :

validasi visual bahwa outlier sudah berkurang & data lebih “sehat”.

8. menyusun **dataset final** setelah cleaning.

```
df_final = df_clean.copy()

# 1) Panjang judul
if "title" in df_final.columns:
    df_final["title_len"] = df_final["title"].astype(str).str.len()

# 2) Flag film panjang: durasi > 120 menit - 1, else 0
if "duration_min" in df_final.columns:
    df_final["is_long"] = (df_final["duration_min"] > 120).astype(int)

# 3) Kategori rating: low (<6), mid (6-8], high (>8)
if "rating" in df_final.columns:
    df_final["rating_bin"] = pd.cut(
        df_final["rating"],
        bins=[-1, 6, 8, 10],
        labels=["low", "mid", "high"]
    )

df_final.head()
```

	title	year	genre	duration_min	rating	votes	revenue_musd	release_date	language	production_country	director	title_len	is_long	rating_bin
0	Popular Film #1	2018	Thriller	138.0	7.4	22083.0	6.72	2018-10-26	English	Germany	Director A.A	15	1	mid
1	Popular Film #2	2008	Action	132.0	7.8	25949.0	18.19	2008-10-09	Hindi	Germany	Director B.H	15	1	mid
2	Popular Film #3	1994	Comedy	132.0	7.4	56861.0	31.33	1994-03-17	Hindi	Canada	Director C.O	15	1	mid
3	Popular Film #4	2022	Action	119.0	8.2	52376.0	31.33	2022-08-09	Spanish	UK	Director D.V	15	0	high
4	Popular Film #5	1987	Animation	132.0	6.1	35862.0	21.11	1987-03-07	Korean	Spain	Director E.C	15	1	mid

Penjelasan:

satu variabel akhir yang siap dianalisis/dimodelkan.

9. summary & simpan hasil

```
df_final.to_csv("movies_clean_with_features.csv", index=False)

# Simpan ringkasan deskriptif (sebelum & sesudah) untuk lampiran
desc_before = df.describe(include="all").T
desc_after = df_final.describe(include="all").T

desc_before.to_csv("numeric_describe_before.csv")
desc_after.to_csv("numeric_describe_after.csv")

print("Saved:")
print("- movies_clean_with_features.csv")
print("- numeric_describe_before.csv")
print("- numeric_describe_after.csv")
```

Saved:

- movies_clean_with_features.csv
- numeric_describe_before.csv
- numeric_describe_after.csv

Penjelasan :

file CSV hasil preprocessing + (opsional) file ringkasan statistik.

10. muat balik file hasil simpan (validasi)

```
data_final1 = pd.read_csv("movies_clean_with_features.csv")
data_final2 = pd.read_csv("numeric_describe_before.csv")
data_final3 = pd.read_csv("numeric_describe_after.csv")

data_final1.isna().sum()
```

```
title      0
year       0
genre      0
duration_min  0
rating     0
votes      0
revenue_musd  0
release_date  0
language   0
production_country  0
director   0
title_len  0
is_long    0
rating_bin  0
dtype: int64
```

```
data_final2.isna().sum() Unnamed: 0    0
count              0
unique             3
top                3
freq              3
mean              8
std               8
min               8
25%              8
50%              8
75%              8
max               8
dtype: int64
```

```
data_final3.isna().sum() Unnamed: 0    0
count              0
unique             8
top               8
freq             8
mean             6
min             6
25%             6
50%             6
75%             6
max             6
std             7
dtype: int64
```

Penjelasan : memastikan file keluaran bisa dipakai ulang.