1 Praktikum 3

Muhamad Rizal Arfiyan - 22.11.5227 - IF11

https://github.com/rizalarfiyan/big-data

1.1 1. Import library

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
```

1.2 2. Baca file CSV menggunakan library pandas

[2]:
```
path = "./smartphones_cleaned_v6.csv"
data = pd.read_csv(path)
data.head()
```

[2]:

	brand_name	model	price	rating	has_5g	has_nfc	\
0	oneplus	OnePlus 11 5G	54999	89.0	True	True	
1	oneplus	OnePlus Nord CE 2 Lite 5G	19989	81.0	True	False	
2	samsung	Samsung Galaxy A14 5G	16499	75.0	True	False	
3	motorola	Motorola Moto G62 5G	14999	81.0	True	False	
4	realme	Realme 10 Pro Plus	24999	82.0	True	False	

	has_ir_blaster	processor_brand	num_cores	processor_speed	…	\
0	False	snapdragon	8.0	3.2	…	
1	False	snapdragon	8.0	2.2	…	
2	False	exynos	8.0	2.4	…	
3	False	snapdragon	8.0	2.2	…	
4	False	dimensity	8.0	2.6	…	

	refresh_rate	num_rear_cameras	num_front_cameras	os	\
0	120	3	1.0	android	
1	120	3	1.0	android	
2	90	3	1.0	android	
3	120	3	1.0	android	
4	120	3	1.0	android	

	primary_camera_rear	primary_camera_front	extended_memory_available	\
0	50.0	16.0	0	
1	64.0	16.0	1	
2	50.0	13.0	1	
3	50.0	16.0	1	
4	108.0	16.0	0	

```
   extended_upto  resolution_width resolution_height
0            NaN              1440              3216
1         1024.0              1080              2412
2         1024.0              1080              2408
3         1024.0              1080              2400
4            NaN              1080              2412

[5 rows x 26 columns]
```

1.3 3. Normalisasi data

- Normalisasi data dengan mengubah nilai kosong (null) dengan dengan 0
- Mengubah nilai boolean True dan False menjadi 1 dan 0
- Mengurutkan data jenis menjadi angka, semisal android = 0, ios = 1, other = 2, dst. Pada cara ini bisa menggunakan library pandas fungsi factorize (documentation).

[3]:
```
data.fillna(0, inplace=True)

for column in data.select_dtypes(include="bool").columns:
    data[column] = data[column].astype(int)

data["os"] = pd.factorize(data["os"])[0]
data["brand_name"] = pd.factorize(data["brand_name"])[0]
data["processor_brand"] = pd.factorize(data["processor_brand"])[0]

print(data.head())
```

```
   brand_name                     model  price  rating  has_5g  has_nfc  \
0           0              OnePlus 11 5G  54999    89.0       1        1
1           0  OnePlus Nord CE 2 Lite 5G  19989    81.0       1        0
2           1      Samsung Galaxy A14 5G  16499    75.0       1        0
3           2       Motorola Moto G62 5G  14999    81.0       1        0
4           3         Realme 10 Pro Plus  24999    82.0       1        0

   has_ir_blaster  processor_brand  num_cores  processor_speed  …  \
0               0                0        8.0              3.2  …
1               0                0        8.0              2.2  …
2               0                1        8.0              2.4  …
3               0                0        8.0              2.2  …
4               0                2        8.0              2.6  …

   refresh_rate  num_rear_cameras  num_front_cameras  os  primary_camera_rear  \
0           120                 3                1.0   0                 50.0
1           120                 3                1.0   0                 64.0
2            90                 3                1.0   0                 50.0
3           120                 3                1.0   0                 50.0
4           120                 3                1.0   0                108.0
```

```
   primary_camera_front  extended_memory_available  extended_upto  \
0                  16.0                          0            0.0
1                  16.0                          1         1024.0
2                  13.0                          1         1024.0
3                  16.0                          1         1024.0
4                  16.0                          0            0.0

   resolution_width  resolution_height
0              1440               3216
1              1080               2412
2              1080               2408
3              1080               2400
4              1080               2412

[5 rows x 26 columns]
```

1.4 4. Cari nilai korelasi tertinggi

- Bisa menggunakan library pandas fungsi corr untuk korelasi di setiap kolom (documentation). Attribute numeric_only=True adalah untuk memfilter hanya kolom yang bertipe data numeric (int, float, unsigned int, dll).

[4]:
```
correlation_matrix = data.corr(numeric_only=True)
price_correlation = correlation_matrix["price"].sort_values(ascending=False)
print(price_correlation)
```

```
price                       1.000000
internal_memory             0.557168
has_nfc                     0.470951
ram_capacity                0.386002
resolution_height           0.353578
os                          0.348308
resolution_width            0.340592
has_5g                      0.305066
refresh_rate                0.244115
processor_speed             0.175386
primary_camera_front        0.146122
fast_charging               0.139824
num_rear_cameras            0.125330
fast_charging_available     0.116739
screen_size                 0.113253
primary_camera_rear         0.092095
brand_name                  0.089374
num_front_cameras           0.055305
has_ir_blaster             -0.015807
num_cores                  -0.104890
rating                     -0.129441
processor_brand            -0.141318
```

```
battery_capacity            -0.189916
extended_upto               -0.272838
extended_memory_available   -0.448628
Name: price, dtype: float64
```

1.5 5. Cek korelasi menggunakan rumus korelasi pearson

[5]:
```
from scipy import stats

pearson_coef, p_value = stats.pearsonr(data["internal_memory"], data["price"])
print(
    "The Pearson Correlation Coefficient is",
    pearson_coef,
    " with a P-value of P =",
    p_value,
)
```

```
The Pearson Correlation Coefficient is 0.5571676328262642  with a P-value of P =
5.281230323250923e-81
```

1.6 6. Cek nama kolom dan value korelasi tertinggi

[6]:
```
highest_variable = price_correlation.index[1]
highest_value = price_correlation[1]
print(highest_variable, " - ", highest_value)
```

```
internal_memory  -  0.5571676328262649
```

1.7 7. Tampilkan dengan diagram scatter plot

[7]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x=highest_variable, y="price", data=data)
plt.title(
    f"Scatterplot of Price vs {highest_variable}\nCorrelation: {highest_value:.
 ↪2f}"
)
plt.xlabel(highest_variable)
plt.ylabel("Price (INR)")
plt.grid(True)
plt.text(
    0.5,
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
    fontsize=24,
    color="black",
    ha="center",
    va="center",
```

```
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=30,
)
plt.show()
```