data

November 16, 2024

Nazwa przedmiotu	Dataset
Projekt zespołowy - sztuczna inteligencja	laptop_prices_dataset
Piotr	Goraj
Numer albumu	55529
Bartosz	Kiałka
Numer albumu	55528
Data oddania sprawozdania	2024.11.16
Kierunek	Informatyka, II stopnia P, Stacjonarne

1 1. Instalacja i import potrzebnych bibliotek

[1]:
```
# !pip install pandas
# !pip install scikit-learn
# !pip install numpy
```

[2]:
```
import pandas as pd
import numpy as np
import re
from sklearn.preprocessing import LabelEncoder
```

2 2. Odczyt danych

[3]:
```
import pandas as pd

# załadowanie pliku CSV
FILE_PATH = './laptop_prices_dataset.csv'
data = pd.read_csv(FILE_PATH)
```

[4]:
```
# data info
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1275 entries, 0 to 1274
Data columns (total 23 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Company               1275 non-null   object
 1   Product               1275 non-null   object
 2   TypeName              1275 non-null   object
 3   Inches                1275 non-null   float64
 4   Ram                   1275 non-null   int64
 5   OS                    1275 non-null   object
 6   Weight                1275 non-null   float64
 7   Price_euros           1275 non-null   float64
 8   Screen                1275 non-null   object
 9   ScreenW               1275 non-null   int64
 10  ScreenH               1275 non-null   int64
 11  Touchscreen           1275 non-null   object
 12  IPSpanel              1275 non-null   object
 13  RetinaDisplay         1275 non-null   object
 14  CPU_company           1275 non-null   object
 15  CPU_freq              1275 non-null   float64
 16  CPU_model             1275 non-null   object
 17  PrimaryStorage        1275 non-null   int64
 18  SecondaryStorage      1275 non-null   int64
 19  PrimaryStorageType    1275 non-null   object
 20  SecondaryStorageType  1275 non-null   object
 21  GPU_company           1275 non-null   object
 22  GPU_model             1275 non-null   object
dtypes: float64(4), int64(5), object(14)
memory usage: 229.2+ KB
```

[5]:
```
# pierwsze trzy rekordy
data.head(3)
```

[5]:

	Company	Product	TypeName	Inches	Ram	OS	Weight	Price_euros	\
0	Apple	MacBook Pro	Ultrabook	13.3	8	macOS	1.37	1339.69	
1	Apple	Macbook Air	Ultrabook	13.3	8	macOS	1.34	898.94	
2	HP	250 G6	Notebook	15.6	8	No OS	1.86	575.00	

	Screen	ScreenW	…	RetinaDisplay	CPU_company	CPU_freq	CPU_model	\
0	Standard	2560	…	Yes	Intel	2.3	Core i5	
1	Standard	1440	…	No	Intel	1.8	Core i5	
2	Full HD	1920	…	No	Intel	2.5	Core i5 7200U	

	PrimaryStorage	SecondaryStorage	PrimaryStorageType	SecondaryStorageType	\
0	128	0	SSD	No	
1	128	0	Flash Storage	No	
2	256	0	SSD	No	

```
   GPU_company                GPU_model
0        Intel  Iris Plus Graphics 640
1        Intel        HD Graphics 6000
2        Intel         HD Graphics 620

[3 rows x 23 columns]
```

[6]:
```
# podsumowanie statystyczne
print(data.describe())
```

```
             Inches          Ram       Weight  Price_euros      ScreenW  \
count  1275.000000  1275.000000  1275.000000  1275.000000  1275.000000
mean     15.022902     8.440784     2.040525  1134.969059  1900.043922
std       1.429470     5.097809     0.669196   700.752504   493.346186
min      10.100000     2.000000     0.690000   174.000000  1366.000000
25%      14.000000     4.000000     1.500000   609.000000  1920.000000
50%      15.600000     8.000000     2.040000   989.000000  1920.000000
75%      15.600000     8.000000     2.310000  1496.500000  1920.000000
max      18.400000    64.000000     4.700000  6099.000000  3840.000000

            ScreenH     CPU_freq  PrimaryStorage  SecondaryStorage
count  1275.000000  1275.000000     1275.000000       1275.000000
mean   1073.904314     2.302980      444.517647        176.069020
std     283.883940     0.503846      365.537726        415.960655
min     768.000000     0.900000        8.000000          0.000000
25%    1080.000000     2.000000      256.000000          0.000000
50%    1080.000000     2.500000      256.000000          0.000000
75%    1080.000000     2.700000      512.000000          0.000000
max    2160.000000     3.600000     2048.000000       2048.000000
```

[7]:
```
# czy występują wartości null
display(data.isnull().sum())
```

```
Company             0
Product             0
TypeName            0
Inches              0
Ram                 0
OS                  0
Weight              0
Price_euros         0
Screen              0
ScreenW             0
ScreenH             0
Touchscreen         0
IPSpanel            0
RetinaDisplay       0
```

```
CPU_company             0
CPU_freq                0
CPU_model               0
PrimaryStorage          0
SecondaryStorage        0
PrimaryStorageType      0
SecondaryStorageType    0
GPU_company             0
GPU_model               0
dtype: int64
```

[8]:
```
# Wybrane kolumny do modeli
selected_columns = [ 'Company','TypeName', 'Price_euros',␣
 ↪'Inches','Ram','OS','Touchscreen','CPU_company','CPU_freq','CPU_model','GPU_company','GPU_m

# Wczytanie pliku CSV z wybranymi kolumnami
print("Wybrane kolumny CSV:")
df_selected_columns_csv = pd.read_csv(FILE_PATH, usecols=selected_columns)

# pierwsze 10 rekordów
display(df_selected_columns_csv.head(10))
```

```
Wybrane kolumny CSV:
```

	Company	TypeName	Inches	Ram	OS	Price_euros	Touchscreen	\
0	Apple	Ultrabook	13.3	8	macOS	1339.69	No	
1	Apple	Ultrabook	13.3	8	macOS	898.94	No	
2	HP	Notebook	15.6	8	No OS	575.00	No	
3	Apple	Ultrabook	15.4	16	macOS	2537.45	No	
4	Apple	Ultrabook	13.3	8	macOS	1803.60	No	
5	Acer	Notebook	15.6	4	Windows 10	400.00	No	
6	Apple	Ultrabook	15.4	16	Mac OS X	2139.97	No	
7	Apple	Ultrabook	13.3	8	macOS	1158.70	No	
8	Asus	Ultrabook	14.0	16	Windows 10	1495.00	No	
9	Acer	Ultrabook	14.0	8	Windows 10	770.00	No	

	CPU_company	CPU_freq	CPU_model	GPU_company	GPU_model
0	Intel	2.3	Core i5	Intel	Iris Plus Graphics 640
1	Intel	1.8	Core i5	Intel	HD Graphics 6000
2	Intel	2.5	Core i5 7200U	Intel	HD Graphics 620
3	Intel	2.7	Core i7	AMD	Radeon Pro 455
4	Intel	3.1	Core i5	Intel	Iris Plus Graphics 650
5	AMD	3.0	A9-Series 9420	AMD	Radeon R5
6	Intel	2.2	Core i7	Intel	Iris Pro Graphics
7	Intel	1.8	Core i5	Intel	HD Graphics 6000
8	Intel	1.8	Core i7 8550U	Nvidia	GeForce MX150
9	Intel	1.6	Core i5 8250U	Intel	UHD Graphics 620

3 3. Przygotowanie danych do modelu

[9]:
```python
def replace_with_top_n(column: str, n: int, other_label: str ="Other"):
    """
    Zamienia wartości w kolumnie na n najczęściej występujących,
    a pozostałe wartości na 'Other'.

    Params:
    - column: Nazwa kolumny do przetworzenia.
    - n: Liczba najczęściej występujących wartości, które mają zostać zachowane.
    - other_label: Etykieta dla pozostałych wartości (domyślnie "Other").

    Returns:
    - DataFrame z przekształconą kolumną.
    """
    top_n = df_selected_columns_csv[column].value_counts().nlargest(n).index
    df_selected_columns_csv[column] = df_selected_columns_csv[column].
 ↪apply(lambda x: x if x in top_n else other_label)
    return df_selected_columns_csv

def replace_with_log_transformation(column: str):
    """
    Zamienia wartości w kolumnie na wartości zlogarytmowane.

    Params:
    - column: Nazwa kolumny do przetworzenia.
    """
    df_selected_columns_csv[column] = np.log1p(df_selected_columns_csv[column])

def remove_model_number(column: str):
    """
    Usuwa końcowe cyfry (model) z nazwy, pozostawiając tylko część tekstową.

    Parameters:
    - text (str): Tekst do przetworzenia (np. "Core i5 7200U").

    Returns:
    - str: Przetworzony tekst bez końcowych cyfr (np. "Core i5").
    """
    df_selected_columns_csv[column] = df_selected_columns_csv[column].
 ↪apply(lambda text: re.sub(r'\s+\d+\w*$', '', text))
```

[10]:
```python
# top 5 Company
replace_with_top_n('Company', 5)

# top 3 TypeName
replace_with_top_n('TypeName', 3)
```

```
# top 4 OS
replace_with_top_n('OS', 4)

# top 2 CPU_company
replace_with_top_n('CPU_company', 2)

# Top 5 CPU_model
remove_model_number('CPU_model')
replace_with_top_n('CPU_model', 5)

# Top 5 GPU_company
replace_with_top_n('GPU_company', 5)

# Top 5 GPU_model
remove_model_number('GPU_model')
replace_with_top_n('GPU_model', 5)

"""
"""

# Transformacja logarytmiczna Ram
replace_with_log_transformation('Ram')

# Transformacja logarytmiczna Price_euros
replace_with_log_transformation('Price_euros')

# Transformacja logarytmiczna Inches
replace_with_log_transformation('Inches')

# Transformacja logarytmiczna CPU_freq
replace_with_log_transformation('CPU_freq')
```

4 4. Enkodowanie wartości kategorycznych

[ ]:
```
data_encoded = df_selected_columns_csv.copy()
label_encoders = {}

# konwersja kolumn kategorycznych na numeryczne korzystając z kodowania etykiet
for col in data_encoded.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    data_encoded[col] = le.fit_transform(data_encoded[col])
    label_encoders[col] = le

# pierwsze 20 enkodowanych rekortów
data_encoded.head(20)
```

[ ]:

	Company	TypeName	Inches	Ram	OS	Price_euros	Touchscreen	\
0	5	3	2.660260	2.197225	2	7.200940	0	
1	5	3	2.660260	2.197225	2	6.802328	0	
2	3	1	2.809403	2.197225	1	6.356108	0	
3	5	3	2.797281	2.833213	2	7.839309	0	
4	5	3	2.660260	2.197225	2	7.498094	0	
5	0	1	2.809403	1.609438	3	5.993961	0	
6	5	3	2.797281	2.833213	2	7.669014	0	
7	5	3	2.660260	2.197225	2	7.055917	0	
8	1	3	2.708050	2.833213	3	7.310550	0	
9	0	3	2.708050	2.197225	3	6.647688	0	
10	3	1	2.809403	1.609438	1	5.978633	0	
11	3	1	2.809403	1.609438	1	5.846410	0	
12	5	3	2.797281	2.833213	2	7.800151	0	
13	2	1	2.809403	1.609438	3	6.214408	0	
14	5	3	2.564949	2.197225	2	7.141562	0	
15	5	3	2.660260	2.197225	2	7.326170	0	
16	2	1	2.809403	2.197225	3	6.614726	0	
17	5	3	2.797281	2.833213	2	7.958227	0	
18	4	1	2.809403	2.197225	1	6.214608	0	
19	2	3	2.660260	2.197225	3	6.887553	1	

	CPU_company	CPU_freq	CPU_model	GPU_company	GPU_model
0	1	1.193922	3	2	3
1	1	1.029619	3	2	2
2	1	1.252763	3	2	2
3	1	1.308333	4	0	3
4	1	1.410987	3	2	3
5	0	1.386294	5	0	3
6	1	1.163151	4	2	3
7	1	1.029619	3	2	2
8	1	1.029619	4	3	3
9	1	0.955511	3	2	5
10	1	1.252763	3	2	2
11	1	1.098612	2	2	2
12	1	1.335001	4	0	3
13	1	1.098612	2	0	3
14	1	0.788457	5	2	2
15	1	1.193922	3	2	3
16	1	1.308333	4	0	3
17	1	1.360977	4	0	3
18	1	1.223775	2	3	0
19	1	0.955511	3	2	5

5 5. Zapis przygotowanych danych do pliku

[12]:
```
data_encoded.to_csv('model_data.csv', index=False)
```