

SPRAWOZDANIE

Zajęcia: Nauka o danych I

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 5 Data 23.11.2024 Temat: "Wykorzystanie narzędzi do eksploracyjnej analizy danych (EDA)" Wariant 11	Szymon Nycz Informatyka II stopień, niestacjonarne, 1 semestr, gr.1b
--	---

1. Polecenie:

Premise General Population COVID-19 Health Services Disruption Survey 2020

<http://ghdx.healthdata.org/record/ihme-data/premise-general-population-covid-19-health-services-disruption-survey-2020>

Link do repozytorium:

https://github.com/Maciek332/Semestr_1_Nycz/tree/master/NoD

2. Opis programu opracowanego

- Przygotowanie środowiska pracy

```
import pandas as pd

# Wczytanie danych
df = pd.read_csv('Housing.csv')

# Podstawowe informacje o danych
print(df.info())
print(df.describe())
```

0.0s Python

- Wczytanie i wstępne przetwarzanie danych

```
... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   price                 545 non-null   int64
1   area                 545 non-null   int64
2   bedrooms             545 non-null   int64
3   bathrooms            545 non-null   int64
4   stories              545 non-null   int64
5   mainroad             545 non-null   object
6   guestroom            545 non-null   object
7   basement             545 non-null   object
8   hotwaterheating      545 non-null   object
9   airconditioning      545 non-null   object
10  parking              545 non-null   int64
11  prefarea             545 non-null   object
12  furnishingstatus     545 non-null   object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB
None
```

	price	area	bedrooms	bathrooms	stories
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492
...					
25%	0.000000				
50%	0.000000				
75%	1.000000				
max	3.000000				

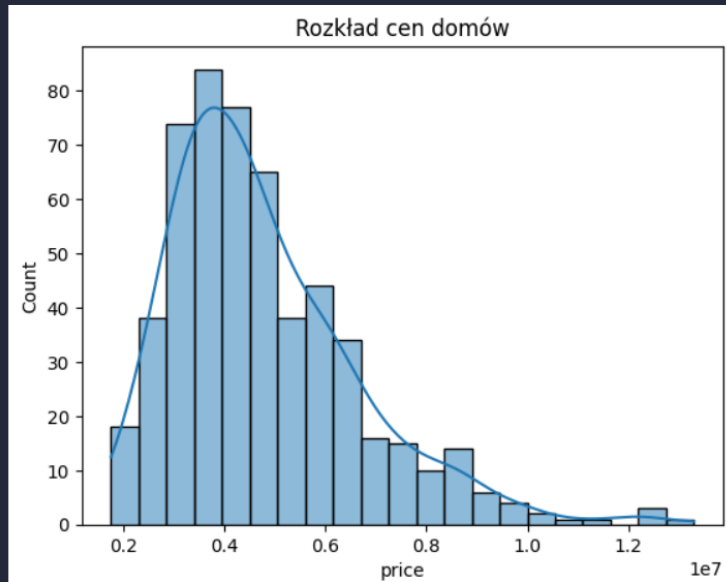
Output is truncated. View as a [scrollable element](#) or open in a [text editor](#). Adjust cell output [settings...](#)

+ Code + Markdown

- Detekcja wartości odstających

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Histogram rozkładu cen
sns.histplot(df['price'], kde=True)
plt.title('Rozkład cen domów')
plt.show()
```

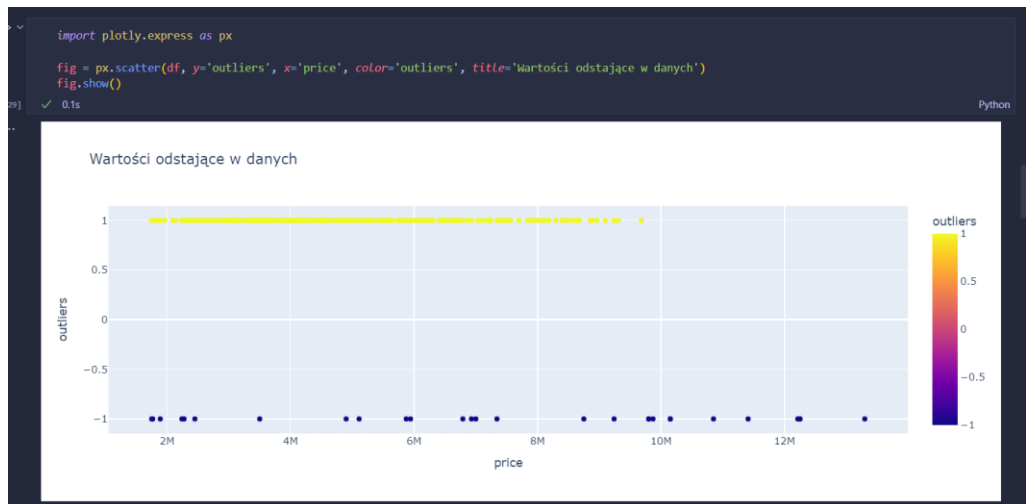


```
from sklearn.ensemble import IsolationForest

# Dostosowanie modelu Isolation Forest
isolation_forest = IsolationForest(contamination=0.05)
df['outliers'] = isolation_forest.fit_predict(df[['price', 'area']])
# Wyświetlenie wartości odstających
print(df[df['outliers'] == -1])
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	\
0	13300000	7420	4	2	3	yes	no	
1	12250000	8960	4	4	4	yes	no	
2	12250000	9960	3	2	2	yes	no	
3	12215000	7500	4	2	2	yes	no	
4	11410000	7420	4	1	2	yes	yes	
5	10850000	7500	3	3	1	yes	no	
6	10150000	8580	4	3	4	yes	no	
7	10150000	16200	5	3	2	yes	no	
8	9870000	8100	4	1	2	yes	yes	
9	9800000	5750	3	2	4	yes	yes	
10	9800000	13200	3	1	2	yes	no	
13	9240000	3500	4	2	2	yes	no	
20	8750000	4320	3	1	2	yes	no	
56	7343000	11440	4	1	2	yes	no	
64	7000000	11175	3	1	1	yes	no	
66	6930000	13200	2	1	1	yes	no	
69	6790000	12090	4	2	2	yes	no	
125	5943000	15600	3	1	1	yes	no	
129	5873000	11460	3	1	3	yes	no	
186	5110000	11410	2	1	2	yes	no	
211	4900000	12900	3	1	1	yes	no	
403	3500000	12944	3	1	1	yes	no	
520	2450000	7700	2	1	1	yes	no	
527	2275000	1836	2	1	1	no	no	
...								
530	unfurnished		-1					
537	unfurnished		-1					
541	semi-furnished		-1					
544	unfurnished		-1					

Output is truncated. View as a [scrollable element](#) or open in a [text editor](#). Adjust cell output settings...



- Analiza głównych składowych (PCA)
 - Wizualizacja redukcji wymiarowości- t-SNE

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Skalowanie danych
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df[['area', 'price', 'bedrooms']])

# PCA
pca = PCA(n_components=2)
principal_components = pca.fit_transform(scaled_data)

# Wynik PCA
df['PC1'] = principal_components[:, 0]
df['PC2'] = principal_components[:, 1]
print(pca.explained_variance_ratio_)

[0.57528546 0.28653528]
```

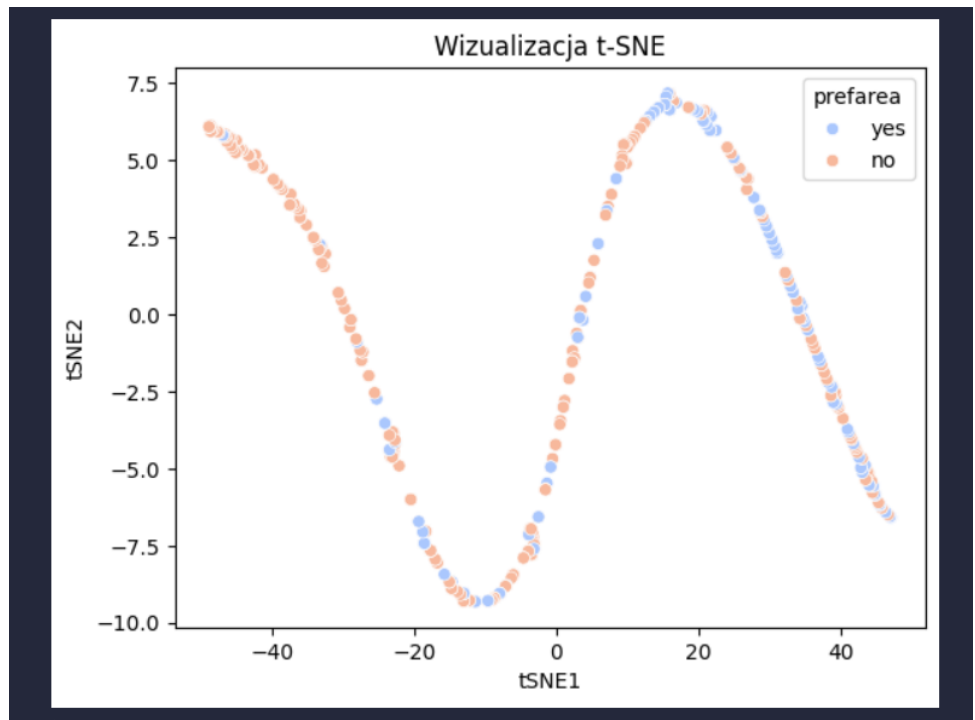


```
from sklearn.manifold import TSNE

# t-SNE
tsne = TSNE(n_components=2, random_state=42)
tsne_results = tsne.fit_transform(df[['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'parking']])

# Dodanie wyników do ramki danych
df['tSNE1'] = tsne_results[:, 0]
df['tSNE2'] = tsne_results[:, 1]

# Wizualizacja
sns.scatterplot(data=df, x='tSNE1', y='tSNE2', hue='prefarea', palette='coolwarm')
plt.title('Wizualizacja t-SNE')
plt.show()
```



- Wizualizacja redukcji wymiarowości-UMAP

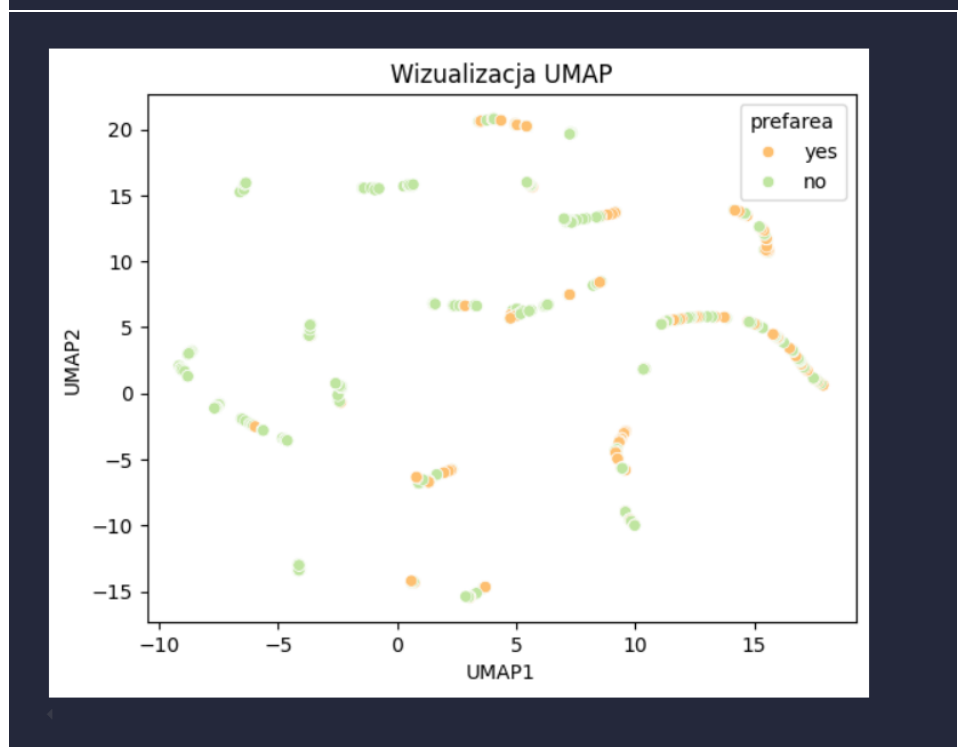
```
import umap

# UMAP
reducer = umap.UMAP(n_neighbors=10, min_dist=0.1, random_state=42)
umap_results = reducer.fit_transform(df[['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'parking']])

# Dodanie wyników do ramki danych
df['UMAP1'] = umap_results[:, 0]
df['UMAP2'] = umap_results[:, 1]

# wizualizacja
sns.scatterplot(data=df, x='UMAP1', y='UMAP2', hue='prefarea', palette='Spectral')
plt.title('Wizualizacja UMAP')
plt.show()
```

3] ✓ 1.8s



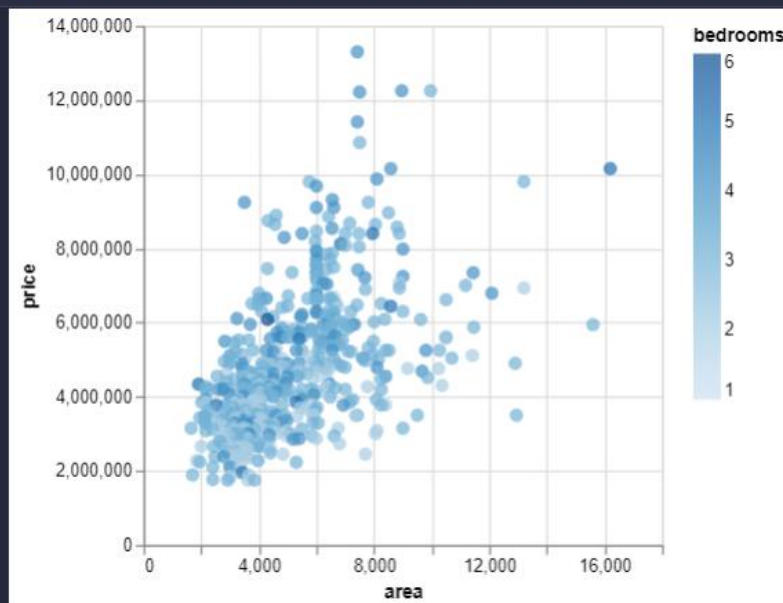
- Testy statystyczne

```
import altair as alt

chart = alt.Chart(df).mark_circle(size=60).encode(
    x='area',
    y='price',
    color='bedrooms',
    tooltip=['area', 'price', 'bedrooms']
).interactive()

chart.show()
```

✓ 0.1s



⌵ ⏴ ⏵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵

```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

# Model ANOVA
model = ols('price ~ C(bedrooms)', data=df).fit()
anova_results = anova_lm(model)

print(anova_results)
```

✓ 0.5s

	df	sum_sq	mean_sq	F	PR(>F)
C(bedrooms)	5.0	2.933324e+14	5.866649e+13	19.642037	5.359906e-18
Residual	539.0	1.609876e+15	2.986782e+12	NaN	NaN

3. Pytania kontrolne

- Jak działa algorytm Isolation Forest i jak interpretować jego wyniki?
Algorytm Isolation Forest wykrywa anomalie przez "izolowanie" rzadkich punktów w danych, budując losowe drzewa decyzyjne. Wyniki interpretujemy na podstawie tzw. score anomalności — punkty z wysokim wynikiem (bliskim 1) są anomaliami, a te z wynikiem bliskim 0 to dane normalne.
- W jaki sposób analiza PCA może pomóc w eksploracyjnej analizie danych?
PCA (Principal Component Analysis) redukuje wymiarowość danych, przekształcając je w nowe, niezależne zmienne (główne składowe), co ułatwia ich wizualizację i wykrywanie ukrytych wzorców.
- Jakie są zalety wykorzystania interaktywnych wizualizacji?
Interaktywne wizualizacje umożliwiają użytkownikom dynamiczne eksplorowanie danych, pozwalając na łatwiejsze odkrywanie wzorców, zrozumienie zależności i szybsze podejmowanie decyzji.
- Jak interpretować wyniki testu ANOVA?
ANOVA porównuje średnie wartości w różnych grupach. Jeśli wynik testu (p-wartość) jest mniejszy niż 0,05, oznacza to, że przynajmniej jedna grupa różni się statystycznie od innych.
- Jak działa algorytm t-SNE i kiedy warto go stosować?
t-SNE (t-Distributed Stochastic Neighbor Embedding) jest metodą redukcji wymiarowości, która zachowuje lokalne struktury danych, idealna do wizualizacji skomplikowanych danych w niskich wymiarach (np. 2D, 3D).
- W jaki sposób algorytm UMAP różni się od t-SNE?
UMAP (Uniform Manifold Approximation and Projection) jest podobny do t-SNE, ale jest szybszy i skalowalny na większe zbiory danych. UMAP zachowuje zarówno lokalne, jak i globalne struktury danych.
- Jak interpretować macierz korelacji?
Macierz korelacji przedstawia zależności między zmiennymi. Wartości bliskie 1 lub -1 wskazują na silną pozytywną lub negatywną korelację, natomiast wartości bliskie 0 oznaczają brak zależności.

4. Wnioski

t-SNE to zaawansowane narzędzie do redukcji wymiarowości i wizualizacji danych. Dzięki zdolności do zachowywania lokalnych zależności w danych o wysokiej liczbie wymiarów, pozwala na intuicyjne odkrywanie klastrów i wzorców. Jednak przy interpretacji wyników należy pamiętać o jego ograniczeniach w zakresie globalnych zależności i konieczności optymalizacji hiperparametrów.

UMAP stanowi wartościowe narzędzie do wizualizacji danych o dużej liczbie wymiarów, oferując przejrzystą, niskowymiarową reprezentację. Dzięki zachowywaniu zarówno lokalnych sąsiedztw, jak i globalnych struktur, UMAP ułatwia identyfikowanie kluczowych relacji i wzorców w złożonych zbiorach danych.

ANOVA to skuteczna metoda analizy różnic między grupami, a funkcja `anova_lm` w Pythonie umożliwia wygodne przeprowadzenie tej analizy. Przy interpretacji wyników najważniejsze są wartości p oraz F , które pozwalają określić, czy różnice między grupami są statystycznie istotne.