**Ćwiczenie: Interaktywna EDA**

Stworzona przy użyciu biblioteki Streamlit aplikacja, pozwala przeprowadzić eksploracyjną analizę danych w sposób interaktywny. Dzięki możliwości stworzenia interfejsu użytkownika, dodanie dynamicznych wykresów czy przycisków, zwiększa się przejrzystość – można wyświetlić wybrane wyniki przetwarzania danych, a cały kod pozostaje ukryty. Aplikacja stworzona z pomocą streamlit może zostać zaprezentowana osobom nietechnicznym, które bez konieczności znajomości programowania mogą zobaczyć rezultaty.

Aplikacja wykonana na potrzeby ćwiczenia jest podzielona na sekcje:

1. Podgląd danych.
2. Rozkład zmiennych.
3. Macierz korelacji.
4. Rozrzut zmiennych.

Ad 1. Podgląd danych to sekcja, którą można rozwinąć lub ukryć i zawiera statystyki opisowe, próbkę zbioru danych oraz informacje o nim.

A screenshot of a computer

Description automatically generated

A screenshot of a phone

Description automatically generated

Ad 2. Sekcja rozkład zmiennych zawiera drop-down, na którym można wybrać wyświetlaną zmienną. Wykresy to histogram oraz boxplot.

A screenshot of a graph

Description automatically generated

Ad 3. Sekcja Macierz korelacji to wykres prezentujący jak wygląda wzajemne skorelowanie między zmiennymi.

A screenshot of a graph

Description automatically generated

Ad 4. Sekcja Rozrzut zmiennych zawiera wykres rozrzutu z możliwością wyboru kolumny na osi X oraz Y

A screen shot of a computer

Description automatically generated

Kod który posłużył do wykonania aplikacji:

import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.datasets import load\_iris

from sklearn.datasets import fetch\_california\_housing

import matplotlib as mpl

@st.cache

**def** load\_housing\_data():

    housing = fetch\_california\_housing()

    df = pd.DataFrame(data=housing.data, columns=housing.feature\_names)

    df['target'] = housing.target

    return df

**def** show\_basic\_stats(data):

    st.subheader("Podgląd danych")

    st.write(data.head(10))

    st.subheader("Typy kolumn")

    st.write(data.dtypes)

    st.subheader("Podstawowe statystyki opisowe")

    st.write(data.describe())

    st.subheader("Wartości brakujące")

    st.write(data.isnull().sum())

**def** handle\_stats\_button(data):

    st.subheader("Statystyki zbioru danych")

    if 'button\_clicked' not in st.session\_state:

        st.session\_state.button\_clicked = False

*# Render DataFrame if button is clicked*

    if st.session\_state.button\_clicked:

        show\_basic\_stats(data)

    else:

        st.markdown("Statystyki są schowane")

*# Button to toggle DataFrame visibility*

    if st.button("Statystyki"):

        st.session\_state.button\_clicked = not st.session\_state.button\_clicked

*# Funkcja do wizualizacji rozkładu zmiennych*

**def** show\_distribution\_plots(data):

    st.subheader("Rozkład zmiennych")

    mpl.rcParams['axes.labelsize'] = 20

    mpl.rcParams['axes.titlesize'] = 24

    selected\_col = st.selectbox("Wybierz zmienną", data.columns)

    if selected\_col:

        fig, ax = plt.subplots(1,2,figsize=(20,10))

        col = data[data[selected\_col] <= data[selected\_col].mean() + 3 \* data[selected\_col].std()][selected\_col]

        sns.histplot(x=col, bins=10, kde=True, ax=ax[0])

        sns.boxplot(data=col, ax=ax[1])

        ax[0].set\_title(**f**"Distribution zmiennej {selected\_col}")

        ax[1].set\_title(**f**"Boxplot zmiennej {selected\_col}")

        st.pyplot(fig)

**def** show\_corrplot(data):

    st.subheader("Macierz korelacji")

    correlation\_matrix = data.corr()

    fig, ax = plt.subplots(figsize=(12,8))

*# Create correlation plot using Seaborn*

    sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f", ax=ax)

    ax.set\_title("Corrplot")

    st.pyplot(fig)

**def** show\_scatterplot(data):

    st.subheader("Rozrzut zmiennych")

    data\_frac = data.sample(frac=0.1, random\_state=42)

    selected\_col\_x = st.selectbox("Wybierz oś X", data\_frac.columns)

    selected\_col\_y = st.selectbox("Wybierz oś Y", data.columns)

*# Plot chart based on selected column*

    if selected\_col\_x and selected\_col\_y:

        fig, ax = plt.subplots(figsize=(20,10))

        sns.scatterplot(x=data\_frac[selected\_col\_x], y=data\_frac[selected\_col\_y],ax=ax)

        ax.set\_title(**f**"Wykres rozrzutu '{selected\_col\_y}' względem '{selected\_col\_x}'")

        st.pyplot(fig)

**def** main():

    housing\_data = load\_housing\_data()

    handle\_stats\_button(housing\_data)

    show\_distribution\_plots(housing\_data)

    show\_corrplot(housing\_data)

    show\_scatterplot(housing\_data)

if \_\_name\_\_ == "\_\_main\_\_":

    main()