**METODY EKSPLORACJI DANYCH**

Laboratorium. Klasyfikacja na podstawie klasyfikatora bayesowskiego

i najbliższego sąsiedztwa

Prowadzący: Wykonali:

dr inż. Romuald Hoffmann pchor. Michał ADAMCZEWSKI

pchor. Mikołaj ADAMSKI

pchor. Przemysław SUJECKI

Obraz zawierający tekst, Czcionka, zrzut ekranu, papier

Opis wygenerowany automatycznie

Importujemy biblioteki:

import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.naive\_bayes import MultinomialNB  
from sklearn.feature\_extraction.text import CountVectorizer  
from sklearn.metrics import accuracy\_score  
from nltk.corpus import stopwords  
import nltk  
nltk.download('stopwords')

# Wczytanie danych

try:

try:  
 dane\_dot = pd.read\_excel('C:\\Users\\michu\\Desktop\\WAT\\SEMESTR-7\\MED\\Lab3\\dane\_dot.xlsx', header=None)  
 dane\_nie\_dot = pd.read\_excel('C:\\Users\\michu\\Desktop\\WAT\\SEMESTR-7\\MED\\Lab3\\dane\_niedot.xlsx', header=None)  
except FileNotFoundError:  
 print("Nie można odnaleźć plików Excel. Sprawdź, czy ścieżki są poprawne.")

Definiujemy funkcję, która cały tekst w języku angielskim (lang=’en’) zmieni na małe litery.

def preprocess(text, lang='en'):  
 if text is not None:  
 return text.lower()  
 return ""

# Oznaczanie klas

dane\_dot['label'] = 1  
dane\_nie\_dot['label'] = 0

Łączymy dane z obu zakładek (dane dot. mandrill, dane dot. innych).

# Łączenie danych

dane = pd.concat([dane\_dot, dane\_nie\_dot])  
dane[0] = dane[0].apply(preprocess)

Dzielimy zbiory na dane treningowe i testowe w stosunku 70% do 30%.

# Podział na zbiory

X\_train, X\_val, y\_train, y\_val = train\_test\_split(dane[0], dane['label'], test\_size=0.3, random\_state=42)

Usuwamy tzw. stopwordsy ze zdań.

# Usuwanie słów stopu

stop\_words = 'english' # Zmiana na łańcuch znakowy  
vectorizer = CountVectorizer(stop\_words=stop\_words,lowercase = True)  
X\_train\_counts = vectorizer.fit\_transform(X\_train)  
X\_val\_counts = vectorizer.transform(X\_val)

# Tworzenie i trening modelu

model = MultinomialNB()  
model.fit(X\_train\_counts, y\_train)

# Walidacja modelu

predictions = model.predict(X\_val\_counts)  
accuracy = accuracy\_score(y\_val, predictions)  
print(f'Dokładność modelu: {accuracy}')  
print(dane)

Dokładność modelu:



Fragment danych po processingu:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Ilość połączonych danych:

Obraz zawierający Czcionka, tekst, Grafika, typografia

Opis wygenerowany automatycznie

Bibliografia:

[0] dr inż. Romuald Hoffmann, prof. WAT, Notatki dla studentów, Warszawa 2023.

[1] Larose D. T., Metody i modele eksploracji danych, Wydawnictwo Naukowe PWN, Warszawa 2012.

[2] <https://scikit-learn.org/stable/>

[3] Fred Nwanganga, Mike Chapple, Praktyczne uczenie maszynowe w języku R, Wiley, Warszawa, 2022.

[4] Laurence Moroney, Sztuczna inteligencja i uczenie maszynowe dla programistów, Helion O’Reilly, 2021.

[5] <https://pl.wikipedia.org/wiki/Wikipedia:Strona_g%C5%82%C3%B3wna>

[6] Aurelie Geron, Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow, Helion O’Reilly, 2018