**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**Інститут комп'ютерних наук та інформаційних технологій**

**Кафедра систем штучного інтелекту**



**Звіт до лабораторної роботи №1**

**з дисципліни “ Машинне навчання ”**

**На тему: “ Основи роботи з NumPy ”**

**Виконав**:

ст. гр. КН-311

Ткачук Орест

**Викладач:**

Бойчук А. Р.

Львів – 2021

**Мета:** засвоїти основні відомості про роботу з фреймворком NumPy. Навчитись розпізнавати нетипові елементи в даних.

**Завдання:**

1. Ознайомитись з функціоналом numpy. Для цього виконати вправи з <https://colab.research.google.com/drive/1aLVhNnrRijgbJhN45P4AxYieEz5srZ8Z?usp=sharing> до рівня Intermediate включно.
2. Порахувати z-score незалежно для кожної з цифрових колонок згідно варіанту. Для обрахунку використати лише numpy.
3. Агрегувати пораховані z-score(наприклад усередненням). Візуалізувати датасет і знайдені нетипові дані(ті, для яких z-score > 3) на scatter plot. Нетипові дані і решту датасету візуалізувати різними кольорами. У випадку якщо нетипових елементів менше 5, обрати 5-10 елементів з найвищим Z-score як нетипові.

**Виконання:**

**Варіант 4**

|  |  |  |
| --- | --- | --- |
| Варіант # | Датасет | Назви колонок з якими потрібно зробити аналіз |
| 4 | 'car\_crashes' | 'total', 'speeding', 'alcohol' |

1. Порахувати z-score незалежно для кожної з цифрових колонок згідно варіанту. Для обрахунку використати лише numpy.

for i in columns:

std = float(np.std(car\_crashes[i], ddof = 0))

mean = float(np.mean(car\_crashes[i]))

print(' std: ', std, '\n mean: ', mean)

car\_crashes[i+'\_Zscore'] = abs(car\_crashes[i] - mean) / std

car\_crashes['Zscore'] += car\_crashes[i + '\_Zscore']

1. Агрегувати пораховані z-score(наприклад усередненням). Візуалізувати датасет і знайдені нетипові дані(ті, для яких z-score > 3) на scatter plot. Нетипові дані і решту датасету візуалізувати різними кольорами. У випадку якщо нетипових елементів менше 5, обрати 5-10 елементів з найвищим Z-score як нетипові.

car\_crashes\_top = car\_crashes.loc[car\_crashes['Zscore'] >= 3]

if len(car\_crashes\_top.index) < 5:

car\_crashes = car\_crashes.sort\_values('Zscore',ascending = False)

car\_crashes\_top = car\_crashes.head(7)

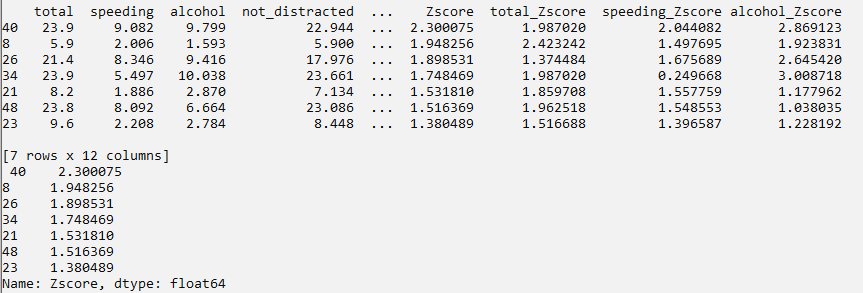


Рис. 1 найбільші Z-score з агрегацію(усередненням)

**Візуалізація:**

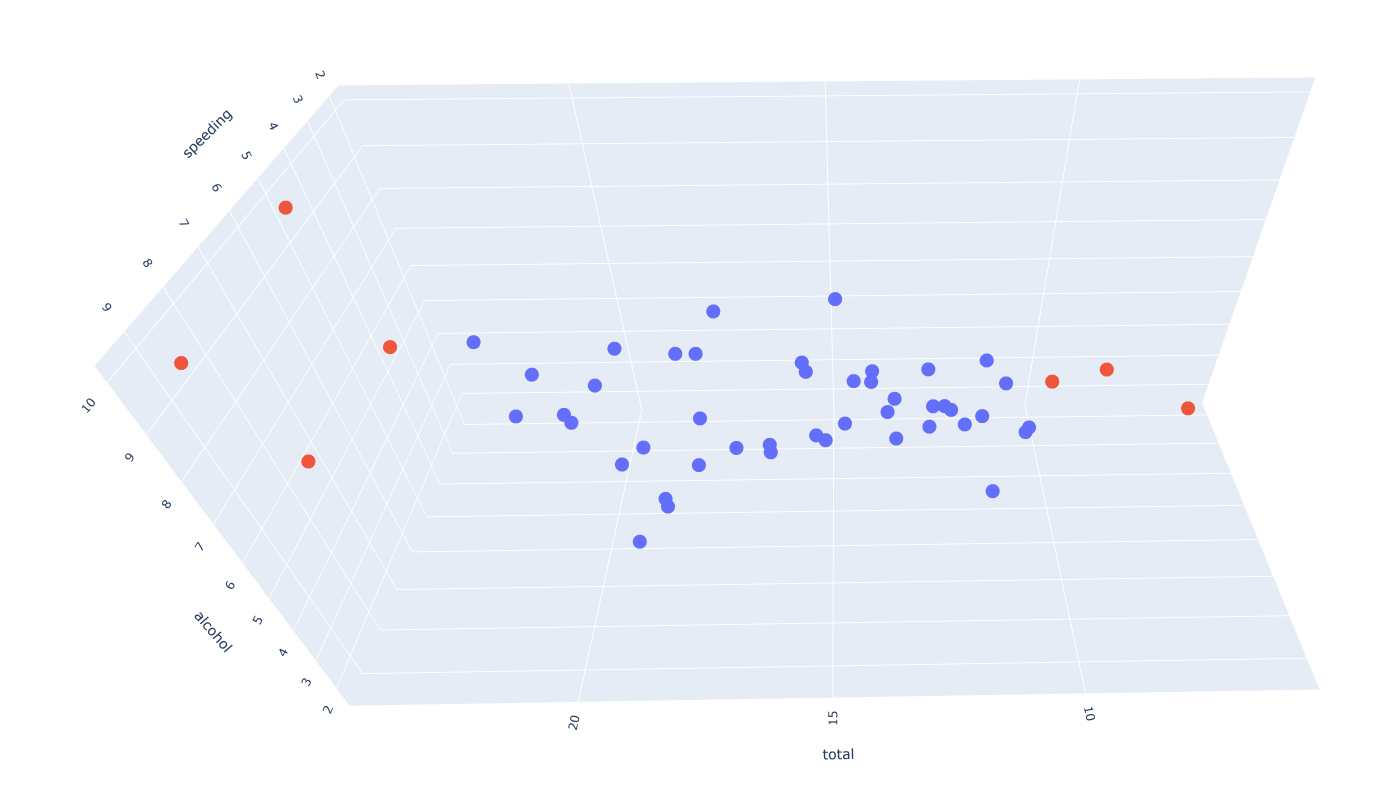


Рис. 2 Візуалізація датасету

**Висновок:** під час даної лабораторної роботи, я виконав підрахунок Z-score для заданих колонок за допомогою NumPy, зробив їх агрегацію та розпізнав нетипові елементи в даних.

**Додаток:**

import numpy as np

import scipy as sp

import seaborn as sns

import matplotlib.pyplot as plt

import plotly

import plotly.graph\_objects as go

car\_crashes = sns.load\_dataset('car\_crashes')

columns = ['total','speeding','alcohol']

print(car\_crashes[columns])

car\_crashes['Zscore'] = 0

for i in columns:

std = float(np.std(car\_crashes[i], ddof = 0))

mean = float(np.mean(car\_crashes[i]))

print(i,'\n np.std: ', std, '\n mean: ', mean)

car\_crashes[i+'\_Zscore'] = abs((car\_crashes[i] - mean) / std)

print(car\_crashes[i+'\_Zscore'],'\n')

car\_crashes['Zscore'] += car\_crashes[i + '\_Zscore']

car\_crashes['Zscore'] = car\_crashes['Zscore'] / len(columns)

car\_crashes\_top = car\_crashes.loc[car\_crashes['Zscore'] >= 3]

if len(car\_crashes\_top.index) < 5:

car\_crashes = car\_crashes.sort\_values('Zscore',ascending = False)

car\_crashes\_top = car\_crashes.head(7)

car\_crashes\_bot = car\_crashes.tail(len(car\_crashes.index) - len(car\_crashes\_top.index))

print(car\_crashes.head(len(car\_crashes\_top.index)), '\n', car\_crashes.head(len(car\_crashes\_top.index))['Zscore'])

fig = go.Figure()

fig.add\_trace(go.Scatter3d(x = car\_crashes\_bot['total'], y = car\_crashes\_bot['speeding'], z = car\_crashes\_bot['alcohol'], mode = 'markers'))

fig.add\_trace(go.Scatter3d(x = car\_crashes\_top['total'], y = car\_crashes\_top['speeding'], z = car\_crashes\_top['alcohol'], mode = 'markers'))

fig.update\_layout(scene = dict(xaxis\_title='total', yaxis\_title='speeding', zaxis\_title='alcohol'))

plotly.offline.plot(fig, filename='visualize.html')