**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Звіт**

З лабораторної роботи №1 з дисципліни

«Прикладні задачі машинного навчання»

На тему:

**«Введення в data science»**

**Виконали студенти**  *ІП-24 Новиков Гліб Костянтинович*

(шифр, прізвище, ім'я, по батькові) *13.02.2024*

*ІК-21 Насібов Рафаель Сафадович*

(шифр, прізвище, ім'я, по батькові)

**Перевірив**  *.Нестерук А. О.*

(прізвище, ім'я, по батькові)

Київ 2024

**Лабораторна робота №1**

**«Введення в data science»**

**Мета:**

Засвоїти основи роботи з даними, включаючи їх збір, аналіз та візуалізацію, а також використання бібліотеки Python для ефективної обробки та представлення даних.

**Постановка задачі лабораторної роботи №1:**

1. На сайті <http://www.ukrstat.gov.ua/> оберати дані які для Вас є цікавими, можна використати будь-який ресурс з відкритими даними, та завантажте дані.
2. Знайти математичне сподівання, медіану, моду, дисперсію, середньоквадратичне відхилення (поясніть їх зміст).
3. Візуалізувати завантажені дані за допомогою гістограми.
4. Для цих даних проробити всі дії з пункту колекції Series і DataFrame бібліотеки pandas.
5. Виконати первинну обробку даних
6. Прочитати набір даних катастрофи «Титаніка»

Набір даних катастрофи «Титаніка» належить до числа найпопулярніших наборів даних машинного навчання і доступний в багатьох форматах, включаючи CSV: <https://vincentarelbundock.github.io/Rdatasets/datasets.html>.

1. Завантажити набір даних катастрофи «Титаніка» за URL адресою. Якщо у вас є URL-адресу, що представляє набір даних в форматі CSV, то ви можете завантажити його в DataFrame функцією read\_csv - припустимо, з GitHub.
2. Переглянути рядки набору даних катастрофи «Титаніка».

Набір даних містить понад 1300 рядків, кожен рядок це інформація про одного пасажира. За даними «Вікіпедії», на борту було приблизно 1317 пасажирів, а 815 з них загинули. Для великих наборів даних при виведенні DataFrame показуються тільки перші 30 рядків, потім йде три крапки «...» і останні 30 рядків.

Для економії місця перегляньте перші і останні п'ять рядків за допомогою методів head і tail колекції DataFrame. Обидва методи за замовчуванням повертають п'ять рядків, але число виведених рядків можна передати в аргументі.

1. Налаштувати назви стовпців

Назва першого стовпчика в наборі даних виглядає досить дивно ('Unnamed: 0'). Цю проблему можна вирішити налаштуванням імен стовпців.

Замініть 'Unnamed: 0' на 'name' і скоротіть 'passengerClass' до 'class'.

1. Провести простий аналіз даних: визначте наймолодшого пасажира, найстаршого, який був середній вік пасажирів та статистику по пасажирам які вижили. Відсортуйте всіх жінок з кают 1-го класу, знайдіть наймолодшу та найстаршу серед них, кількість вижевших. Зверніть увагу на розбіжності в значенні count (1046) і кількості рядків даних в наборі даних (1309 - при виклику tail індекс останнього рядка дорівнював 1308). Тільки 1046 рядків даних (значення count) містили значення age. Решта результатів були відсутні і були помічені NaN, як в рядку 1305. При виконанні обчислень бібліотека pandas за замовчуванням ігнорує відсутні дані (NaN).
2. Побудувати гістограму віку пасажирів.

Візуалізація - хороший спосіб ближче познайомитися з даними. Pandas містить багато вбудованих засобів візуалізації, реалізованих на базі Matplotlib. Щоб використовувати їх, спочатку включіть підтримку Matplotlib в Ipython.

Гістограма наочно показує розподіл числових даних за діапазоном значень. Метод hist колекції DataFrame автоматично аналізує дані кожного числового стовпця і будує відповідну гістограму. Щоб переглянути гістограми по кожному числовому стовпці даних, викличте hist для своєї колекції DataFrame.

**Обрані дані**

Обраними даними для лабораторної роботи були «Регіональні обсяги зовнішньої торгівлі товарами за січень–вересень 2023 року»:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Regional volumes of foreign trade in Goods, January–September 2023 1*** | | | | | | | | |
|  | Експорт/*Exports* | | | Імпорт/*Imports* | | | Сальдо/ *Balance* |  |
| тис.дол. США/ *thsd. USD* | у % до 9 місяців 2022/in % to January-September 2022 | у % до загального обсягу/ *% of the total volume* | тис.дол. США/ *thsd. USD* | у % до 9 місяців 2022/in % to January-September 2022 | у % до загального обсягу/ *% of the total volume* |
|  |
| **Україна** | **27144776,9** | **82,2** | **100,0** | **46547149,3 2** | **119,2** | **100,0** | **########** | ***Ukraine*** |  |
| у тому числі |  | | | | | | | *of which* |  |
| Вiнницька | 1133651,2 | 104,5 | 4,2 | 672785,7 | 151,3 | 1,4 | 460865,5 | *Vinnytsya* |  |
| Волинська | 645737,7 | 81,9 | 2,4 | 1900030,1 | 132,4 | 4,1 | -1254292,4 | *Volyn* |  |
| Днiпропетровська | 3610917,8 | 69,3 | 13,3 | 3386500,0 | 118,3 | 7,3 | 224417,8 | *Dnipropetrovsk* |  |
| Донецька | 175022,9 | 73,7 | 0,6 | 39227,8 | 32,9 | 0,1 | 135795,1 | *Donetsk* |  |
| Житомирська | 418267,6 | 80,2 | 1,5 | 904711,9 | 121,5 | 1,9 | -486444,2 | *Zhytomyr* |  |
| Закарпатська | 1124405,6 | 73,0 | 4,1 | 1053379,2 | 83,5 | 2,3 | 71026,4 | *Zakarpattya* |  |
| Запорiзька | 1057208,0 | 42,3 | 3,9 | 426065,2 | 41,9 | 0,9 | 631142,8 | *Zaporizhzhya* |  |
| Івано–Франківська | 474722,1 | 85,6 | 1,7 | 395948,9 | 82,8 | 0,9 | 78773,2 | *Ivano–Frankivsk* |  |
| Київська | 1325004,2 | 84,8 | 4,9 | 3397623,6 | 131,9 | 7,3 | -2072619,4 | *Kyiv* |  |
| Кiровоградська | 612892,8 | 100,5 | 2,3 | 203150,4 | 113,2 | 0,4 | 409742,5 | *Kirovohrad* |  |
| Луганська | 141,2 | 1,7 | 0,0 | 1028,3 | 9,5 | 0,0 | -887,1 | *Luhansk* |  |
| Львiвська | 2003860,4 | 88,8 | 7,4 | 4302555,9 | 112,5 | 9,2 | -2298695,5 | *Lviv* |  |
| Миколаївська | 766576,3 | 62,1 | 2,8 | 398592,0 | 64,1 | 0,9 | 367984,3 | *Mykolayiv* |  |
| Одеська | 1369283,8 | 91,9 | 5,0 | 1652727,6 | 107,9 | 3,6 | -283443,8 | *Odesa* |  |
| Полтавська | 997829,9 | 66,1 | 3,7 | 631241,8 | 67,1 | 1,4 | 366588,1 | *Poltava* |  |
| Рiвненська | 457773,4 | 82,8 | 1,7 | 573021,8 | 155,9 | 1,2 | -115248,4 | *Rivne* |  |
| Сумська | 525873,0 | 104,5 | 1,9 | 404945,9 | 97,3 | 0,9 | 120927,1 | *Sumy* |  |
| Тернопiльська | 555512,2 | 105,4 | 2,0 | 501334,0 | 138,7 | 1,1 | 54178,2 | *Ternopil* |  |
| Харкiвська | 573763,0 | 87,4 | 2,1 | 1101910,5 | 125,9 | 2,4 | -528147,5 | *Kharkiv* |  |
| Херсонська | 114129,1 | 214,4 | 0,4 | 13988,8 | 46,2 | 0,0 | 100140,3 | *Kherson* |  |
| Хмельницька | 568266,1 | 106,3 | 2,1 | 490491,9 | 94,9 | 1,1 | 77774,2 | *Khmelnytskiy* |  |
| Черкаська | 913789,1 | 106,5 | 3,4 | 536814,4 | 128,0 | 1,2 | 376974,7 | *Cherkasy* |  |
| Чернiвецька | 150856,9 | 72,8 | 0,6 | 367854,7 | 113,0 | 0,8 | -216997,8 | *Chernivtsi* |  |
| Чернігівська | 592111,4 | 118,4 | 2,2 | 227151,6 | 106,7 | 0,5 | 364959,8 | *Chernihiv* |  |
| м.Київ | 6892657,6 | 92,2 | 25,4 | 20101776,8 | 132,3 | 43,2 | -13209119,2 | *Kyiv city* |  |
| |  | | --- | | 1 Дані наведено без урахування тимчасово окупованих російською федерацією територій та частини територій, на яких ведуться (велися) бойові дії/ *1 Data exclude the territories which are temporarily occupied by the russian federation and part of territories where the military actions are/were conducted.* | | | | | | | | | |  |
| 2 З урахуванням не розподілених за регіонами поставок газу природного./ *2 Including natural gas supply undistributed among regions.* | | | | | | | | |  |
| **Примітка.** В окремих випадках сума складових може не дорівнювати підсумку у зв’язку з округленням даних./***Note.*** *In some cases, the sum of the components may not be equal to the result due to rounding data.* | | | | | | | | |  |

Таблиця 1 ­– зміст xls-файлу «Регіональні обсяги зовнішньої торгівлі товарами за січень–вересень 2023 року»

**Код програми на мові Python**

import-export.py

import pandas as pd  
import statistics as s  
import numpy as np  
import matplotlib.pyplot as plt  
import num4 as n4  
from matplotlib.gridspec import GridSpec  
import os  
  
*#1*df=pd.read\_excel("stat.xls",skiprows=6,skipfooter=3)  
saldo=df["Unnamed: 7"]  
names\_saldo=df.iloc[:,[0,7]]  
  
print('-----------------Сальдо по областям України за 2023 рік--------------------')  
print(names\_saldo)  
  
*#2*print('-----------------Статистика по сальдо--------------------')  
mean = saldo.mean()  
print("mean", mean)  
mode = saldo.mode()  
print("mode", mode if mode.size < saldo.size else "no mode")  
median = s.median(sorted(saldo))  
print("median",median)  
dysp = s.pvariance(saldo)  
print("pvariance",dysp)  
std = s.pstdev(saldo)  
print("std",std)  
  
*#3*gs = GridSpec(1, 5)  
fig = plt.figure(figsize=(10, 6))  
def hui(plot\_column\_num:int, column\_dict\_name:str, column\_dict\_num:int, x\_label:str, label:str, multiplier=1.0):  
 ax = fig.add\_subplot(gs[0, plot\_column\_num])  
 cities\_data = df.iloc[:, [0, column\_dict\_num]]  
 cities\_data\_hist = cities\_data.to\_dict('list')  
 ys = cities\_data\_hist[column\_dict\_name]  
 if multiplier != 1.0:  
 temp = []  
 for i in cities\_data\_hist[column\_dict\_name]:  
 temp.append(i \* multiplier)  
 ys = temp  
 ax.barh(cities\_data\_hist['у тому числі'], ys)  
 ax.set\_yticklabels(cities\_data\_hist['у тому числі'], rotation=0)   
 xticks = np.linspace(round(min(ys), 2), round(max(ys), 3), num=5)  
 ax.set\_xticks(xticks)  
 ax.set\_xticklabels(xticks,rotation=90)  
 ax.set\_ylabel(x\_label)   
 ax.set\_title(label)  
 ax.grid(axis='x')   
 return ax  
  
*# Сальдо - гістрограма*hui(0,'Unnamed: 7',7,'Міста','Сальдо, млрд. долл',10e-7)  
  
*# експорт - гістрограма*hui(2,'Unnamed: 1',1,'Міста','Експорт, млрд. долл',10e-7)  
  
*# Імпорт - гістрограма*hui(4,'Unnamed: 4',4,'Міста','Імпорт, млрд. долл',10e-7)  
  
plt.show()  
  
*# 4*print('\n----------------------------------------------------')  
print('-----------------Колекція Series--------------------')  
print('----------------------------------------------------\n')  
n4.do\_series(df)  
  
df.index = [df["у тому числі"]]  
df = df.iloc[:, 1:7]  
  
print('\n-------------------------------------------------------')  
print('-----------------Колекція DataFrame--------------------')  
print('-------------------------------------------------------\n')  
n4.do\_dataframe(df=df,num\_of\_cloumn=1,num\_of\_row\_start=1,num\_of\_row\_end=15)

num4.py

*#4*import pandas as pd  
  
def do\_series(df:pd.DataFrame):  
 print('-----------------Створення Series з індексами за замовчуванням--------------------')  
 print(df.head())  
  
 print('-----------------Звернення до елементів Series--------------------\n'  
 ' (Звернення до колонки)')  
 print(df['у тому числі'].head())  
  
 print('-----------------Обчислення описових статистик для Series--------------------')  
 saldo = df["Unnamed: 7"]  
 mean = saldo.mean()  
 print("mean", mean)  
 mode = saldo.mode()  
 print("mode", mode if mode.size < saldo.size else "no mode")  
 min = saldo.min()  
 print("min", min)  
 max = saldo.max()  
 print("max", max)  
 std = saldo.std()  
 print("std", std)  
  
 print('-----------------Створення колекції Series з нестандартними індексами--------------------')  
 df.index = [df["у тому числі"]]  
 df = pd.DataFrame(df.iloc[:, 1:])  
 df.columns = ['експ. тис. дол.', 'експ. у % до 9 міс.', 'експ. у % до заг.',  
 'імп. тис. дол.', 'імп. у % до 9 міс.', 'імп. у % до заг.',  
 'сальдо', 'region']  
 print(df.head())  
  
 print('-----------------Інші дії--------------------')  
 print('Тип даних: ',df.dtypes)  
 print('\nПовернення базової колекції:\n',df.values,'\n\n\n')  
  
def do\_dataframe(df:pd.DataFrame,num\_of\_cloumn:int,num\_of\_row\_start:int,num\_of\_row\_end:int):  
 print('-----------------Зернення до колонки--------------------')  
 print(df[f'Unnamed: {num\_of\_cloumn}'])  
 df=df.rename(columns={'Unnamed: 1':'|у тому числі|',  
 'Unnamed: 2':'|експорт, тис USD|',  
 'Unnamed: 3':'|експорт у % до 2022|',  
 'Unnamed: 4':'|експорт у % до загального|',   
 'Unnamed: 5':'|імпорт, тис USD|',  
 'Unnamed: 6':'|імпорт у % до 2022|',  
 'Unnamed: 7':'|імпорт % до загального|'})  
 print('------------------Звернення до стовпців-----------------------')  
 print(df.iloc[num\_of\_row\_start])  
 print('------------------Вібір підмножини-----------------------')  
 print(df.iloc[num\_of\_row\_start:num\_of\_row\_end])  
 print('------------------Логічне індексування-----------------------')  
 print(df[df>=45])  
 print('------------------Звернення до конкретного осередку[4,5]-----------------------')  
 print(df.iat[4,5])  
 print('------------------Описова статистика-----------------------')  
 print(df.describe())  
 print('------------------Траноспонування-----------------------')  
 print(df.T)  
 print('------------------Сортування рядків за індексами (спадання)-----------------------')  
 print(df.sort\_index(ascending=False))  
 print('------------------Сортування стовпців за індексами (спадання)-----------------------')  
 print(df.sort\_index(ascending=False,axis=1))

titanic.py

import pandas as pd  
import matplotlib.pyplot as plt  
  
*# 8*titanic = pd.read\_csv('https://vincentarelbundock.github.io/Rdatasets/csv/carData/TitanicSurvival.csv')  
  
*# 9*titanic.columns = ['name', 'survived', 'sex', 'age', 'class']  
  
print('-----------------Зчитані дані--------------------')  
print(titanic)  
  
*# 10*youngest = titanic.loc[titanic['age'].idxmin()]  
eldest = titanic.loc[titanic['age'].idxmax()]  
average = titanic['age'].mean()  
  
first\_class\_women = titanic[(titanic['sex'] == 'female') & (titanic['class'] == '1st')].sort\_values(by='name')  
youngest\_fcw = first\_class\_women.loc[first\_class\_women['age'].idxmin()]  
eldest\_fcw = first\_class\_women.loc[first\_class\_women['age'].idxmax()]  
count\_survived = first\_class\_women[first\_class\_women['survived'] == 'yes'].shape[0]  
  
print('-----------------youngest person--------------------')  
print(youngest)  
print('-----------------eldest person--------------------')  
print(eldest)  
print('-----------------average age--------------------')  
print(average)  
print('-----------------women of 1st class--------------------')  
print(first\_class\_women)  
print('-----------------youngest women of 1st class--------------------')  
print(youngest\_fcw)  
print('-----------------eldest women of 1st class--------------------')  
print(eldest\_fcw)  
print('---------------quantity of survived women of 1st class-------------')  
print(count\_survived)  
  
*#11*titanic.hist()  
plt.xlabel("Вік пасажирів")  
plt.ylabel("Частота")  
plt.show()

**Результат виконання програми**

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований описЗображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Рис. 1 – Вивід «очищених» даних із вхідного файлу та вивід статистичних даних по сальдо.

Зображення, що містить текст, знімок екрана, число, схема

Автоматично згенерований описРис. 2 – Гістограми по сальдо, експорту та імпорту

Далі слідує вивід для виконання 4 пункту завдання лабораторної роботи «Для цих даних проробити всі дії з пункту колекції Series і DataFrame бібліотеки pandas.»

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рис. 3 – Series з індексами за замовчуванням

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рис. 4 – Статистика описових даних та нестандартні індекси

Зображення, що містить текст, знімок екрана, Шрифт, документ

Автоматично згенерований опис

Рис. 5 – інші дії з колекції Series

Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис

Рис. 6 – Робота з колекцією DataFrame

Зображення, що містить текст, знімок екрана, меню, Шрифт

Автоматично згенерований опис

Рис. 7 – Звернення до стовпців та відбір підмножини

Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис

Рис. 8 – Логічне індексування

Зображення, що містить текст, знімок екрана, меню, Шрифт

Автоматично згенерований опис

Рис. 9 – Звернення до осередку, описова статистика та транспонування

Зображення, що містить текст, меню, знімок екрана, документ

Автоматично згенерований описЗображення, що містить текст, знімок екрана, меню, число

Автоматично згенерований опис

Рис. 10 – Сортування рядків

Далі слідує результат виконання програми для даних катастрофи Титаніка.

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рис. 11 – Зчитані дані

Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований описЗображення, що містить текст, знімок екрана, Шрифт, меню

Автоматично згенерований описЗображення, що містить текст, знімок екрана, квитанція, Шрифт

Автоматично згенерований опис

Рис. 12 – Проведення простого аналізу даних із умови завдання

Зображення, що містить схема, ряд, Графік, квадрат

Автоматично згенерований опис

Рис. 13 – Побудована гістограма віку пасажирів

**Висновок:** У результаті виконання лабораторної роботи №1 «Введення в data science» було засвоєно основи роботи з даними, включаючи їх збір, аналіз та візуалізацію, а також використання бібліотеки Python для ефективної обробки та представлення даних.

Були обрані та завантажені дані про зовнішню торгівлю України за січень–вересень 2023 року та набір даних катастрофи «Титаніка».

Проведено аналіз обраних даних, зокрема розраховано основні статистичні показники, такі як математичне сподівання, медіана, мода, дисперсія, середньоквадратичне відхилення.

Було візуалізовано дані за допомогою гістограм.

Отже, лабораторна робота дозволила вивчити та практично застосувати основні концепції та інструменти data science на конкретних даних, що відкриває шлях для подальших досліджень та аналізу в цій області, і, найголовніше, відкриває шлях до машинного навчання.