МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____Ю. П. Кондратенко
«____»_____2023 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ІНТЕЛЕКТУАЛЬНА СИСТЕМА АНАЛІЗУ СОЦІОЛОГІЧНИХ ДАНИХ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21910124

Виконав студент 4-го курсу, групи 401
_____ А. А. Стрельбицький
«___» червня 2023 р.

Керівник: канд. пед. наук, доцент
_____Н. М. Болюбаш
«___» червня 2023 р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р. техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____2022 р.

З А В Д А Н Н Я

на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Стрельбицькому Андрію Андрійовичу.

1.Тема кваліфікаційної роботи «Інтелектуальна система аналізу соціологічних даних».

Керівник роботи Болюбаш Надія Миколаївна, канд. пед. наук, доцент.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «__» ____ 202_ р. № _____

2. Строк представлення кваліфікаційної роботи студентом «__» _____ 2023 р.

3. Вхідні (початкові) дані до роботи: предметна сфера соціологічних досліджень, набір даних соціологічних опитувань для визначення індексу щастя.

Очікуваний результат: інтелектуальна інформаційна система аналізу соціологічних даних.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- дослідження теоретичних засад проведення соціологічних досліджень та методів опрацювання і аналізу соціологічної інформації;

- обґрунтування вибору інструментальних засобів розробки системи аналізу соціологічних даних;
- розробка і здійснення програмної реалізації інтелектуальної системи аналізу рівня щастя на підставі опрацювання даних соціологічних опитувань у різних країнах світу.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: Охорона праці ІТ-фахівців

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	А. О. Алексєєва канд. тенх. наук, доцент	

Керівник роботи __канд. пед. наук, доц. Болюбаш Н. М._____
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання ___Стрельбицький А. А._____
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання «_____»_____20__ р.

КАЛЕНДАРНИЙ ПЛАН
виконання бакалаврської кваліфікаційної роботи

Тема: Інтелектуальна система аналізу соціологічних даних

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Визначення керівника і теми БКР. Подання заяви на затвердження теми БКР	01.09.2022	1.11.2022	Виконано
2.	Отримання завдання на виконання БКР	25.11.2022	25.11.2022	Виконано
3.	Складання календарного плану	26.11.2022	10.12.2022	Виконано
4.	Огляд літератури за темою дослідження. Аналіз предметної сфери соціологічних опитувань, дослідження підходів до опрацювання та аналізу соціологічних даних	11.12.2021	31.12.2021	Виконано
5.	Вибір технологій та інструментальних засобів розробки системи	1.01.2023	31.01.2023	Виконано
6.	Створення дизайну, проєктування та програмна реалізація, тестування системи	1.02.2023	15.04.2023	Виконано
7.	Робота над розділами фахової частини БКР	16.04.2023	31.04.2023	Виконано
8.	Проходження переддипломної практики, збір та аналіз матеріалів, остаточне оформлення розділів фахової частини БКР	1.05.2023	14.05.2023	Виконано
9.	Розробка спеціальної частини з охорони праці	15.05.2023	25.05.2023	Виконано
10.	Обговорення отриманих результатів з керівником та попередній захист БКР	29.05.2023	1.06.2023	Виконано
11.	Корегування роботи за результатами попереднього захисту	2.06.2023	6.06.2023	Виконано
12.	Остаточне оформлення пояснювальної записки та слайдів доповіді до захисту	7.06.2023	11.06.2023	Виконано
13.	Подання рецензенту та рецензування БКР	15.06.2023	18.06.2023	Виконано
14.	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	19.06.2023	22.06.2023	Виконано
15.	Захист БКР перед ЕК	28.06.2023	28.06.2023	Виконано

Розробив студент _____Стрельбицький А.А.____ _____
(прізвище та ініціали) *(підпис)*

Керівник _____канд. пед. наук, доцент Болюбаш Н.М_____ _____
(наук. ступінь, вчене звання, прізвище та ініціали) *(підпис)*

« » 2022 р.

АНОТАЦІЯ

бакалаврської кваліфікаційної роботи студента групи 401 ЧНУ ім. Петра Могили

Стрельбицького Андрія Андрійовича

Тема: «Інтелектуальна система аналізу соціологічних даних»

Бакалаврська кваліфікаційна робота присвячена розробці та здійсненню програмної реалізації системи аналізу соціологічних даних. Що є актуальним в умовах високих темпів інформатизації сучасного суспільства, оскільки це вимагає застосування інноваційних підходів до засобів та методів обробки і аналізу соціологічної інформації при проведенні соціологічних досліджень.

Об’єкт роботи – процес обробки та аналізу соціологічної інформації.

Предмет роботи – програмні засоби для опрацювання даних соціологічних опитувань та методи класифікації і кластеризації даних.

Мета роботи – підвищення ефективності проведення соціологічних досліджень шляхом розробки інтелектуальної системи аналізу соціологічних даних із використанням алгоритмів класифікації та кластеризації.

Бакалаврська кваліфікаційна робота складається з фахової та спеціальної частини з охорони праці. Пояснювальна записка фахової частини складається зі вступу, трьох розділів, висновків та додатків. У першому розділі розкрито теоретичні засади проведення соціологічних досліджень та методи опрацювання і аналізу соціологічної інформації. У другому розділі обґрунтовано вибір інструментальних засобів розробки системи. У третьому розділі описано розробку та програмну реалізацію системи аналізу рівня щастя на підставі опрацювання даних соціологічних опитувань у різних країнах світу.

Бакалаврська кваліфікаційна робота містить 60 сторінок (без додатків), 21 рисунок, 1 таблиця, 24 джерела, 1 додаток.

Ключові слова: індекс щастя, соціологічна інформація, класифікація, кластерний аналіз соціологічних даних.

ABSTRACT

for bachelor's qualification work of a student of 401 group at Petro Mohyla Black Sea National University
Strelbytskoho Andriia Andriiovycha

Theme: «Intelligent system of analysis of sociological data»

The bachelor's thesis is devoted to the development and implementation of the software implementation of the sociological data analysis system. What is relevant in the conditions of high rates of informatization of modern society, as it requires the use of innovative approaches to the means and methods of processing and analyzing sociological information when conducting sociological research.

Object of work – the process of processing and analyzing sociological information.

Subject of work – software tools for processing sociological survey data and data classification and clustering methods.

The purpose of the work is to increasing the effectiveness of conducting sociological research by developing an intelligent system for analyzing sociological data using classification and clustering algorithms.

The bachelor's qualification work consists of a professional and special part on labor protection. The explanatory note of the professional part consists of an introduction, three sections, conclusions and appendices. In the first chapter, the theoretical principles of conducting sociological research and methods of processing and analyzing sociological information are revealed. In the second section, the choice of system development tools is substantiated. The third chapter describes the development, software implementation and application of the system for analyzing the level of happiness based on the processing of data from sociological surveys in different countries of the world.

The bachelor's thesis contains 60 pages (without appendices), 21 figures, 1 table, 24 sources, 1 appendix.

Keywords: Happy Index, sociological information, classification, cluster analysis of sociological data.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

GWP – Gallup World Poll

HLS – Happiness and Life Satisfaction

HPI – Happy Planet Index

KNN – k-Nearest Neighbors

WDI – World Development Indicators

GUI – Graphical User Interface

ВСТУП

Актуальність. В умовах бурхливих темпів інформатизації суспільства успіхи у розвитку соціологічної науки тісно пов’язані з розвитком інноваційний напрямів інтелектуального аналізу даних. Перспективним напрямком підвищення ефективності проведення соціологічних досліджень є застосування інтелектуальних систем аналізу соціологічних даних, отриманих при проведенні опитувань, які дозволяють автоматизовано здійснювати обробку та аналіз первинної соціологічної інформації.

Серед існуючих підходів до збирання та аналізу соціологічної інформації можна умовно виділити два основних підходи. Перший традиційний підхід передбачає розробку анкети та ручне проведення анкетування з подальшим введення та обробкою інформації за допомогою програмних засобів загального призначення, таких як Microsoft Excel та Google Sheets, або спеціалізоване програмне забезпечення (Statistica, SPSS, MatLab). Такий підхід є дуже трудомістким, не забезпечує оперативне коригування параметрів даних та їх необхідну якість, проведення попереднього аналізу даних опитування, достовірність операцій. Другий підхід передбачає проведення опитувань з урахуванням веб-систем організації (SurveyMonkey, Google Forms). При цьому цільовою аудиторією дослідника є користувачі інтернету, однак забезпечити якість, оперативність проведення опитування, репрезентативність вибірки досить важко. А етап аналізу даних необхідно виконувати в сторонній програмі.

Тому існує потреба розробки способів підвищення коректності результатів соціологічних досліджень на основі вдосконалення якості вимірювання емпіричної інформації через створення спеціальної автоматизованої інформаційної системи з вбудованими методами інтелектуального аналізу даних.

Це обумовило **мету роботи**, яка полягає у підвищенні ефективності проведення соціологічних досліджень шляхом розробки інтелектуальної системи аналізу соціологічних даних із використанням алгоритмів класифікації та кластеризації.

Відповідно до поставленої мети було сформульовано **завдання**:

1) дослідити теоретичні засади проведення соціологічних досліджень та методи опрацювання і аналізу соціологічної інформації;

2) обґрунтувати вибір інструментальних засобів розробки системи аналізу соціологічних даних;

3) розробити і здійснити програмну реалізацію інтелектуальної системи аналізу рівня щастя на підставі опрацювання даних соціологічних опитувань у різних країнах світу.

Об’єкт роботи – процес обробки та аналізу соціологічної інформації.

Предмет роботи – програмні засоби для опрацювання даних соціологічних опитувань та методи класифікації і кластеризації даних.

Методологічною основою дослідження є загальнонаукові та методи інтелектуального аналізу даних, які дозволили комплексно вивчити предмет та об’єкт дослідження, дослідити основні підходи до обробки первинної соціологічної інформації та її інтелектуального аналізу з застосуванням алгоритмів класифікації і кластеризації даних.

Практичне значення отриманих результатів полягає в тому, що використання розробленої інтелектуальної системи дозволить підвищити ефективність проведення соціологічних опитувань шляхом удосконалення обробки та аналізу соціологічної інформації.

Структура бакалаврської кваліфікаційної роботи. Відповідно до мети, завдань і предмета дослідження, бакалаврська робота містить основну та спеціальну частини. Основна частина роботи складається із вступу, трьох розділів, висновку, списку використаних джерел та 1 додатот. Загальний обсяг роботи – 69 сторінок, із них основного тексту основної частини – 60 сторінок, спеціальної – 9 сторінок. Кількість використаних джерел – 24.

1 ТЕОРЕТИЧНІ ЗАСАДИ АНАЛІЗУ СОЦІОЛОГІЧНИХ ДАНИХ

1.1 Організація та проведення соціологічних досліджень в умовах інформаційного суспільства

Соціологічне дослідження є системою послідовних, логічних методичних, методологічних та організаційних процедур, метою яких є отримання наукових знань про конкретний соціальний об'єкт. Проведення соціологічних досліджень дозволяє підтвердити здогади та домисли, зібрати та оцінити інформацію про явища, що вивчаються. Це свого роду сполучна ланка між реальною дійсністю та теоретичними знаннями, що дозволяє встановити нові закономірності розвитку структурних елементів чи суспільства загалом [1, 2].

За цілями соціологічні дослідження поділяються на прикладні та фундаментальні. Перші вивчають об’єкти, вирішують певні проблеми соціуму. Організація соціологічного дослідження фундаментального типу проводиться з метою визначення та аналізу суспільних тенденцій, закономірностей розвитку, вони покликані вирішувати складні проблеми суспільства.

За тривалістю виділяють такі види соціологічних досліджень:

1) експрес - від 1 тижня до 1 місяця;

2) короткострокові – 2-6 місяців;

3) середньострокові – 0,5-3 роки;

4) довгострокові – 3 і більше років.

За глибиною аналізу розрізняють наступні види соціологічних досліджень:

1) описові – дозволяють скласти цілісне уявлення про досліджувані процеси та явища. Проводяться відповідно до чіткої програми, об'єктом аналізу є переважно велика спільність людей з конкретними професійними, соціальними та демографічними характеристиками;

2) пошукові – вирішують найпростіші завдання. Використовуються найчастіше як попередній етап масштабного аналізу, у випадках, коли об'єкт, предмет чи програма аналізу є маловивченою або взагалі не вивчена;

3) аналітичні – описують соціальні явища та його компоненти, визначають механізми їх функціонування, причини виникнення. Підготовка, організація та проведення соціологічного дослідження даного типу потребує серйозних зусиль та професіоналізму з боку дослідника – він має вміти правильно інтерпретувати отриману складну інформацію, робити зважені висновки.

Проведення та організація досліджень із соціології здійснюється послідовно. Етапи аналізу, що змінюють один одного, є організаційно автономними і одночасно змістовно взаємопов’язаними:

1) підготовка до досліджень;

2) збір первинних соціологічних відомостей;

3) підготовка та обробка зібраних даних;

4) аналіз інформації, підбиття підсумків, формулювання рекомендацій та висновків.

Дослідження з соціології проводяться у певному середовищі суспільства та спрямовані на пошук справді існуючих фактів за принципом «зараз і тут». Кожен дослідницький захід є унікальним, проте деякі теоретичні узагальнення не виключені. Для отримання потрібних соціальних фактів дослідник повинен проникнути у певну соціальну реальність – роздати анкети всім респондентам, якщо це анкетне опитування, пояснивши зміст та мету вивчення, правила заповнення анкетних даних тощо. Експертне, телефонне опитування чи інтерв’ю вимагають безпосереднього спілкування дослідників та респондентів. «Проникнення» до об'єктів, що досліджуються, відбувається також у випадках використання інших методів збору соціальних фактів.

В умовах становлення інформаційного суспільства змінюються методи та засоби отримання і аналізу соціологічних даних. Це актуалізує необхідність аналізу нових, нещодавно виниклих проблем та пошуку способів їх вирішення. Формування єдиного цифрового простору може стати ключем до погіршення інтеграційних процесів у

суспільстві. При цьому, з одного боку, Інтернет став дуже важливим каналом поширення інформації, якому довіряє населення. З іншого – загострилася потреба у надійній верифікації одержуваної інформації, у можливості її критичного осмислення.

Для отримання об'єктивної оперативної та достовірної інформації про стан суспільства пропонується організувати своєчасний та об’єктивний моніторинг розвитку його структур. Для моніторингу пропонується використовувати дослідницький комплекс (автоматизовану інформаційну систему і пов’язану з нею методику соціологічних досліджень), призначений не тільки для дослідження соціального капіталу та інших структур суспільства, але і для формування необхідної культури мислення, науково обґрунтованого прийняття тих чи інших управлінських рішень. Ключовим завданням удосконалення системи вимірювання в соціології в даному контексті є розробка спеціальної автоматизованої інформаційної системи підвищення коректності результатів соціологічних досліджень на основі вдосконалення якості вимірювання емпіричної інформації.

З одного боку є необхідність посилення використання та розробки сучасних методів збору та аналізу соціологічної інформації, моделювання соціальних явищ, включаючи використання при зборі даних різних методів шкалування, вимірювання близькості, парних порівнянь, порівнянь у тріадах, а також різні програмні продукти, моделі та алгоритми аналізу текстів, інтелектуального аналізу даних та інші сучасні методи, напрацьовані у світовій та вітчизняній практиці.

З іншого боку швидкі темпи розвитку інформаційно-комунікаційних технологій та обчислювальних можливостей призвели до розвитку та широкого використання сучасних методів Data Mining, теорії штучного інтелекту і такої гілки науки, як комп’ютерна соціологія (англ. Computational Sociology) [3].

Соціологічні дослідження охоплюють різноманітні аспекти суспільного життя. У даній роботі розроблено інтелектуальну інформаційну систему аналізу рівня щастя. У сучасному суспільстві, що трансформується під впливом інформатизації та діджиталізації, зміни можуть відбуватися дуже швидко: змінюються як культурні норми, так і рівень

економічного благополуччя. Дані зміни ведуть до того, що змінюється рівень щастя членів різних спільнот у різних країнах світу.

На даний момент дослідженням рівня щастя займаються різноманітні великі організації, використовуючи різні методи відслідковування та вимірювання рівня щастя. Одним із таких методів є визначення показника The Legatum Prosperity Index – індексу, що розраховується інститутом Легатума, на підставі даних опитувань інституту Геллапа, індикатора світового розвитку, міжнародного союзу електрозв'язку, рейтингу недієздатності держав, світових показників управління, даних ВООЗ, Freedom house, WVS, Centre for Systemic Peace. Індекс розраховується за показниками: розвиток економіки, розвиток бізнесу, якість управління, освіта, здоров'я, безпека та особиста свобода, соціальний капітал, довкілля. Ще одним методом визначення рівня щастя у світі та країнах є Happiness and Life Satisfaction (авторів Ортіц-Оспіни Е., Розера М.). Вони розраховують кілька показників: суб'єктивну задоволеність життям та кількість людей, які називають себе щасливими. Велику увагу дані дослідники приділяють динаміці даних показників у різних країнах. Ще одне велике дослідження рівня щастя у світі зветься Happy Planet Index. Цей показник розраховується за допомогою формули:

$$HPI = \frac{тривал.життя \cdot суб'єкт.благополуччя \cdot соціальна нерівн.}{стан екології} \qquad (1.1)$$

Більшість країн, які знаходяться на високих позиціях в інших рейтингах щастя, в даному рейтингу знаходяться не на найвищих позиціях, тому що незважаючи на високий показник суб'єктивного благополуччя, низький показник соціальної нерівності та високу тривалість життя в цій країні може бути вкрай забрудненим екологічним середовищем.

Динаміка змін рівня щасті у різних країнах та регіонах є різною, що обумовлено різним рівнем економічного, екологічного, соціального розвитку країн. На рисунках 1.1 та 1.2 зображено динаміку зміни рівня щастя у США та Індії з використанням інтегрального індексу World Happiness Report за період з 2013 по 2019 роки.

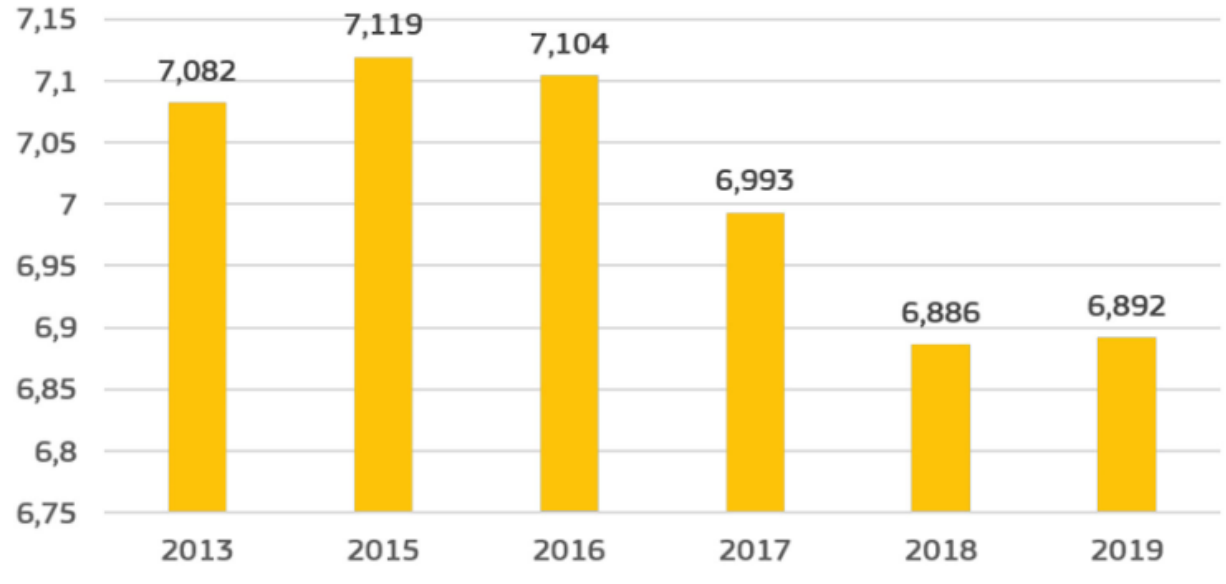


Рисунок 1.1 – Динаміка змін індексу щастя для США

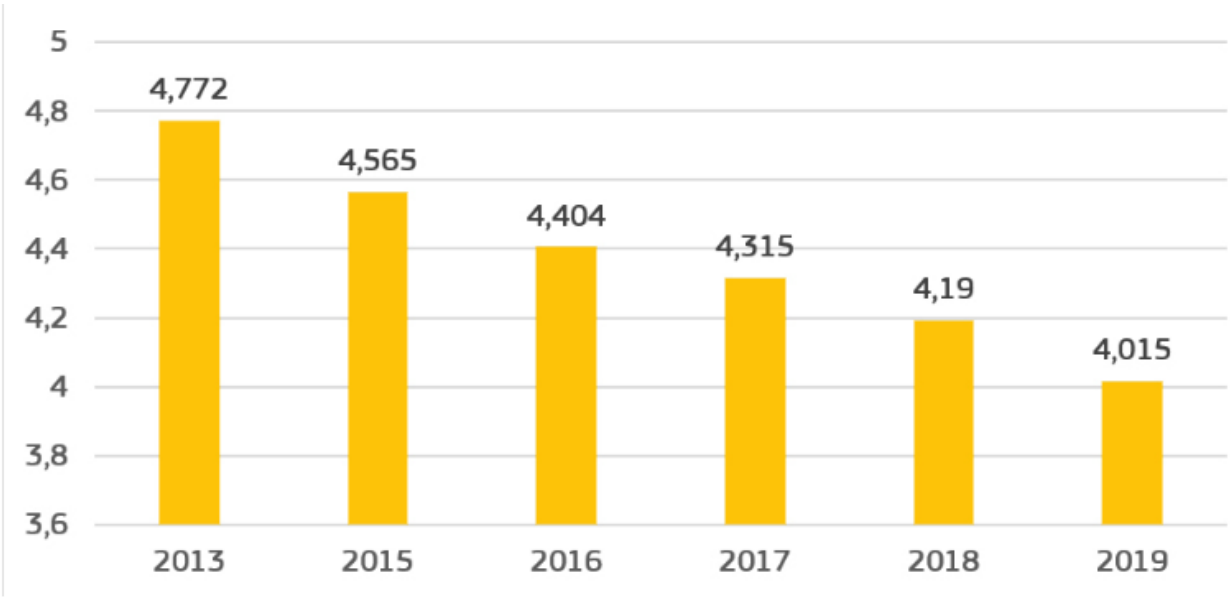


Рисунок 1.1 – Динаміка змін індексу щастя для Індії

World Happiness Report – Всесвітня доповідь про щастя є найбільш відомою і популярною доповіддю підрозділу UN Sustainable Development Solution Network (ООН з пошуку рішень стабільного розвитку). У цій доповіді, на підставі даних опитувань інституту Геллапа та статистичних матеріалах, виводиться інтегральний показник рівня щастя у певній країні чи регіоні. Для його обчислення використовуються ВВП на душу населення, очікувана тривалість здорового життя, свобода у виборі життєвого шляху,

щедрість/довіра, рівень сприйняття корупції, рівень вираження позитивних та негативних емоцій.

Серед існуючих підходів до збирання та аналізу соціологічної інформації можна умовно виділити два основних підходи. Опишемо їх більш детально.

Проведення традиційним методом на основі розробки анкети, ручного проведення анкетування, введення та обробки інформації за допомогою програмних засобів. Як правило, на етапі обробки використовуються або широкодоступні програмні засоби загального призначення (такі як табличні пакети з рядом математичних функцій Microsoft Excel, Google Sheets), або спеціалізоване програмне забезпечення (Statistica, SPSS, MatLab). За такого підходу неможливо забезпечити оперативне коригування параметрів вибірки, забезпечити необхідну якість вибірки, проводити попередній аналіз даних опитування, забезпечити достовірність операцій, проведених інтерв'юером. Зазначимо також, що утруднено забезпечення відсутності помилок під час введення даних, а сама операція введення є дуже трудомісткою.

Проведення з урахуванням веб-систем організації опитувань (SurveyMonkey, Google Forms). При цьому цільовою аудиторією дослідника є користувачі інтернету (що суттєво погіршує якість вибірки), проте забезпечити якість, оперативність проведення опитування, репрезентативність вибірки досить важко. Крім того, етап аналізу даних необхідно виконувати в сторонній програмі.

Таким чином, існує потреба розробки способів підвищення коректності результатів соціологічних досліджень на основі вдосконалення якості обробки емпіричної інформації через створення спеціальної автоматизованої інформаційної системи з вбудованими методами інтелектуального аналізу даних.

1.2 Опрацювання та аналіз первинної соціологічної інформації

Опрацювання й аналіз зібраної первинної соціологічної інформації полягає в кількісній оцінці впливу різних чинників на розвиток соціальних процесів у різни сферах суспільства. Первинну соціологічну інформацію можна опрацьовувати з

використанням різних методів статистики та інтелектуального аналізу даних: кластеризації, класифікації, регресивного, кореляційного, факторного аналізу [4, 5]. Опрацьована інформація може бути подана в таблицях, графіках, діаграмах, рисунках, схемах, які дають змогу інтерпретувати зібрані дані, аналізувати й виявляти певні залежності, робити висновки, розробляти рекомендації.

Однак опрацювання соціологічної інформації можливе за умови кількісного вимірювання ознак досліджуваного явища. Тому опрацювання первинної статистичної інформації передбачає перетворення категоріальних даних у числові. Соціологічне вимірювання є процедурою, за допомогою якої якісні ознаки соціального явища чи об’єкта, що вивчається, порівнюють з певним еталоном і отримують числове значення. Еталоном виміру є шкала, яку створює сам соціолог у процесі дослідження. Надання кількісної визначеності якісним ознакам, що вивчаються, називають шкалуванням. За допомогою шкалування якісно різнорідні соціальні ознаки приводять до порівнянних кількісних показників.

У процесі шкалування спочатку виявляють зовнішні ознаки досліджуваного явища (об’єкта), тобто ті властивості й характеристики, які підлягають спостереженню й вимірюванню. Кожна ознака характеризується певною сукупністю змінних. Показники варіювання кожної з цих змінних є індикаторами – доступними для спостереження й вимірювання характеристиками.

Для того щоб вибрати індикатор, слід попередньо здійснити інтерпретацію та операціоналізацію досліджуваного явища. Якщо інтерпретація дає змогу визначити напрям аналізу, за яким має здійснюватися збирання кількісної інформації, а операціоналізація – види цієї інформації, то визначення індикаторів допомагає вибрати способи й форми її збирання. Крім того, індикатори дають можливість правильно сформулювати інструментарій дослідження, визначити структури відповідей на поставлені запитання. Кожний індикатор має певні характеристики, які в інструментарії є варіантами відповідей на запитання. Розміщені у певній послідовності, вони утворюють шкалу вимірювання. Комбінація кількох індикаторів утворює індекс. Індекс можна виразити за

допомогою простої або зваженої середньої арифметичної значень кожного варіанта відповідей у шкалі чи з допомогою різниці між високими і низькими, позитивними й негативними виявами ознаки.

Шкалування в сукупності з індексацією утворюють процедуру, що називається в соціології квантифікацією. Квантифікація – це процедура вимірювання і кількісного вираження якісних ознак і відносин соціальних об’єктів. Наприклад, вимір у балах позитивних емоцій, задоволення тощо.

Щоб побудувати шкалу вимірювання, спочатку знаходять діапазон судження, тобто визначають крайні значення вияву певної ознаки (максимум і мінімум), які називають встановленням його континууму (тривалості). Потім континуум розбивають на частини. Це процедура встановлення дрібності, чи градуювання шкали. Словесне формулювання здогадних (можливих) відповідей на запитання є лише точками на безперервній шкалі суджень. Якщо вимірюється ознака явища, відображена операціональним поняттям «задоволення», то позиціями шкали вимірювання можуть бути характеристики такого суб’єктивного індикатора, як «міра задоволення»: «частково задоволений», «більше задоволений, ніж незадоволений», «зовсім незадоволений», «скоріше незадоволений, ніж задоволений», «незадоволений».

Кількість градацій визначає так звану чутливість шкали — здатність її виявляти ставлення респондента до різних аспектів досліджуваного соціального явища з відповідною мірою диференціації.

Отже, завдяки шкалуванню з’являється можливість не тільки фіксувати наявність або брак якісної ознаки, а й виміряти її, тобто оцінити ступінь її вияву. Відтак шкалування виконує три функції: класифікації, ранжирування і запровадження метрики – вимірювання інтенсивності вияву соціальних ознак, що вивчаються, визначення різниці такої інтенсивності. Зупинимося більш детально на характеристиці шкал вимірювання.

1. Номінальна шкала містить перелік характеристик об’єкта чи явища, що інколи взаємно виключають одна одну. За допомогою цих шкал вимірюють такі

об’єктивні ознаки, як стать, національність, сімейний стан, вік, стаж роботи, кваліфікація, а також суб’єктивне ставлення респондентів до певних аспектів соціального явища, процесу.

2. Рангова шкала, або шкала порядку, утворюється за допомогою кумуляції (додавання, накопичення) на підставі упорядкування шкали назв. Рангові шкали дають змогу впорядковувати властивості, що вивчаються, від найбільш до найменш значущої чи навпаки. Рангова шкала у загальному вигляді може бути подана так:

а) максимально заперечна (негативна) відповідь;

б) заперечна відповідь;

с) радше заперечна, ніж ствердна (позитивна) відповідь;

д) нейтральна відповідь;

е) радше ствердна, ніж заперечна відповідь;

є) ствердна відповідь;

ж) максимально ствердна відповідь.

Більшість закритих запитань з різних анкет, є, по суті, ранговими шкалами.

3. Інтервальна (метрична) шкала утворюється на основі рангової наданням певної кількості балів кожній позиції. Приклад такої шкали:

а) зовсім не задовольняє (– 1);

б) радше не задовольняє, ніж задовольняє (– 0,5);

с) важко відповісти (0);

д) радше задовольняє, ніж не задовольняє (+ 0,5);

е) повністю задовольняє (+1).

На відміну від рангової, інтервальна шкала дає змогу не тільки впорядкувати вияви властивості, що вивчається, а й розрахувати різницю між окремими позиціями шкали, тобто визначити інтервали.

Отже, опрацювання й аналіз первинної соціологічної інформації гуртуються на наданні згідно з певними правилами числових значень якісним характеристикам явищ, процесів і об’єктів, що досліджуються. Ця операція здійснюється на підставі встановленої (внаслідок інтерпретації та операціоналізації) системи показників у

вигляді емпіричних індикаторів і математичних індексів. Вона і дістала назву соціологічного вимірювання.

Якісні й кількісні характеристики, що використовуються при цьому, відтворюють структуру й динаміку досліджуваних соціальних явищ, процесів і об'єктів, утворюють складну систему соціальних показників. Вони складаються з індикаторів, що фіксують якісні сторони (наявність чи відсутність ознаки), та індексів чи коефіцієнтів, які фіксують кількісні сторони (інтенсивність вияву ознаки).

Ефективність дослідження залежить не від обсягу зібраної інформації, а від глибини та всебічності аналізу. Аналіз розпочинається з перевірки інструментарію на точність, повноту та якість заповнення. Перевірка на точність заповнення передбачає виявлення помилок у відповідях та їх корекцію. Після того як проведено перевірку інструментарію і ту частину його, що не відповідає названим критеріям, вилучено, проводиться кодування інформації, тобто її формалізація. Процедура кодування полягає у наданні кожному варіанту відповідей певних умовних чисел - кодів. Унаслідок кодування вся інформація перетворюється на числову систему. Кодування розпочинається ще на початку дослідження, коли певні коди надаються варіантам відповідей, що їх закладено в самому інструменті у закритих та напівзакритих питаннях.

Кодування відповідей на відкриті питання відбувається після опитування. Для цього виписуються всі відповіді, визначається частота повторення відповідей певного змісту, проводиться їх класифікація, тобто відповіді зводять у певні смислові групи, варіанти і розробляють їх формалізований список — кодифікатор, за допомогою якого кодуються різні варіанти відповідей. При цьому використовується дві системи кодування:

1) порядкова з наскрізною нумерацією всіх позицій;

2) позиційна з нумерацією варіантів лише в межах одного запитання.

Узагальнення інформації розпочинається з групування респондентів за одним вибраним показником. Отримані таким чином однорідні за складом групи

стають об’єктом аналізу. Використання комбінаційного групування (розподіл респондентів за двома і більше показниками) дає змогу поглибити аналіз. Залежно від завдань дослідження таке групування може бути структурним, типологічним або аналітичним. Структурне групування – це класифікація за певним показником, що об’єктивно властивий усій сукупності даних; типологічне – за показником, створеним самим дослідником, або суб’єктивним за своєю природою, аналітичне – має місце, коли групування проводиться за двома чи більше показниками з метою визначення їхньої взаємодії. Таким чином отримують атрибутивні та варіаційні ряди розподілу, до яких застосовують методи статистичного аналізу:

1) аналіз середніх величин, дисперсії, коефіцієнта кореляції та спряженості;

2) аналіз відсоткового розподілу, графічний аналіз розподілу;

3) аналіз кореляційної залежності кількісних і якісних ознак;

4) змістова інтерпретація даних, визначення залежностей причинно-наслідкових зв’язків.

У процесі опрацювання й аналізу первинної соціологічної інформації інтегральної шкали індекси є показниками інтенсивності вияву ознаки чи рівня розвитку процесу, що визначаються за певною шкалою. Вивчаючи статистичні зв’язки соціолог натрапляє на різні співвідношення між фактором і результативною ознакою. Інтенсивний розвиток соціологічної науки потребує розширення методичного та інформаційного забезпечення соціологічних досліджень.

1.3 Методи інтелектуального аналізу соціологічних даних

У сучасному технізованому світі розвиток соціологічної науки тісно пов’язаний із використанням нових інформаційних технологій [6]. Успіхи в галузі штучного інтелекту визначили вектор подальшого технологічного розвитку, ініціювали процес інтелектуалізації технічних засобів, що, зокрема, призвело до становлення нової парадигми пізнання – інтелектуального аналізу даних

(англ. Data Mining). Основна ідея Data Mining полягає в тому, що дані зберігають інформацію, невидиму під звичним кутом зору.

Data Mining – це не один, а ціла сукупність різних методів виявлення закономірностей, гіпотез, шаблонів та знань: пошуку асоціативних правил, класифікації, прогнозування, кластеризації тощо [7]. Вибір конкретного методу залежить від типу даних та від того, яку інформацію потрібно отримати.

Перед початком роботи з алгоритмами та методами Data Mining потрібно зрозуміти, які потрібні дані, чи є потреба у їх попередній обробці та яка саме. Зараз у мережі можна знайти вже готові набори соціологічних даних, вони зазвичай непідготовлені та мають викиди, пропуски, проблемні дані. Знайдений набір даних повинен бути застосований у майбутньому, разом із новими даними, з достатньо високим ступенем достовірності.

Для подальшого соціологічного дослідження набору даних, отриманих у результаті опитування стосовно рівня щастя у різних кранах світу, доцільно виявити внутрішню структуру даних, застосувавши методи кластеризації. Виявивши кластери споріднених за рівнем щастя країн світу та проаналізувати розподіл країн світу за рівнем щастя, доцільно побудувати класифікатор, який дозволяє класифікувати нові країни.

Застосування методів класифікації та кластеризації потребує визначення мір близькості між об’єктами, які можуть бути визначені як для числових шкал, так і для категоріальних [8]. Для даних, представлених числовими метричними ознаками частіше усього у якості міри несхожості використовують наступні міри:

1) відстань Евкліда:

$$d_E(x_i, x_j) = \sqrt{\sum_{t=1}^{m} (x_{it} - x_{jt})^2}, \qquad (1.1)$$

2) квадрат відстані Евкліда:

$$d_E^2(x_i, x_j) = \sum_{t=1}^{m} (x_{it} - x_{jt})^2, \qquad (1.2)$$

де x_i та x_j – *i*-й та *j*–й об'єкти набору даних,

x_{it} та x_{jt} – ознака *t* для *i*-го та *j*–го об'єктів набору даних,

m – кількість змінних.

Для даних, представлених категоріальними атрибутами, у якості міри несхожості об'єктів x_i та x_j використовують:

1) відстань Хеммінга, яка дорівнює кількості атрибутів, значення яких для об'єктів відрізняються й є метрикою;

2) відсоток незгоди, який розраховують як відношення кількості неспівпадаючих ознак до загальної кількості категоріальних ознак:

$$d(x_i, x_j) = \frac{l}{m}, \tag{1.3}$$

де m – загальна кількість категоріальних ознак;

l – кількість не співпадаючих ознак, для яких $x_{ik} \neq x_{jk}$, $k \in \{1,.2,..,m\}$.

Перед застосуванням методів кластеризації та класифікації ознаки набору даних соціологічних опитувань необхідно нормалізувати – привести до одного числового діапазону.

Серед методів кластеризації для виявлення груп споріднених країн за даними соціологічного опитування було вирішено використати алгоритм k-means [9]. Дія цього алгоритму спрямована на мінімізацію цільової функції:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n_j} d^2(x_{ij}, c_j), \tag{1.4}$$

де k – кількість кластерів,

c_j – j –й кластер,

x_{ij} – i-й об'єкт j –го кластера,

n_j – кількість об'єктів у j –му кластері,

$d(x_{ij}, c_j)$ – відстань між i-м об'єктом j –го кластера та його центром ваги.

Центром ваги кластера є точка у просторі ознак із координатами, які є середніми арифметичними значеннями відповідних ознак країн, що входять до кластеру.

Алгоритм k-means є ітераційним із такими етапами:

1) задається значення k – кількості кластерів, на які необхідно розбити набір даних;

2) рандомно обираються k початкових центрів ваги кластерів;

3) розраховують відстані від усіх об’єктів до центрів ваги кластерів;

4) відносять кожен об’єкт до кластера з найближчим центром ваги;

5) перераховують центри ваги кластерів відповідно до їх поточного складу;

6) повертаються до пункту 3 та повторюють пункти 4 і 5 до тих пір, поки не буде переміщення об’єктів із кластера в кластер – на цьому роботу алгоритму припиняють.

До основних методів класифікації, які можна застосувати для аналізу соціологічних даних, відносять наступні: метод опорних векторів, метод k-найближчих сусідів, Naïve Bayes, дерева рішень та інші. Для побудови класифікатора країн за рівнем щастя було вирішено використати алгоритм k-найближчих сусідів.

Метод k-найближчих сусідів (k-Nearest Neighbors, KNN) – популярний алгоритм класифікації, один із найзрозуміліших підходів до класифікації. Алгоритм здатний виділити серед усіх об’єктів k найближчих сусідів, схожих на новий об’єкт. На основі класів найближчих сусідів виноситься рішення відносно класу, до якого буде належати новий об’єкт.

Для класифікації нового об'єкту за алгоритмом k-найближчих сусідів необхідно [10]:

1) задати число k – кількість найближчих сусідів;

2) обрати метод розрахунку близькості (відстані між об’єктами) та розрахувати відстань до кожного з об'єктів навчаючої множини;

3) відібрати k об'єктів, відстань до яких є мінімальною;

4) класом нового об’єкта є клас, що найчастіше зустрічається серед найближчих сусідів (при простому незваженому голосуванні) або клас, який набрав найбільшу кількість голосів (при зваженому голосуванні).

Слабким місцем даного алгоритму є підбір коефіцієнта k - кількості об’єктів, які будуть вважатися схожими (сусідами). Якщо прийняти k = 1, то алгоритм втратить узагальнюючу здатність (тобто здатність видавати правильний результат для даних, що не зустрічалися раніше), новому запису буде привласнений клас найближчий до нього. Якщо встановити занадто велике значення, то багато локальних особливостей не будуть виявлені.

Переваги алгоритму kNN є наступними:

1) алгоритм стійкий до аномальних викидів, тому що ймовірність влучення такого запису в число к-найближчих сусідів мала. Якщо ж це відбулося, то вплив на голосування (особливо зважене) (при k>2) також буде незначним;

2) програмна реалізація алгоритму відносно проста;

3) результат роботи алгоритму легко піддається інтерпретації. Експертам у різних областях цілком зрозуміла логіка роботи алгоритму, заснована на знаходженні схожих об’єктів;

4) можливість модифікації алгоритму, шляхом використання найбільш підходящих функцій сполучення й метрик дозволяє підбудувати алгоритм під конкретне завдання.

Недоліки алгоритму kNN:

1) набір даних, використовуваний для алгоритму, повинен бути репрезентативним.

2) модель не можна «відокремити» від даних: для класифікації нового об’єкта потрібно використати всі приклади – ця особливість сильно обмежує використання алгоритму.

1.6 Постановка задачі

Здійснивши аналіз предметної сфери досліджень у галузі соціології та основних підходів до обробки соціологічної інформації і методів її аналізу, було зроблено висновок про необхідність розробки інтелектуальної системи аналізу соціологічної інформації з використанням алгоритмів кластеризації та класифікації даних.

Об’єктом роботи є процес обробки та аналізу соціологічної інформації.

Предметом роботи є програмні засоби для опрацювання даних соціологічних опитувань та методи класифікації і кластеризації даних.

Метою роботи є підвищення ефективності проведення соціологічних досліджень шляхом розробки інтелектуальної системи аналізу соціологічних даних із використанням алгоритмів класифікації та кластеризації.

Досягнення поставленої мети обумовлює необхідність вирішення наступних завдань:

1) дослідити теоретичні засади проведення соціологічних досліджень та методи опрацювання і аналізу соціологічної інформації;

2) обґрунтувати вибір інструментальних засобів розробки системи аналізу соціологічних даних;

3) розробити і здійснити програмну реалізацію інтелектуальної системи аналізу рівня щастя на підставі опрацювання даних соціологічних опитувань у різних країнах світу.

Поставлені завдання визначають основну функціональність розроблюваної системи, яка повинна надавати можливість для:

1) завантаження даних опитування із файлу формату csv;

2) перегляду даних користувачем;

3) здійснення нормалізації завантажених даних;

4) проведення кластеризації нормалізованих даних за алгоритмом k-means;

5) перегляд користувачем характеристик країн, які потрапили до кожного кластеру, та середніх значень ознак кожного кластеру;

6) на основі здійсненої кластеризації з урахуванням розподілу країн на групи споріднених об’єктів побудову класифікатора за алгоритмом k-найближчих сусідів та класифікацію нових об’єктів.

Висновки до розділу 1

У даному розділі бакалаврської кваліфікаційної роботи було здійснено аналіз предметної області – сфери соціологічних досліджень, розглянуто основні підходи до аналізу рівня щастя у різних країнах світу, визначено особливості аналізу соціологічних даних. було обґрунтовано вибір методів аналізу соціологічних даних для здійснення кластеризації та класифікації – алгоритмів k-means та k–найближчих сусідів. Розкрито основні етапи обраних методів та здійснено постановку задачі.

2 ІНСТРУМЕНТАЛЬНІ ЗАСОБИ РОЗРОБКИ СИСТЕМИ АНАЛІЗУ СОЦІОЛОГІЧНИХ ДАНИХ

2.1 Мова програмування Python

Python – це дуже популярна, широко використовувана високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробки і читання коду [11]. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий набір корисних функцій для здійснення аналізу. Бібліотеки, такі як Tensorflow від Google, роблять Python дуже цікавою мовою для роботи в області машинного навчання.

Python підтримує структурне, узагальнене, об’єктно-орієнтоване, функціональне і аспектно-орієнтоване програмування. Основні архітектурні риси: динамічна типізація, автоматичне керування пам’яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопотокових обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об’єднуватися в пакети.

Багата стандартна бібліотека є однією з привабливих сторін Python. Тут є засоби для роботи з багатьма мережевими протоколами і форматами Інтернету, наприклад, модулі для написання HTTP-серверів і клієнтів, для розбору і створення поштових повідомлень, для роботи з XML тощо. Набір модулів для роботи з операційною системою дозволяє писати крос-платформні додатки. Існують модулі для роботи з регулярними виразами, текстовими кодуваннями, мультимедійними форматами, криптографічними протоколами, архівами, серіалізації даних, підтримка юніт-тестування.

Але найважливіше, що має ця мова – це робота з великим обсягом даних, яка розкриває усі особливості напряму Data Mining.

Розглянемо основні переваги цієї мови програмування:

1) такі програмні пакети як pandas [12, 13], scikit-learn [14] і Tensorflow, роблять Python надійним варіантом для сучасних додатків в області машинного навчання;

2) Python дуже проста у вивченні, низький поріг входження робить його ідеальною першою мовою для тих, хто займається програмуванням;

3) Python має великий набір спеціально розроблених модулів і широко використовується розробниками, багато онлайн-сервісів надають API для Python;

4) є масштабованою мовою програмування, тому що вона забезпечує поліпшену структуру для підтримки великих програм, ніж shell-скріпти.

Однак є наступні недоліки:

1) типобезпека: Python є динамічно типізованою мовою, помилки невідповідності типів (наприклад, передача рядка (string) в якості аргументу методу, який очікує ціле число (integer)) можуть час від часу траплятися;

2) в разі якщо є конкретні цілі аналізу даних та статистичного аналізу, то великий набір пакетів мови R дає їй перевагу перед Python;

3) існують більш швидкі та безпечні альтернативи Python серед мов програмування.

Проведений аналіз дозволяє зробити висновок, що мова програмування Python є хорошим варіантом для інтелектуального аналізу соціологічних даних.

Суттєва частина при розробці проєкту для аналізу соціологічної інформації зосереджена навколо процесу ETL (витяг-перетворення-завантаження). Наявність вбудованих функцій для здійснення нормалізації й реалізації алгоритмів k-means та k–найближчих сусідів надає зручні можливості для створення інтелектуальної системи аналізу соціологічних даних [14, 18, 20]. Завдяки вбудованим бібліотекам можна розробити гарно спроєктований графічний інтерфейс для візуалізації результатів та реалізації класифікації та проведення кластерного аналізу соціологічних даних.

2.2 Бібліотеки Python для інтелектуального аналізу даних

Для програмної реалізації алгоритмів інтелектуального аналізу даних при програмуванні на мові Python широко використовують наступні бібліотеки: Pandas, Matplotlib, SciPy, NumPy, PyQt5, Math, Sklearn. Щоб зрозуміти, які функції та переваги мають дані бібліотеки і яку користь вони мають для вирішення поставленої задачі, розглянемо їх докладніше [16, 24, 22].

Найголовнішою є бібліотека Pandas – бібліотека, написана для мови програмування Python для обробки та аналізу даних [13, 15, 19]. Зокрема, вона пропонує структури даних та операції для маніпулювання з числовими таблицями та часовими рядами. Це безкоштовне програмне забезпечення, випущене за ліцензією BSD. Назва походить від терміну «дані панелі» – терміну для наборів даних, які включають спостереження протягом декількох періодів часу для одних і тих самих об’єктів. Pandas дозволяє імпортувати дані з різних форматів файлів, таких як значення, розділені комами, JSON, SQL, Microsoft Excel. Pandas дозволяє проводити різні операції з обробкою даних, такі як злиття, переформатування, вибір, а також функції очищення даних та переміщення даних. що є цінним при завантаженні даних та їх попередній обробці.

Бібліотека Matplotlib дозволяє здійснювати візуалізацію даних для Python та її числового математичного розширення NumPy [17, 21]. Вона надає об'єктно-орієнтований API для побудови двовимірних та тривимірних графіків у застосунках із використанням наборів інструментів загального користувацького інтерфейсу, таких як Tkinter, wxPython, Qt або GTK +. Існує також процедурний інтерфейс «pylab», заснований на машині стану (наприклад, OpenGL), розроблений так, щоб дуже нагадувати інтерфейс MatLab, хоча його використання не рекомендується. Python Matplotlib є альтернатива модуля візуалізації програми для технічних обчислень MatLab. Бібліотека SciPy використовує Matplotlib.

Бібліотека NumPy — це бібліотека Python, яка використовується для роботи зі спеціальним структурами - багатомірними масивами. Вона також має функції для

роботи в області лінійної алгебри, перетворення Фур'є та матриць. NumPy означає Numerical Python [18]. У Python є списки, які служать для цілей масивів, але вони повільно обробляються. NumPy має на меті надати об'єкт масиву, який у 50 разів швидший за традиційні списки Python. Масиви часто використовуються в Data Science, де швидкість і ресурси дуже важливі.

SciPy – це бібліотека Python з відкритим кодом, яка містить модулі для оптимізації, лінійної алгебри, інтеграції, інтерполяції, спеціальних функцій, обробки сигналів та зображень, вирішувачів ODE та інших завдань, поширених у науці та техніці. SciPy побудована на базі NumPy та розширює його можливості і використовує Matplotlib для візуалізації даних.

Бібліотека SciPy містить набір пакетів з алгоритмами кластерного аналізу, дозволяє завантажувати дані, редагувати, зберігати у файлі. SciPy підтримує стандарти MatLab, що дає можливість працювати зі стеками технологій.

Математична бібліотека Math дозволяє здійснювати в Python широкий функціонал з числовими даними, розв'язувати складні рівняння, містить набір функцій для виконання математичних, тригонометричних, логарифмічних операцій.

Бібліотека Sklearn (або Scikit-Learn) – це безкоштовна бібліотека Python, яка найчастіше використовується при проведенні інтелектуального аналізу даних [14, 23]. Вона містить функції та алгоритми для класифікації, прогнозування, регресії та кластеризації, включаючи метод опорних векторів, випадковий ліс, k найближчих сусідів, k-means та DBSCAN, і призначена для взаємодії з числовими та науковими бібліотеками Python NumPy та SciPy.

2.3 Середовище розробки PyCharm

PyCharm є кросплатформенним інтегрованим середовищем професійної розробки проєктів на Python, розроблене компанією JetBrains на основе IntelliJ IDEA (рис. 2.1). PyCharm має повнофункціональний візуальний налагоджувач,

дозволяє контролювати якість коду з допомогою перевірок відповідності вимогам PEP8, розумних рефакторингів та багато інспекцій, а також надає допомогу при тестуванні. Працює під операційними системами Windows, Mac OS X і Linux, підтримує фреймворки Django, web2py та Flask.

PyCharm є комерційним продуктом, але має обмежену безкоштовну версію, яку також можна використовувати для розробки проєктів (див. табл. 2.1).

Таблиця 2.1 – Версії IDE PyCharm

Функціональні можливості	**PyCharm Professional Edition**	**PyCharm Community Edition**
Функціональний редактор Python	+	+
Інструмент запуску текстів та графічний налагоджувач	+	+
Навігація по коду та рефракторинги	+	+
Інстпекція коду	+	+
Підтримка систем контролю версій	+	+
Інструменти для наукових очислень	+	-
Веброзробка	+	-
Вебфреймворки Python	+	-
Python-профілювальник	+	-
Можливість віддаленої розробки	+	-
Підтримка баз даних та SQL	+	-

До недоліків середовища можна віднести його ресурсомісткість. Однак PyCharm має багато переваг, оскільки дозволяє розробникам:

1) здійснювати налагодження та автозаповнення коду;

2) запускати тести;
3) профілювати код та знаходити вузькі місця;
4) працювати зі службами контролю версій;
5) иконувати автоматичне розгортання;
6) здійснювати віддалену розробка;
7) працювати з базами даних;
8) реалізувати налаштовуваний інтерфейс;
9) розробляти кросплатформенні застосунки;
10) розширювати функціонал за допомогою плагінів.

PyCharm має зручний інтерфес (див. рис. 2.1, рис. 2.2). Крім Python PyCharm підтримує такі мови програмування, як Jython, HTML, XML, JSON, YAML, XSL, XPath и Markdown. Використовуючи плагіни, можна також установити Rust і Dart. У бекенді – підтримує Cython та SQL, у фронтенді – працює з JavaScript, TypeScript, CSS, SASS, SCSS, Less. Як плагіни доступні Haml, Slim, Liquid.

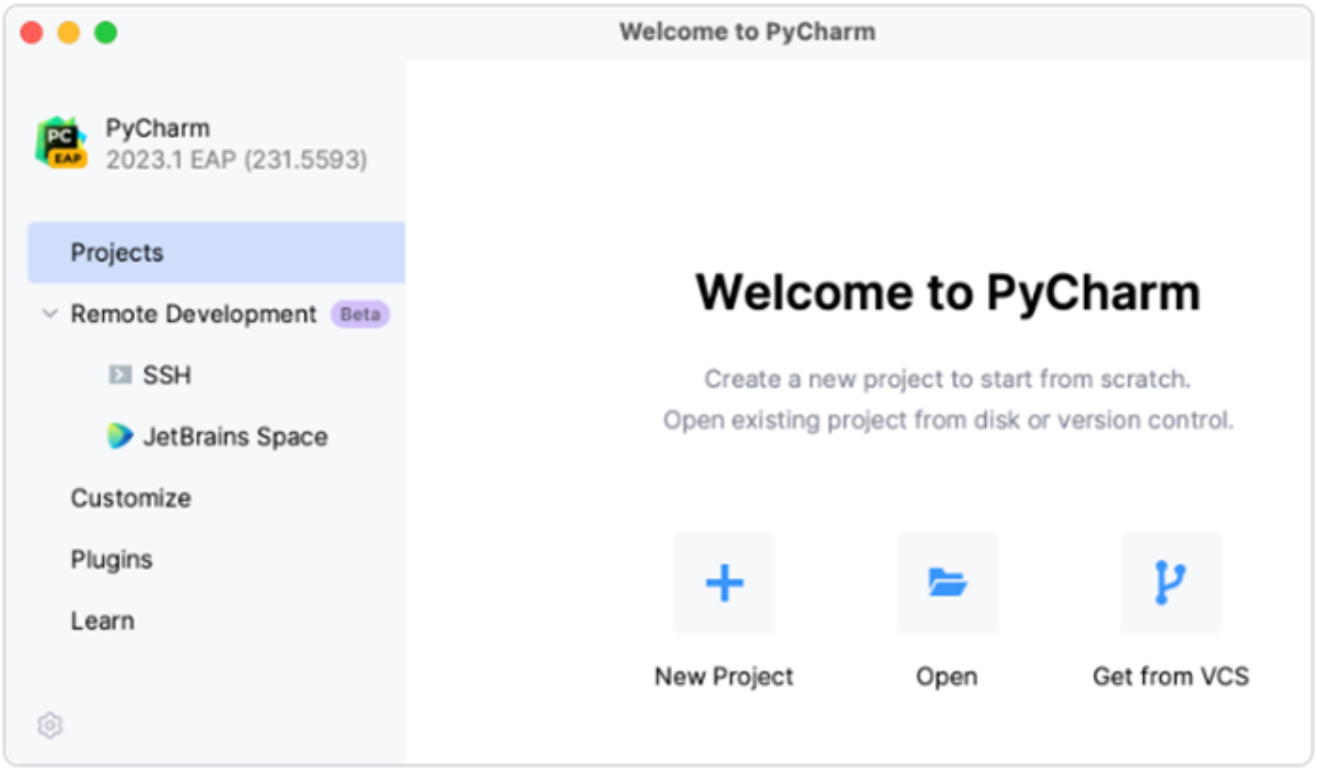


Рисунок 2.1 – Вікно PyCharm для створення нового проєкту

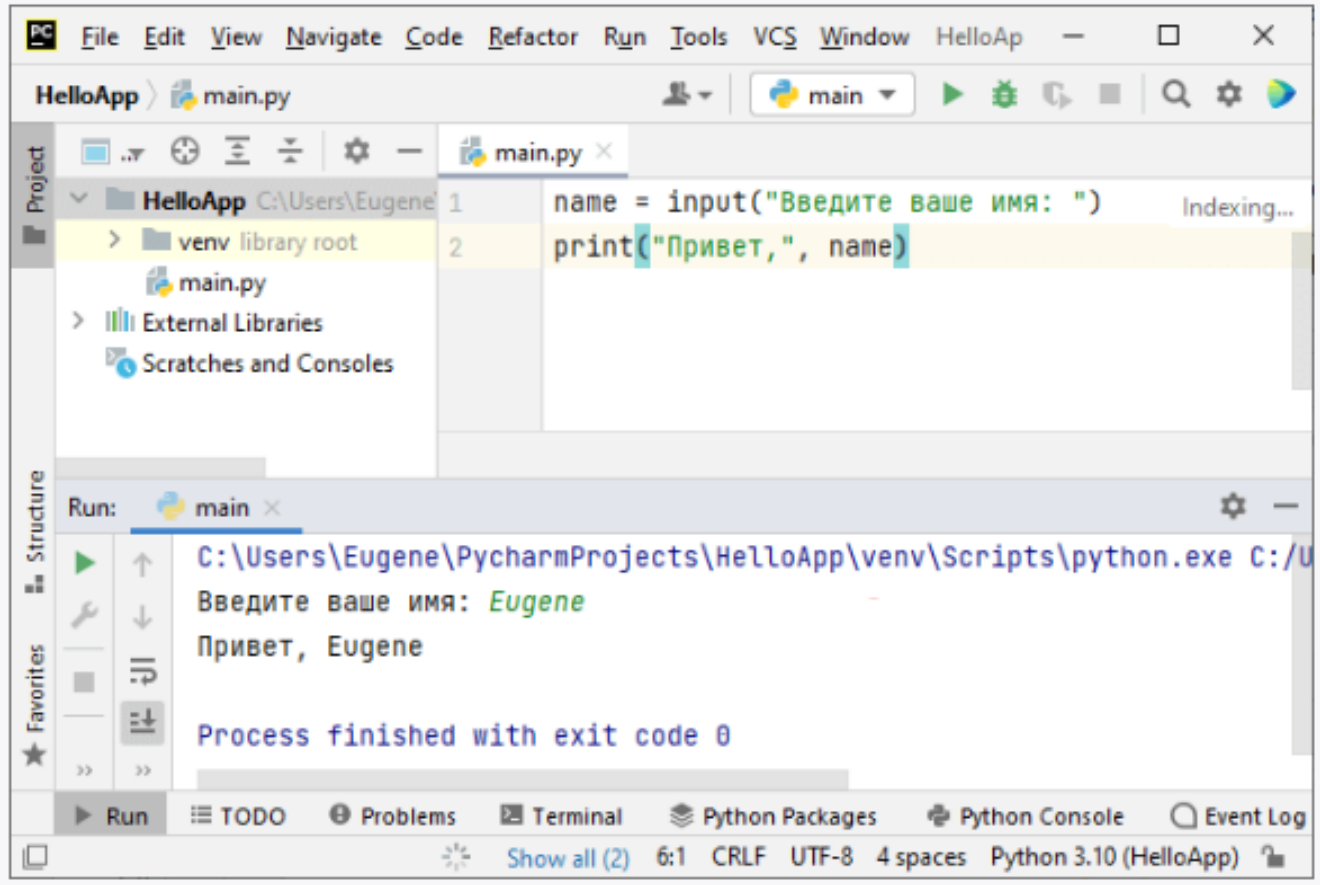


Рисунок 2.2 – Інтерфейс середовища розробки PyCharm

Здійснений аналіз дозволив обрати середовище для створення інтелектуальної системи аналізу соціологічних даних.

2.4 Фреймворк Qt

Qt є фреймворком для розробки кросплатформеного програмного забезпечення на мові програмування C++. Для мови програмування Python є бібліотека PyQt, яка дозволяє використовувати переваги Qt.

Qt використовується більш тисячами розробників провідних компаній у всьому світі для створення кросплатформених застосунків і користувацьких інтерфейсів. Qt дозволяє запускати написане з його допомогою програмне забезпечення в більшості сучасних операційних систем шляхом простої компіляції програм для кожної системи без зміни вихідного коду. Включає в себе всі основні

класи, які можуть знадобитися при розробці прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу і закінчуючи класами для роботи з мережею, базами даних і XML. Являється повністю об'єктно-орієнтованим, розширюваним і підтримуючим технологію компонентного програмування.

Відмінна особливість – використання метаоб'єктного компілятора – попередньої системи обробки вихідного коду. Розширення можливостей забезпечується системою плагінів, які можна розмістити безпосередньо в панелі візуального редактора. Також існує можливість розширення звичайної функціональності віджетів, пов'язаної з розміщенням їх на екрані, відображенням, перерисовкою при зміні розмірів вікна.

Ключовими перевагами версії Qt5, яка була обрана для розробки системи, є наступні:

1) покращена якість графіки;

2) покращена продуктивність на апаратних платформах з обмеженим функціоналом;

3) кросплатформена переносимість;

4) підтримка C++ 11;

5) підтримка HTML5 в QtWebKit2;

6) значно покращений рух QML з новим API;

7) простота використання та сумісність із попередніми версіями Qt4.

Можливість Qt надавати друковану графіку була покращена з можливістю використання OpenGL ES (версія графічного API, спеціально розроблена для встановлюваних систем і мобільних пристроїв). Це полегшує розробку і виконання додатків з багатою графікою як з елементами анімації і переходів, так і з плавним рендерингом 2D і 3D анімації, при цьому – хоч на високопродуктивних системах, хоч на пристроях із відносно обмеженою продуктивністю (таких як мобільні телефони, планшети та недорогі платформи для розробки).

Кросплатформена переносимість стала ще простішою в Qt5 завдяки новій модульності, розділяючи модулі на необхідні (essentials) і додаткові (add-on), що дозволяє зменшити розмір застосунків. Використання QPA (Qt Platform Abstraction) також забезпечує кросплатформенну переносимість, що дозволяє розробляти застосунок для розвертання на багатьох кінцевих платформах.

Qt підтримує всі основні настільні операційні системи – Windows, Mac OS X і Linux, вбудовані операційні системи – такі як Embedded Linux, Windows Embedded, а також найбільш широко розповсюджені операційні системи реального часу (операційні системи реального часу, RTOS) для встановлюваних пристроїв - VxWorks, Neutrino і INTEGRITY.

Використання вбудованого браузера руху Qt WebKit2 дозволяє легко інтегрувати веб-контент і додатки. Це дозволяє розробникам, які використовують HTML5, розробляти гібридні додатки, що поєднують відгук/швидкість і потужний нативний код із великою кількістю динамічного контенту.

Міграція застосунків, розроблених із використанням попередньої версії Qt – Qt4 – дуже проста і вимагає лише простої перекомпіляції застосунків.

PyQt5 – це набір Python-зв'язків для фреймворку Qt5 від Digia. Набір PyQt5 доступний для Python 2.x та Python 3.x. PyQt5 реалізовано як комплект Python-модулів. Він включає близько 620 класів і 6000 функцій і методів. Це кросплатформений інструментарій, який запускається на більшості операційних систем, серед яких Unix, Windows та MacOS. PyQt5 реалізований під двома ліцензіями. Розробники можуть вибрати між GPL та комерційною ліцензією.

Висновки до розділу 2

Для розробки системи інтелектуального аналізу соціологічних даних обґрунтовано використання кросплатформеного інтегрованого середовища розробки PyCharm, яке є потужним інструментом для розробки застосунків, підтримує обрані для розробки мови програмування, надає зручний інтерфейс, має

розширену функціональність, редактор коду та інші корисні інструменти для розробників.

Розробка проєкту здійснювалася на мові програмування Python із використанням бібліотек Pandas, Matplotlib, SciPy, NumPy, PyQt5, Math, Sklearn. Бібліотека Pandas використовувалася для імпортування даних з різних форматів файлів та проведення злиття, переформатування, очищення даних. Бібліотека Matplotlib та побудована на базі неї SciPy використовувалися для візуалізації даних, їх редагування та зберігання. Для реалізації основних алгоритмів кластерного аналізу даних та класифікації застосовувалася бібліотека Sklearn.

3 РОЗРОБКА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АНАЛІЗУ СОЦІОЛОГІЧНИХ ДАНИХ

3.1 Data Set результатів соціологічного опитування щодо рівня щастя

Світовий рейтинг щастя фокусується на соціальному, міському та природному середовищі. Зокрема, рейтинг спирається на самозвіти жителів про те, як вони оцінюють якість життя, яку вони відчувають на даний момент, що включає три основні моменти: оцінка поточного життя, оцінка очікуваного майбутнього життя, позитивний і негативний вплив (емоції).

Для проведення аналізу було використано Dataset – Happiness.csv, який містить дані соціологічного опитування стосовно рівня щастя зі 152 країн світу. Із вказаного набору даних було обрано шість змінних, аналіз яких дозволяє пояснити сприйняття людьми якості свого життя: Log GDP per capita, Healthy life expectancy, Social support, Freedom to make life choices, Generosity, Perceptions of corruption. Дамо більш детальну характеристику цим ознакам.

1. Log GDP per capita – ВВП на душу населення виражено в паритеті купівельної спроможності (ПКС), взятому з Індикаторів світового розвитку (WDI) за 2019 рік. Дана ознака містить натуральний логарифм ВВП на душу населення, оскільки ця форма значно краще відповідає даним, ніж ВВП на душу населення.

2. Healthy life expectancy – очікувана тривалість здорового життя при народженні. Значення даної ознаки розраховане на основі даних від Всесвітньої організації охорони здоров’я.

3. Social support – соціальна підтримка, є розрахованим для кожної країни середнім значенням двійкових відповідей (0 = ні, 1 = так) на запитання опитування Gallup World Poll (GWP): «Якщо ви потрапили в біду, чи є у вас родичі чи друзі, на допомогу яких ви можете розраховувати, коли вони потрібні, чи ні?».

4. Freedom to make life choices – свобода робити життєвий вибір, є розрахованим для кожної країни середнім значенням бінарних відповідей (0 = ні, 1

= так) на запитання опитування GWP: «Чи задоволені ви чи не задоволені своєю свободою вибирати, що робити зі своїм життям?».

5. Generosity – щедрість, розраховують як залишок регресії середнього національного значення відповідей на запитання опитування GWP: «Чи жертвували ви гроші на благодійність за останній місяць?» на ВВП на душу населення.

6. Perceptions of corruption – сприйняття корупції, розраховують для кожної країни як середнє значення двійкових відповідей на два запитання GWP: «Чи поширена корупція в уряді чи ні?» та «Чи поширена корупція в бізнесі чи ні?» Якщо дані про корупцію в уряді відсутні, сприйняття корупції в бізнесі використовується як загальний показник сприйняття корупції.

7. Positive affect – позитивний афект, визначається як середнє значення відповідей на запитання, чи відчували ви під час подій вчорашнього дня емоції сміху, насолоди, щастя.

8. Negative affect – негативний афект, визначається як середнє значення відповідей на запитання, чи відчували ви під час подій вчорашнього дня емоції хвилювання, смутку, гніву.

3.2 Розробка інтерфейсу застосунку

Розробку застосунку для аналізу соціологічних даних країн світу здійснено у на мові програмування Python із використанням фреймворку Qt5 від Digia, який дозволяє створювати кросплатформені застосунки та користувацький інтерфейс. Для мови програмування Python є бібліотека PyQt, яка дозволяє використовувати переваги фреймворку Qt5.

Застосунок потребує графічного зображення, для реалізації якого було використано GUI (Graphical User Interface). Головною перевагою GUI є те, що ці системи, доступні для людей всіх рівнів знань, від абсолютного новачка до просунутого розробника або інших технологів. Вони дозволяють простим

користувачам відкривати меню, переміщати файли, запускати програми або шукати в Інтернеті, не вказуючи комп'ютера функції для виконання через командний рядок.

Розглянемо основні можливості, які надає модуль PyQt5 для розробки інтерфейсу застосунку.

1. Віджет (QWidget): цей елемент в модулі стосується прямокутної області на екрані дисплея користувача (див. рис. 3.1).

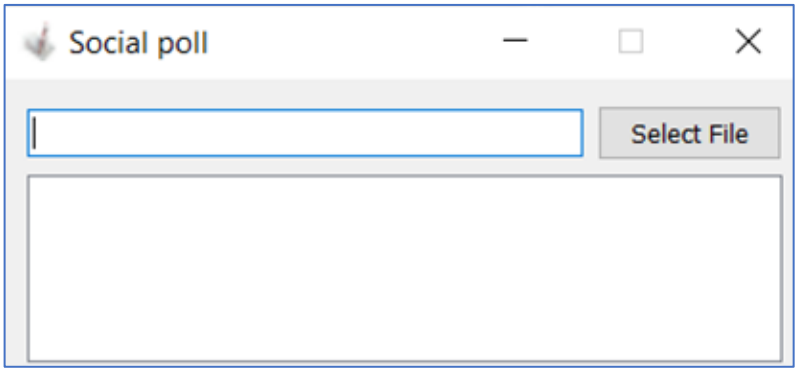


Рисунок 3.1 – Розробка стандартного вікна застосунку в PyQt5

2. Кнопка QPushButton в PyQt API є об’єктом класу QPushButton і представляє кнопку, при натисканні на яку можна запрограмувати виклик певної функції (див. рис. 3.2).

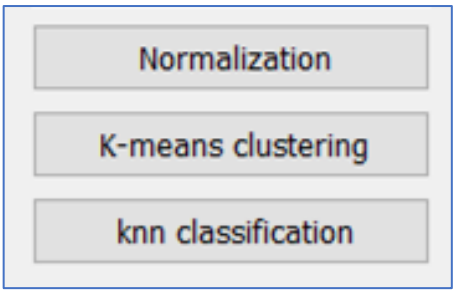


Рисунок 3.2 – Створення кнопок застосунку в PyQt5

3. Вікно повідомлень QMessageBox використовується у програмі з графічним інтерфейсом користувача, щоб надати необхідну інформацію для користувача або

попросити користувача виконати дії на основі повідомлення. Для будь-якої програми з графічним інтерфейсом можна створити чотири типи вікон повідомлень (див. рис. 3.3): вікно інформаційних повідомлень, вікно попереджень, вікно критичних повідомлень та вікно повідомлень із запитаннями.

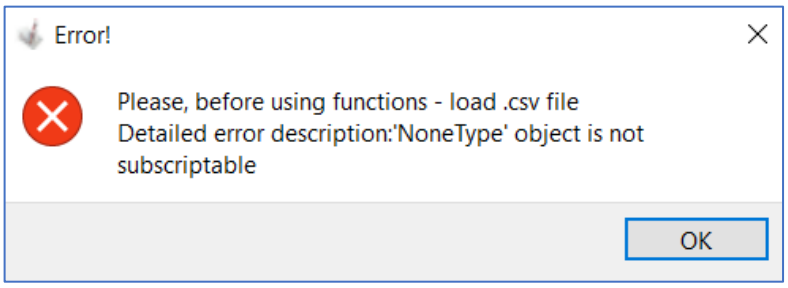


Рисунок 3.3 – Приклад вікна повідомлення QMessageBox

4. Діалогове вікно QFileDialog є діалоговим вікном вибору файлів. Це дає змогу користувачеві переміщатися файловою системою та обирати файл для відкриття або збереження даних. Діалогове вікно викликається або за допомогою статичних функцій, або викликом функції exec() для об’єкта діалогу.

Статичні функції класу QFileDialog getOpenFileName() і getSaveFileName() викликають діалогове вікно власного файлу поточної операційної системи. Фільтр файлів також можна застосувати для відображення лише файлів із зазначеними розширеннями. Також можна встановити початковий каталог і назву файлу за замовчуванням.

Після запуску вікна застосунку Social Poll на екрані відображається призначений для користувача інтерфейс, який показаний на рис. 3.4. У верхній частині вікна програми знаходиться вкладка Select File (див. рис. 3.4., маркер 1), яка призначена для завантаження набору соціологічних даних у форматі .csv, які необхідно проаналізувати з використанням алгоритмів кластеризації та класифікації. Відразу під вкладкою Select File розташована панель для відображення даних, які будуть вивантажені з файлу, обраного користувачем. (див. рис. 3.4., маркер 2).

Рисунок 3.4 – Інтерфейс вікна застосунку

Наступна панель містить кнопки, які виконують головні функції програми, пов’язані з поставленою задачею аналізу даних. Кнопка Normalization запускає процедуру нормалізації, яка здійснює перетворення значень кожної ознаки набору даних до одного числового діапазону – від 0 до 1 (див. рис. 3.4., маркер 3)

Кнопка K-means clustering запускає ітеративний алгоритм кластеризації, заснований на мінімізації сумарних квадратичних відхилень точок кластерів від центроїдів (середніх координат) цих кластерів. (див. рис. 3.4., маркер 4)

Кнопка knn classification (див. рис. 3.4., маркер 5) відкриває вікно для вибору мір близькості, які було використано у даному проекті – відстані Евкліда та квадрату відстані Евкліда (див. рис. 3.5).

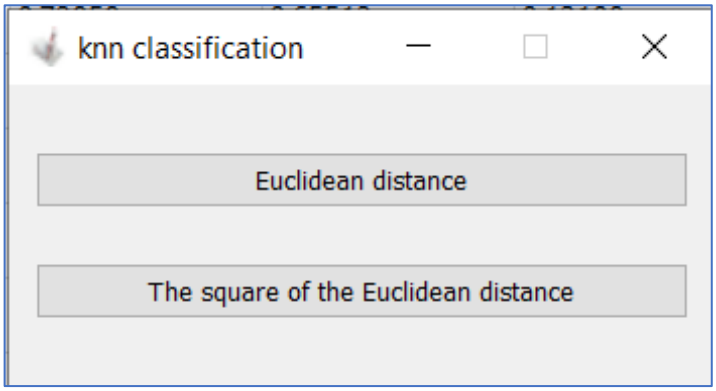


Рисунок 3.5 – Вікно для вибору мір близькості

Для початку роботи треба завантажити файл .csv формату, для цього натискається вкладка Файл та обирається пункт Open file. Після цих дій користувача відкривається провідник, який показує допустимі файли, які можна обрати – файли формату .csv (див. рис. 3.6).

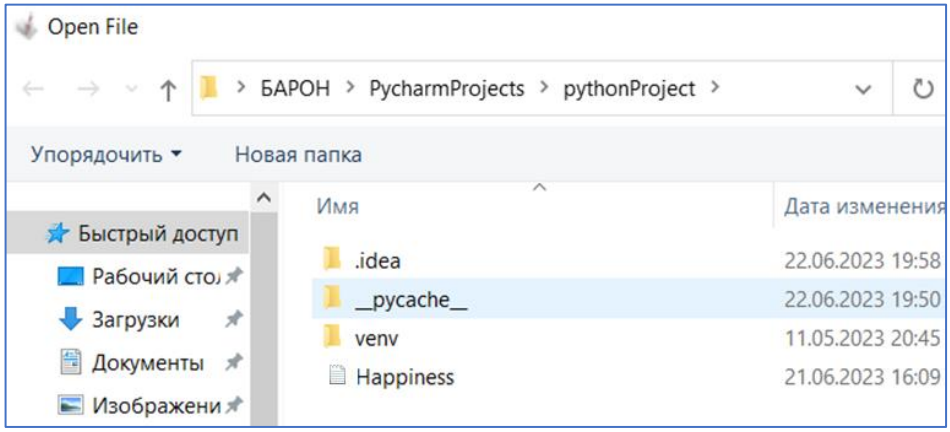


Рисунок 3.6 – Завантаження файлу з даними

Обравши файл та відкривши його усі дані відображаються у центральній частині вікна програми (див. рис.3.7).

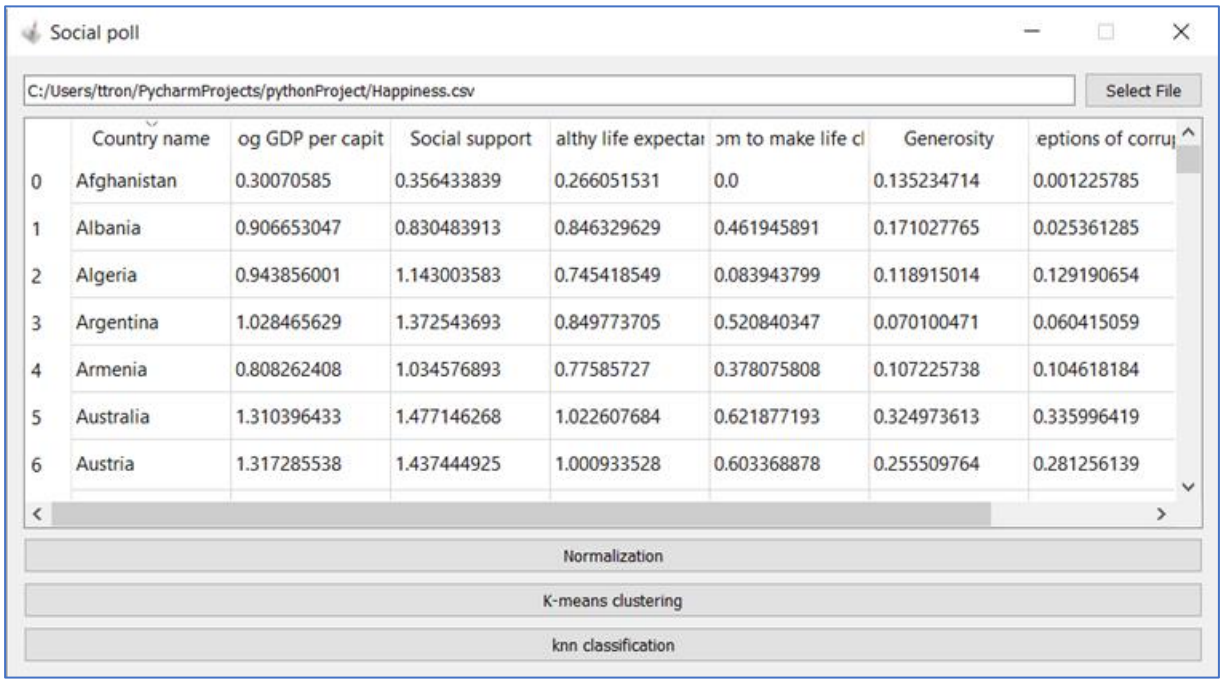


Рисунок 3.7 Завантажений для аналізу набір даних

Після завершення розробки застосунку було здійснено – контроль за якістю розробленого проєкту, тобто перевірка відповідності між тим, що програма повинна робити і що вона реально робить. У системі передбачено перевірка наявності помилок і вивід повідомлень про це. Якщо користувач відкриє вікно для вибору файлу та закриє його, не обравши файл, виводиться повідомлення, зображене на рис. 3.8. провідник та зачинить його не вибравши файл.

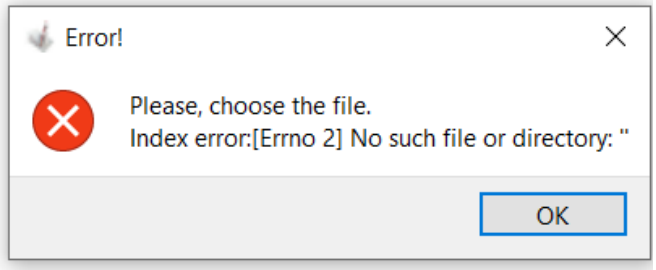


Рисунок 3.8 – Повідомлення про не відкритий файл

Якщо користувач не завантажить набір даних і натисне на кнопки для здійснення аналізу даних, буде виведене вікно з повідомленням, зображене на рис. 3.9.

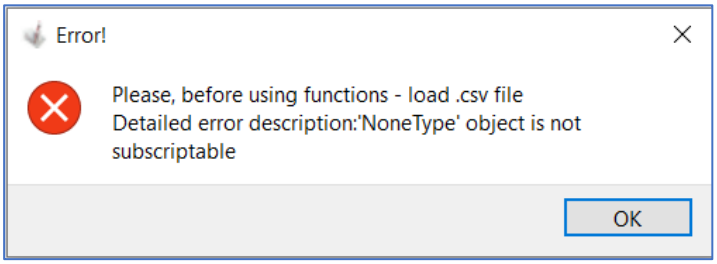


Рисунок 3.9 – Повідомлення при спробі використати функції без завантажених даних.

Якщо користувач почне використовувати методи кластеризації та класифікації без нормалізації, буде виведене вікно з повідомленням, зображене на рис. 3.10.

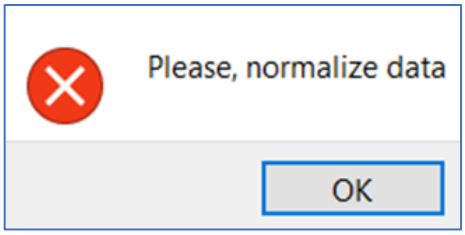


Рисунок 3.10 – Повідомлення при спробі використати функції без нормалізації

3.3 Програмна реалізація алгоритмів інтелектуального аналізу даних

Перейдемо до опису реалізації алгоритмів кластеризації та класифікації.

Фрагмент коду, за допомогою якого Dataset із соціологічними даними зчитується з файлу та відображається у вікні програми у вигляді таблиці наведено нижче:

```
fileName, _ = QtWidgets.QFileDialog.getOpenFileName(self, "Open File", "",
"CSV Files (*.csv)");
        self.pathLE.setText(fileName)
        self.df_table = pd.read_csv(fileName)
        model = PandasModel(self.df_table)
        self.pandasTv.setModel(model)
```

Першим, шо потрібно зробити, це нормалізувати дані. У коді нормалізація здійснюється з використанням функції norm():

```
def norm(table, colum):
   table[colum] = table[colum].apply(
      lambda v: round((v - table[colum].min()) / (table[colum].max() -
table[colum].min()), 9))
```

Кластеризація k-середніх – один із найбільш широко використовуваних алгоритмів неконтрольованого машинного навчання, що формує кластери даних на основі подібності між екземплярами даних. Для цього конкретного алгоритму необхідно заздалегідь визначити кількість кластерів – k.

Для створення трьох кластерів за алгоритмом k-means було написано наступний скрипт:

```
kmeans = KMeans(n_clusters=3, random_state=0, n_init="auto")
kmeans.fit(data)
```

У першому рядку створено об’єкт KMeans і передано йому 3 як значення параметра n_clusters. Далі викликано метод fit для k-means та передано дані для кластеризації.

Метод fit() реалізується кожним оцінювачем і приймає вхідні дані для зразкових даних (X), а для контрольованих моделей він також приймає аргумент для міток (тобто цільових даних y). За бажанням він також може приймати додаткові властивості.

Методи fit() зазвичай відповідають за численні операції. Як правило, вони повинні почати з очищення всіх атрибутів, які вже зберігаються, а потім виконати перевірку параметрів і даних. Вони також відповідають за оцінку атрибутів із вхідних даних, зберігають атрибути моделі та, нарешті, повертають підібрану оцінку.

Для класифікації даних методом k-ближніх сусідів було обрано дві метрики: відстань Евкліда та квадрат відстані Евкліда. Їх реалізація має такий вигляд для даних, що мають дві ознаки:

1) math.sqrt((a[0] - b[0]) ** 2 + (a[1] - b[1]) ** 2)

2) (a[0] - b[0]) ** 2 + (a[1] - b[1]) ** 2

У нашому випадку таких ознак шість, тому рівняння було розкладено на основні частини такі як:

```
def euclidean_distance(a, b):
    return (a - b) ** 2
```

Далі цей фрагмент коду проходить через цикл та формує масив, у якому дані сумують. Після цього ми отримуємо результат квадрату відстані Евкліда. Фрагмент коду наведено нижче.

```
    for row_country_id in range(len(table["Country name"])):
        temp = []
        for data_col_name in data_col_names:
            first = table[data_col_name][row_country_id]
            second = table[data_col_name][country_row_num]
            euclidean_result = euclidean_distance(first, second)
            temp.append(euclidean_result)
        formula_result = sum(temp)
```

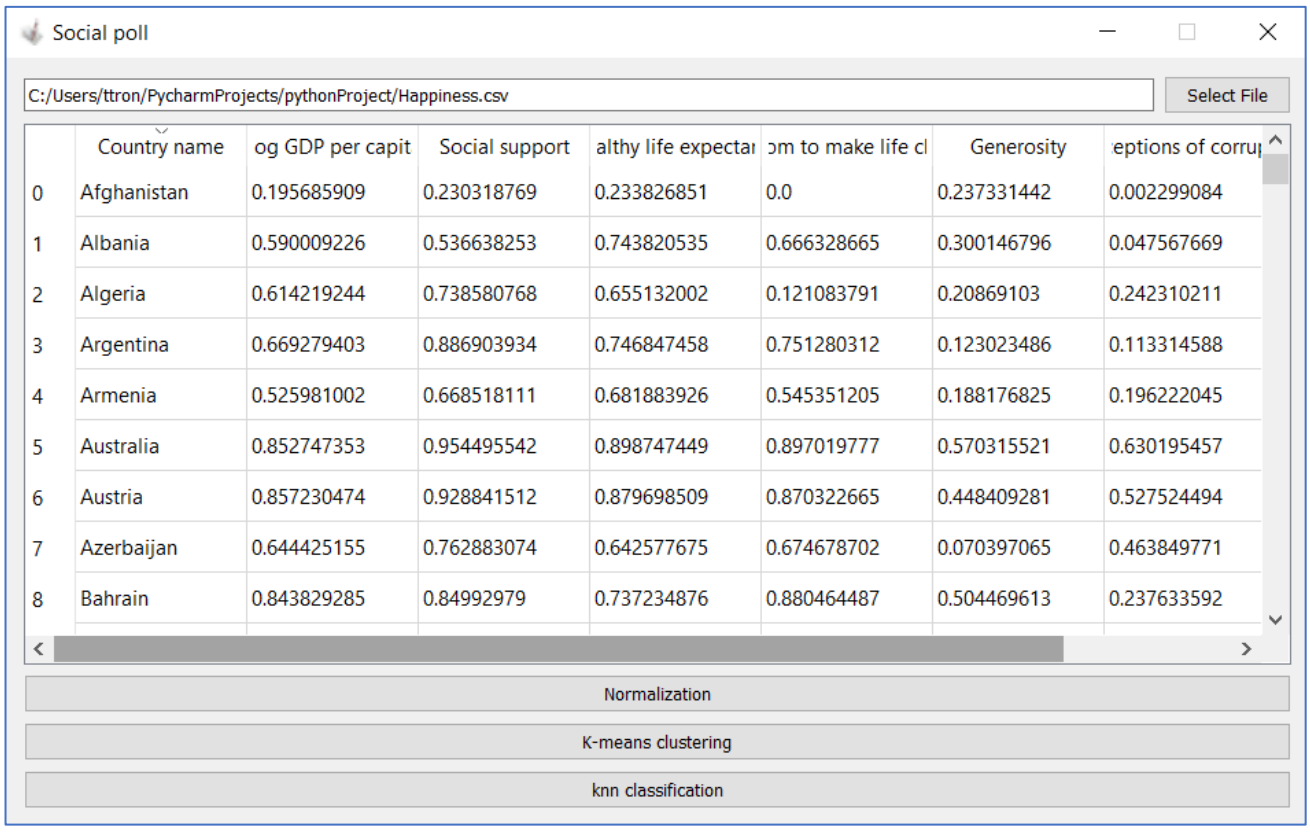
Щоб отримати відстань Евкліда formula_result було внесено під квадратний корінь:

```
if is_square:
    formula_result = math.sqrt(formula_result)
```

Програмний код застосунку наведено у додатку А.

3.4 Проведення кластеризації та класифікації соціологічних даних

Після завантаження набору даних, описаного у параграфі 3.1, його вміст буде відображено у центральній частині вікна програми (див. рис. 3.7). Ознаки набору даних представлені у різних числових діапазонах, тому для проведення кластеризації та класифікації даних необхідно здійснити нормалізацію значень ознак. Для цього необхідно натиснути кнопку Normalization. Після цього у таблиці будуть відображені нормалізовані значення атрибутів (див. рис.3.8).



	Country name	og GDP per capit	Social support	althy life expectar	om to make life cl	Generosity	eptions of corru
0	Afghanistan	0.195685909	0.230318769	0.233826851	0.0	0.237331442	0.002299084
1	Albania	0.590009226	0.536638253	0.743820535	0.666328665	0.300146796	0.047567669
2	Algeria	0.614219244	0.738580768	0.655132002	0.121083791	0.20869103	0.242310211
3	Argentina	0.669279403	0.886903934	0.746847458	0.751280312	0.123023486	0.113314588
4	Armenia	0.525981002	0.668518111	0.681883926	0.545351205	0.188176825	0.196222045
5	Australia	0.852747353	0.954495542	0.898747449	0.897019777	0.570315521	0.630195457
6	Austria	0.857230474	0.928841512	0.879698509	0.870322665	0.448409281	0.527524494
7	Azerbaijan	0.644425155	0.762883074	0.642577675	0.674678702	0.070397065	0.463849771
8	Bahrain	0.843829285	0.84992979	0.737234876	0.880464487	0.504469613	0.237633592

Рисунок 3.8 – Нормалізовані значення ознак набору даних

Після проведення нормалізації для виявлення внутрішньої структури даних у системі передбачено здійснення кластеризації за алгоритмом k-means. Для цього необхідно натиснути кнопку k-means clustering. Результат проведеної кластеризації буде відображено візуально у вигляді графіку та у таблиці, у якій навпроти кожної країни буде вказано номер кластеру, до якого вона потрапила.

Графік результатів кластеризації, в якому можна змінювати ракурс для вибору кращої точки перегляду даних методу кластеризації, ми можемо бачити у трьохвимірному просторі ознак (див. рис. 3.12 та рис. 3.13), що не відображає у повній мірі отриманий результат, оскільки аналізований набір даних містить шість ознак. На наведених нижче графіках результат кластеризації відображено у системі координат ознак: Social support (соціальна підтримка), Freedom to make life choices (свобода робити життєві вибори) та Perceptions of corruption (сприйняття корупції).

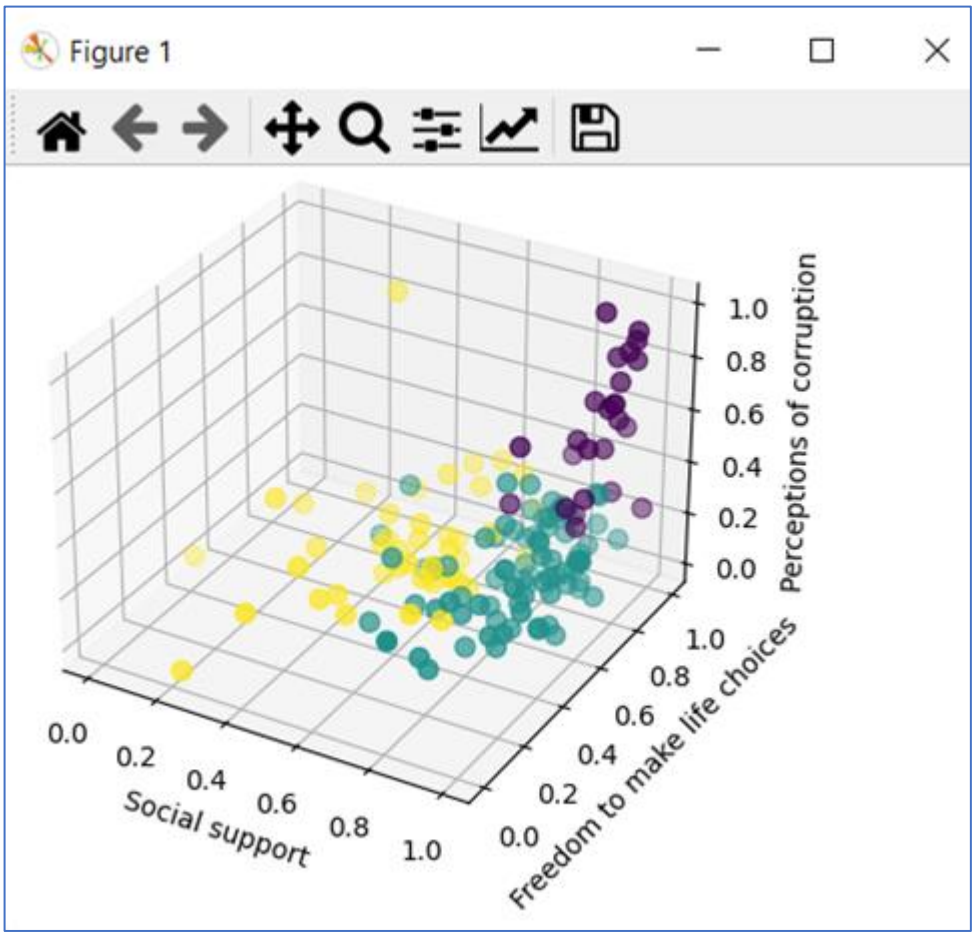


Рисунок 3.12 – Стартове відображення графіку кластерів

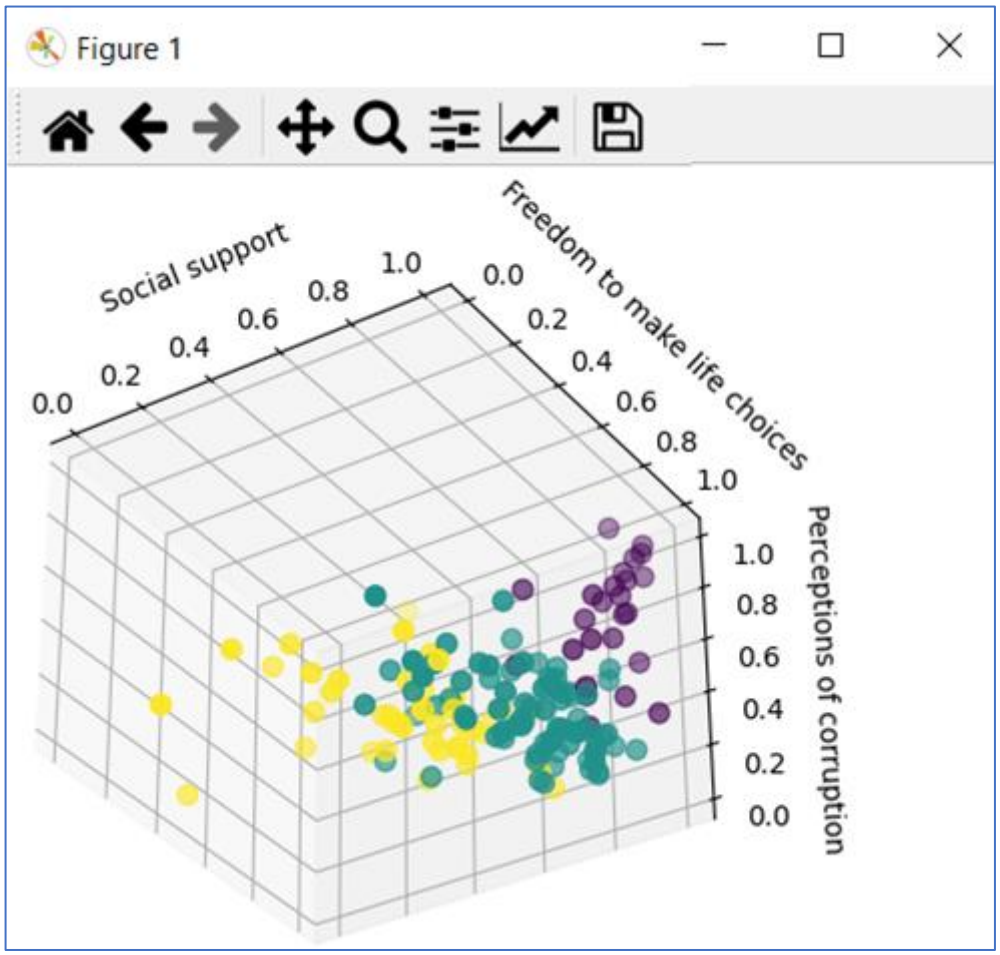
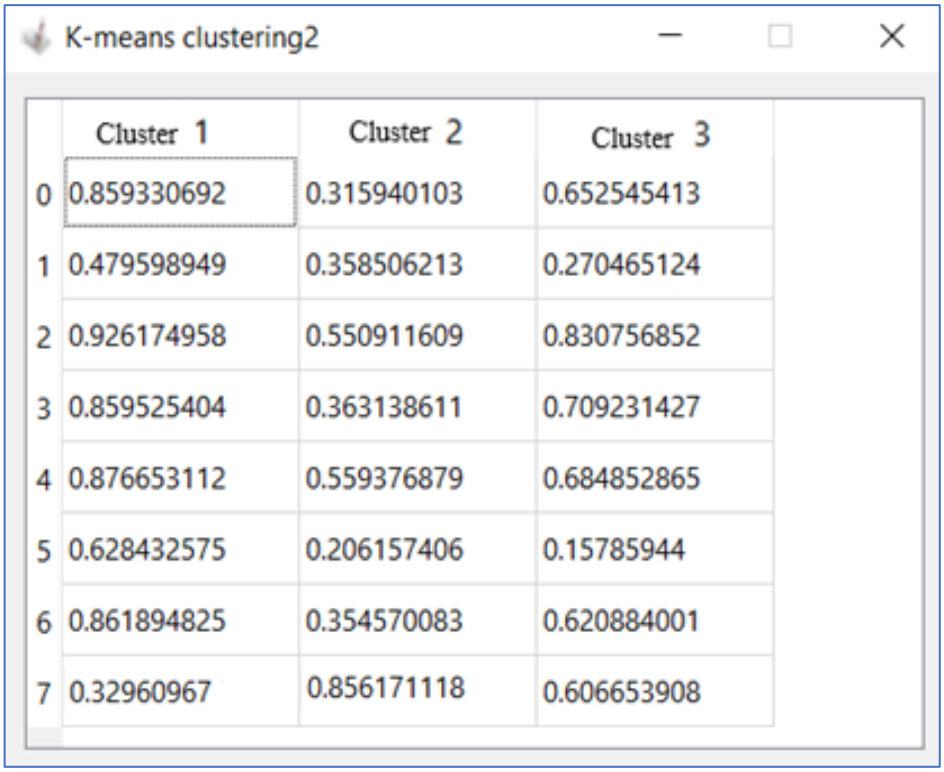


Рисунок 3.13 Графік результатів кластеризації у зміненому положенні

Для узагальненого відображення результатів кластеризації було сформовано та відображено таблицю, яка містить середні значення кожної ознаки кластерів – центроїди (див. рис 3.14). Тут 0 – Log GDP per capita (ВВП на душу населення), 1 – Healthy life expectancy (очікувана тривалість здорового життя при народженні), 2 – Social support (соціальна підтримка), 3 – Freedom to make life choices (свобода робити життєвий вибір), 4 – Generosity (щедрість), 5 – Perceptions of corruption (сприйняття корупції) 6 – Positive affect (позитивний афект),. 7 – Negative affect (Negative affect).

Проаналізувавши отримані дані ми бачимо, що для країн, які віднесено до кластеру 1, характерним є рівень ВВП на душу населення, соціальної підтримки, свободи робити життєвий вибір та щедрості. У кластері 3 ці характеристики є трохи нижчими, але задовільними, а у кластері 2 вони є самим низькими. Для кластеру 1

позитивний ефект є найвищим, а негативний ефект – найнижчим, для кластеру 2 навпаки: позитивний ефект є найнижчим, а негативний ефект – найвищим. Для кластеру 3 значення цих ознак знаходиться посередині. Оскільки рівень щастя фокусується на соціальному, міському та природному середовищі й на емоційному сприйнятті подій, можна стверджувати, що до кластеру 1 потрапили країни з високим рівнем щастя, до кластеру 3 – із середнім, а до кластеру 2 – із низьким рівнем щастя. Виходячи із цього, можна побачити, що для країн із високим рівнем щастя, характерним є найвищий рівень сприйняття корупції. А очікувана тривалість здорового життя при народженні для країн з високим рівнем щастя виявилася найвищою, найменшою – для країн з низьким рівнем щастя.



	Cluster 1	Cluster 2	Cluster 3
0	0.859330692	0.315940103	0.652545413
1	0.479598949	0.358506213	0.270465124
2	0.926174958	0.550911609	0.830756852
3	0.859525404	0.363138611	0.709231427
4	0.876653112	0.559376879	0.684852865
5	0.628432575	0.206157406	0.15785944
6	0.861894825	0.354570083	0.620884001
7	0.32960967	0.856171118	0.606653908

Рисунок 3.14 Визначені центри ваги кластерів

Для більш детального аналізу результатів кластеризації у системі передбачено виведення таблиці із вказівкою, до якого кластеру потрапила кожна країна (див. рис. 3.15). У таблиці країни відсортовано за алфавітом, що є зручним для пошуку та подальшого аналізу даних.

	Country name	Cluster
0	Afghanistan	3
1	Albania	2
2	Algeria	2
3	Argentina	2
4	Armenia	2
5	Australia	1
6	Austria	1
7	Azerbaijan	2
8	Bahrain	1
9	Bangladesh	3
10	Belarus	2
11	Belgium	1
12	Benin	3
13	Bolivia	2
14	Bosnia and ...	2
15	Botswana	2
16	Brazil	2
17	Bulgaria	2
18	Burkina Faso	3
19	Burundi	3
20	Cambodia	3
21	Cameroon	3
22	Canada	1

Рисунок 3.15 Таблиця із відображенням належності країн до кластерів

До країн із високим рівнем щастя віднесено Канаду, Нідерланди, Швецію, Швейцарію, Великобританію, США. До країн із середнім рівнем щастя потрапили Турція, Латвія. А до країн із низьким рівнем щастя – Пакистан, Азербайджан, Лаос. Проаналізувавши склад кластерів, можемо зробити висновок, що рівень щастя в країні корелює з рівнем її економічного розвитку.

Маючи результати кластеризації можна переходити до класифікації країн, які не були задіяні у аналізі. Для цього ми установимо відповідність:

1) клас 1 буде відповідати кластеру 1 (високий рівень щастя);

2) клас 2 буде відповідати кластеру 2 (низький рівень щастя);

3) клас 3 буде відповідати кластеру 3 (середній рівень щастя).

Для здійснення класифікації країн необхідно натиснути кнопку knn classification. Після цього буде запущено виконання алгоритму k-найближчих сусідів та виведено у графічному вигляді країни, які виявилися найближчими до тієї, що класифікується. На рис. 3.16 відображено результат країни, які виявилися найближчими для України. Як бачимо, усі країни відносяться до третього класу, тому, використовуючи незважене голосування, Україну можна віднести до країн із середнім рівнем щастя.

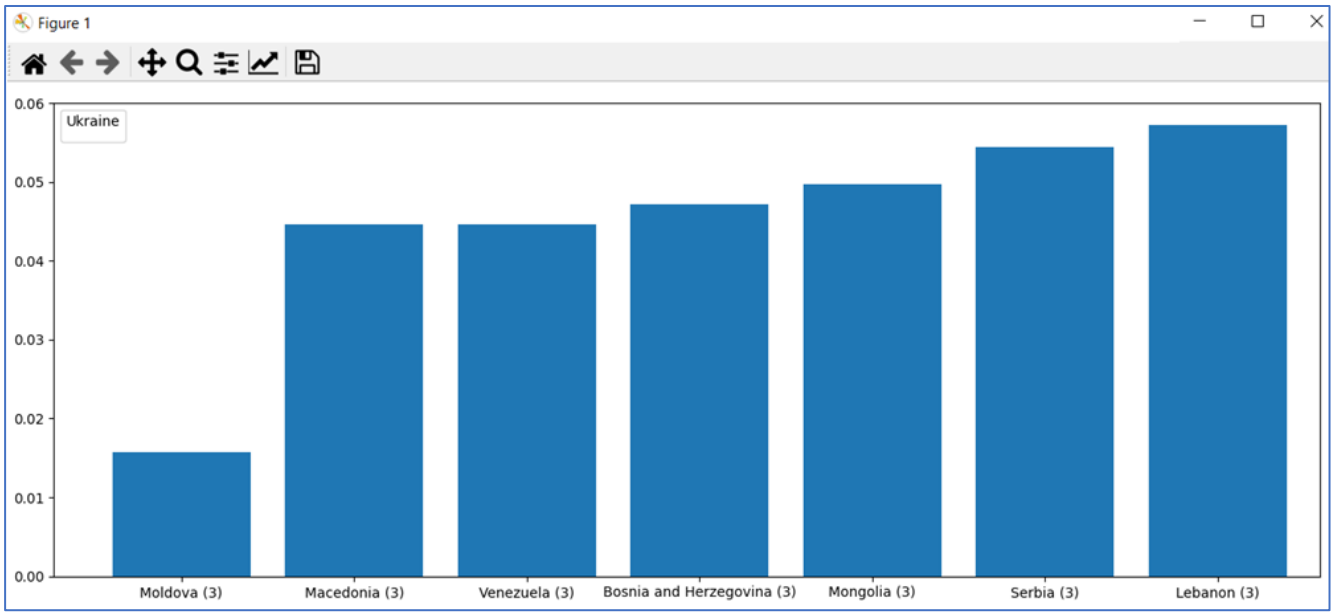


Рисунок 3.16 – Результат класифікації країни Україна за рівнем щастя

Застосувавши класифікатор для класифікації країни Естонія, ми виявили, що вона є близькою до однієї країни першого класу та шести країн третього класу. Тому її також можна віднести до країн із середнім рівнем щастя (див. рис. 3.17).

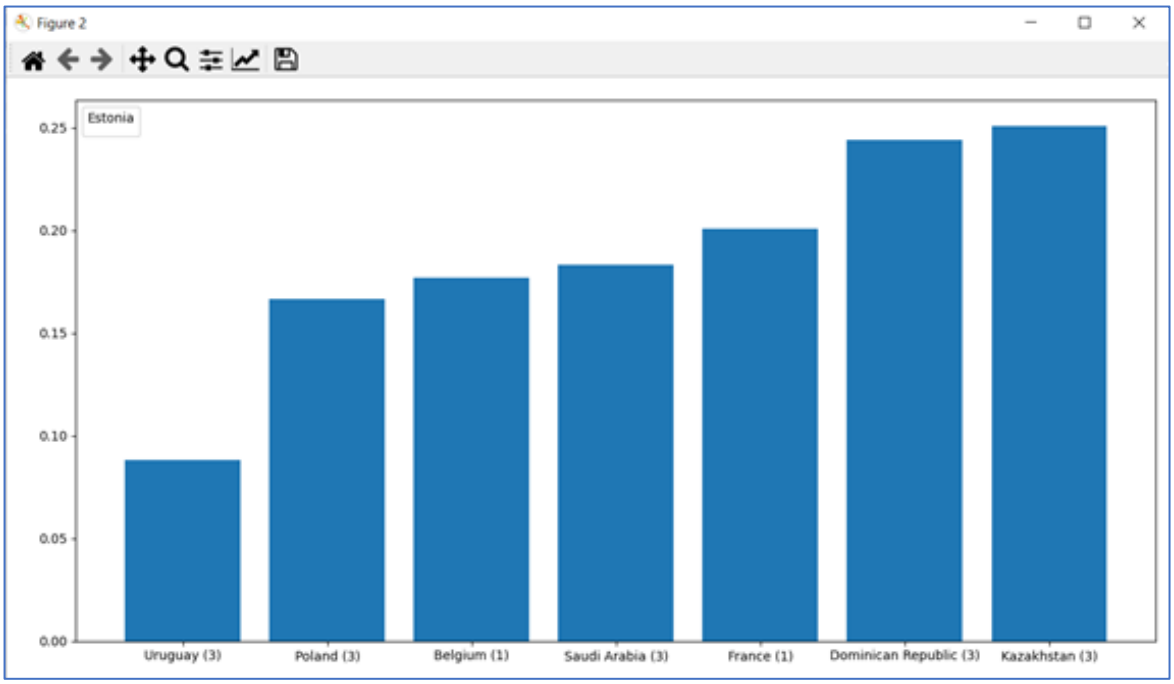


Рисунок 3.17 – Результат класифікації країни Естонія за рівнем щастя

Висновки до розділу 3

У даному розділі було описано розробку та програмну реалізацію інтелектуальної системи аналізу соціологічних даних. Розроблена система дозволяє здійснювати завантаження даних соціологічного опитування стосовно рівня життя у різних країнах світу із файлу формату csv. Для здійснення інтелектуального аналізу даних у системі передбачено проведення нормалізації даних та їх кластеризації за алгоритмом k-means. Проведена кластеризація дозволяє установити групи споріднених країн за групою ознак, які визначають рівень щастя людей, які проживають у країні: ВВП на душу населення в паритеті купівельної спроможності, соціальна підтримка, свобода вибору, щедрість, сприйняття корупції, позитивний вплив, негативний афект. Користувач має можливість переглядати характеристики країн, які потрапили до кожного кластеру, та середні значення ознак кожного кластеру. За результатами проведеної кластеризації у системі передбачено побудову класифікатора за алгоритмом k-найближчих сусідів та класифікацію нових об’єктів.

ВИСНОВКИ

Застосування інтелектуальних систем аналізу соціологічних даних є перспективним напрямком підвищення ефективності проведення соціологічних досліджень. Проведена робота дозволяє зробити наступні висновки.

Установлено, що аналіз соціологічних даних проводяться у певному середовищі суспільства шляхом збору анкетних даних, які мають відношення до різноманітні аспекти суспільного життя. В умовах становлення інформаційного суспільства змінюються методи та засоби отримання і аналізу соціологічних даних. Швидкі темпи розвитку інформаційно-комунікаційних технологій та обчислювальних можливостей призвели виникнення комп'ютерної соціології та розвитку та широкого використання сучасних методів інтелектуального аналізу соціологічних даних.

Однією із сфер соціологічних досліджень є аналіз соціологічних даних з метою визначення рівня щастя у різних країнах світу. Розглянуто основні підходи до аналізу рівня щастя у різних країнах світу та визначено особливості аналізу соціологічних даних. установлено, що є підходи, що передбачають на основі отриманих даних розрахунок індексів щастя, які дозволяють визначити рейтинг щастя у певній країні:The Legatum Prosperity Index, Happiness and Life Satisfaction, Happy Planet Index. Застосування методів Data Mining до обробки даних соціологічних опитувань дає можливість із використанням методів кластеризації розбити країни на споріднені групи за показниками, які визначають рівень щастя. А побудова класифікатора дає можливість класифікувати нові країни. Обґрунтовано вибір методів аналізу соціологічних даних для здійснення кластеризації та класифікації – алгоритмів k-means та k–найближчих сусідів та розкрито їх основні етапи.

Для розробки системи інтелектуального аналізу соціологічних даних було використано кросплатформенне інтегроване середовище розробки PyCharm, яке є потужним інструментом для розробки застосунків, підтримує обрані для розробки мови програмування, надає зручний інтерфейс, має розширену функціональність, редактор коду та інші корисні інструменти для розробників. Розробка проєкту здійснювалася на мові програмування Python із використанням бібліотек Pandas,

Matplotlib, SciPy, NumPy, PyQt5, Math, Sklearn. Бібліотека Pandas використовувалася для імпортування даних з різних форматів файлів та проведення злиття, переформатування, очищення даних. Бібліотека Matplotlib та побудована на базі неї SciPy використовувалися для візуалізації даних, їх редагування та зберігання. Для реалізації основних алгоритмів кластерного аналізу даних та класифікації застосовувалася бібліотека Sklearn.

Здійснено розробку та програмну реалізацію інтелектуальної системи аналізу соціологічних даних, яка дозволяє здійснювати завантаження даних соціологічного опитування стосовно рівня життя у різних країнах світу із файлу формату csv. Для здійснення інтелектуального аналізу даних у системі передбачено проведення нормалізації завантажених даних та їх кластеризація за алгоритмом k-means. Кластеризація дозволяє установити групи споріднених країн за групою ознак, які визначають рівень щастя людей, які проживають у країні: ВВП на душу населення в паритеті купівельної спроможності, соціальна підтримка, свобода вибору, щедрість, сприйняття корупції, позитивний вплив, негативний афект. Користувач має можливість переглядати характеристики країн, які потрапили до кожного кластеру, та середні значення ознак кожного кластеру. За результатами проведеної кластеризації у системі передбачено побудову класифікатора за алгоритмом k-найближчих сусідів та класифікацію нових об’єктів.

Поставлені завдання виконано, однак у подальшому функціонал системи може бути розширено за рахунок більш детального налаштування параметрів обраних алгоритмів кластеризації та класифікації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Паніотто В.І., Максименко В.С., Марченко Н.М. Статистичний аналіз соціологічних даних. – К.: Вид. Дім «КМ Академія», 2004. 270 с.

2. Кислова О.Н. Бути чи не бути цифровій соціології? *Вісник Харківського національного університету імені В. Н. Каразіна*, серія «Соціологічні дослідження сучасного суспільства: методологія, теорія, методи». № 1045, 2013. С. 9-15.

3. Chen Y., Yan F. Centuries of sociology in millions of books. *The Sociological Review*. Vol. 64 (4), 2016. P. 872-893.

4. Соціологія: організація емпіричного дослідження / Балановський Я.М. – Умань: Візаві, 2019. – 471 с.

5. Maione K., Nelson D., Barbosa R. Research on social data by means of cluster analysis. *Applied Computing and Informatics*. Vol. 15 (2), 2019. pp. 153-162. URL: https://www.sciencedirect.com/science/article/pii/S2210832717303526 (дата звернення 15.03.2023).

6. Губа К. Большие данные в социологии: новые данные, новая социология? *Социологическое обозрение*. Т. 17 (1)., 2018. С. 213-236.

7. Силен Д., Мейсман А., Али М. Основы Data Science и Big Data. Python и наука о данных. СПБ: Питер, 2017. 336 с. URL: https://lib.intuit.kg/wp-content/uploads/2020/04/Основы-Data-Science-и-Big-Data-ru.pdf (дата звернення 25.03.2023).

8. Басюк Т.М., Литвин В.В., Захарія Л.М., Кунанець Н.Е. Машинне навчання: навчальний посібник Львів: Видавництво «Новий Світ - 2000», 2021. 315 с.

9. Марченко О., Россада Т. Актуальні проблеми Data Mining: навчальний посібник. – К.: вид. КНУ ім. Т. Шевченка. 2017. 150 с.

10. Математичні методи інтелектуального аналізу даних: навчальний посібник / Т. Шабельник, О. Дяченко. Маріуполь, 2021. 163 с.

11. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Python. 2nd Edition. Publisher(s): O'Reilly Media, Inc. 2017. 337 p.

12. Pandas. What's new in 1.0.0: вебсайт. URL: https://pandas.pydata.org/pandas-docs/version/1.0.0/whatsnew/v1.0.0.html (дата звернення: 13.05.2023).

13. McKinney W. Pandas: a Foundational Python Library for Data Analysis and Statistics. URL: https://www.researchgate.net/publication/265194455_pandas_a_Foundational_Python_Library_for_Data_Analysis_and_Statistics (дата звернення: 1.05.2023).

14. Кластеризация К-средних с помощью Scikit-Learn в Python: вебсайт. URL: https://tonais.ru/library/klasterizatsiya-k-srednih-s-pomoschyu-scikit-learn-v-python (дата звернення 5.04.2023).

15. Welcome to Python.org: вебсайт. URL: https://www.python.org/ (дата звернення: 7.04.2023).

16. Kopf D. Meet the man behind the most important tool in data science. QUARTZ: вебсайт. 2017. URL: https://qz.com/1126615/the-story-of-the-most-important-tool-in-data-science?utm_source=flipboard&utm_content=siva_54%2Fmagazine%2FCode (дата звернення: 22.04.2023).

17. Scientific Computing Tools for Python: вебсайт. URL: https://svn.scipy.org/about.html (дата звернення: 12.05.2023).

18. Array programming with NumPy / Harris Ch.R., Millman K.J., Stefan J. van der Walt S.J. et al. *Nature*. 2020. 585. Pp. 357-362. URL: https://www.nature.com/articles/s41586-020-2649-2 (дата звернення: 30.05.2023).

19. Python v2.7.0 documentation: вебсайт. URL: https://docs.python.org/release/2.7.5/ (дата звернення: 3.05.2023).

20. Guide to the K-Nearest Neighbors Algorithm in Python and Scikit-Learn: вебсайт. URL: https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/ (дата звернення: 14.04.2023).

21. Matplotlib. Lines, bars and markers: вебсайт. URL: https://matplotlib.org/stable/gallery/index (дата звернення 24.04.2023).

22. The k-Nearest Neighbors (kNN) Algorithm in Python: вебсайт. URL: https://realpython.com/knn-python/ (дата звернення 20.04.2023).

23. fit() vs predict() vs fit_predict() in Python scikit-learn: вебсайт. URL: https://towardsdatascience.com/fit-vs-predict-vs-fit-predict-in-python-scikit-learn-f15a34a8d39f (дата звернення 15.04.2023).

24. Getting started with k-means clustering in Python: вебсайт. URL: https://www.dominodatalab.com/blog/getting-started-with-k-means-clustering-in-python (дата звернення 5.04.2023).

Додаток А

Код застосунку системи аналізу соціологічних даних

main.py

```
from PyQt5 import QtGui, QtWidgets
from PyQt5.QtWidgets import QMessageBox, QHeaderView
import pandas as pd

import pylab

import tools
from pandas_model import PandasModel
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

class GraphicWindow(QtWidgets.QWidget):
    def __init__(self, class_table):
        super().__init__()
        layout = QtWidgets.QVBoxLayout()
        self.setWindowTitle("K-means clustering")
        self.setWindowIcon(QtGui.QIcon('Sociall.jpg'))
        self.setFixedSize(350, 760)
        self.pandasTv = QtWidgets.QTableView(self)
        layout.addWidget(self.pandasTv)
        self.pandasTv.setSortingEnabled(True)
        self.setLayout(layout)
        model = PandasModel(class_table)
        self.pandasTv.setModel(model)

class GraphicWindowK(QtWidgets.QWidget):
    def __init__(self, class_table):
        super().__init__()
        layout = QtWidgets.QVBoxLayout()
        self.setWindowTitle("K-means clustering2")
        self.setWindowIcon(QtGui.QIcon('Sociall.jpg'))
        self.setFixedSize(500, 300)
        self.pandasTv = QtWidgets.QTableView(self)
        layout.addWidget(self.pandasTv)
        self.pandasTv.setSortingEnabled(True)
        self.setLayout(layout)
        model = PandasModel(class_table)
        self.pandasTv.setModel(model)

class KnnClassificationWindow(QtWidgets.QWidget):
    def __init__(self, df_table):
        self.table = df_table
        super().__init__()
        layout = QtWidgets.QVBoxLayout()
        self.setWindowTitle("knn classification")
```

```
        self.setWindowIcon(QtGui.QIcon('Sociall.jpg'))
        self.setFixedSize(350, 150)
        self.add_button("knn_butt_eu", "Euclidean distance", self.knn_euclidean, layout)
        self.add_button("knn_butt_ha", "The square of the Euclidean distance",
self.knn_square_euclidean, layout)
        self.setLayout(layout)

    def knn_euclidean(self):
        try:
            tools.classifyKNN(self.table, False)
        except Exception:
            e = sys.exc_info()[1]
            print(e.args[0])
            QMessageBox.critical(self, 'Error!', "Please, choose the file.\n"
                                 f"Index error:{e}", QMessageBox.Ok)

    def knn_square_euclidean(self):
        try:
            tools.classifyKNN(self.table, True)
        except Exception:
            e = sys.exc_info()[1]
            print(e.args[0])
            QMessageBox.critical(self, 'Error!', "Please, choose the file.\n"
                                 f"Index error:{e}", QMessageBox.Ok)

    def add_button(self, name, ui_name, on_click_func, layout):
        setattr(self, name, QtWidgets.QPushButton(ui_name, self))
        layout.addWidget(getattr(self, name))
        getattr(self, name).clicked.connect(on_click_func)

class Widget(QtWidgets.QWidget):
    def __init__(self, parent=None):
        QtWidgets.QWidget.__init__(self, parent=None)
        self.df_table = None
        vLayout = QtWidgets.QVBoxLayout(self)
        hLayout = QtWidgets.QHBoxLayout()
        self.setFixedSize(950, 560)
        self.is_normalized = False
        self.setWindowTitle("Social poll")
        self.setWindowIcon(QtGui.QIcon('Sociall.jpg'))
        self.pathLE = QtWidgets.QLineEdit(self)
        hLayout.addWidget(self.pathLE)
        self.add_button("loadBtn", "Select File", self.loadFile, hLayout)
        vLayout.addLayout(hLayout)
        self.pandasTv = QtWidgets.QTableView(self)
        vLayout.addWidget(self.pandasTv)
        self.pandasTv.setSortingEnabled(True)
        self.add_button("norm_butt", "Normalization", self.norm, vLayout)
        self.add_button("k_means_butt", "K-means clustering", self.k_means, vLayout)
        self.add_button("knn_classification", "knn classification",
```

```
self.open_knn_classification_window, vLayout)

    def add_button(self, name, ui_name, on_click_func, layout):
        setattr(self, name, QtWidgets.QPushButton(ui_name, self))
        layout.addWidget(getattr(self, name))
        getattr(self, name).clicked.connect(on_click_func)

    def loadFile(self):
        try:
            fileName, _ = QtWidgets.QFileDialog.getOpenFileName(self, "Open File", "", "CSV Files
(*.csv)");
            self.pathLE.setText(fileName)
            self.df_table = pd.read_csv(fileName)
            model = PandasModel(self.df_table)
            self.pandasTv.setModel(model)
        except Exception:
            e = sys.exc_info()[1]
            print(e.args[0])
            QMessageBox.critical(self, 'Error!', "Please, choose the file.\n"
                                 f"Index error:{e}", QMessageBox.Ok)

    def norm(self):
        try:
            tools.norm(self.df_table, "Log GDP per capita")
            tools.norm(self.df_table, "Generosity")
            tools.norm(self.df_table, "Social support")
            tools.norm(self.df_table, "Healthy life expectancy")
            tools.norm(self.df_table, "Freedom to make life choices")
            tools.norm(self.df_table, "Perceptions of corruption")

            self.is_normalized = True

        except:
            e = sys.exc_info()[1]
            print(e.args[0])
            QMessageBox.critical(self, 'Error!', "Please, before using functions - load .csv file\n"
                                 f"Detailed error description:{e}", QMessageBox.Ok)

    def open_knn_classification_window(self):
        try:
            some_res = self.df_table["Country name"]
            if not self.is_normalized:
                QMessageBox.critical(self, 'Error!', "Please, normalize data", QMessageBox.Ok)
                return
            self.w = KnnClassificationWindow(self.df_table)
            self.w.show()
        except Exception:
            e = sys.exc_info()[1]
            print(e.args[0])
            QMessageBox.critical(self, 'Error!', "Please, before using functions - load .csv file \n"
```

```
                f"Detailed error description:{e}", QMessageBox.Ok)


def k_means(self):
    try:
        self.df_table["Country name"]

        if not self.is_normalized:
            QMessageBox.critical(self, 'Error!', "Please, normalize data", QMessageBox.Ok)
            return

        x_l, y_l, z_l, f_l, b_l, m_l, x, y, z, f, b, m, kmeans = tools.k_means(self.df_table)

        fig = pylab.figure()
        ax = fig.add_subplot(111, projection='3d')

        # specify the s parameter to control the size of the markers
        sc = ax.scatter(z, b, m, c=kmeans.labels_, s=50)

        print(kmeans.cluster_centers_)

        ax.set_xlabel(z_l)
        ax.set_ylabel(b_l)
        ax.set_zlabel(m_l)
        # ax.set_flabel(f_l)
        # ax.set_blabel(b_l)
        # ax.set_mlabel(m_l)

        # for i in range(len(z)):
        #     ax.text(y[i], z[i], x[i], '{}'.format(self.df_table["Country name"][i]), fontsize=7)

        pylab.show()

        class_data = kmeans.labels_
        class_data = list(map(lambda x: x + 1, class_data))
        class_table = self.df_table[["Country name", "Generosity"]]
        class_table["Generosity"] = class_data
        class_table.rename(columns={'Generosity': 'Class'}, inplace=True)

        clust_cent = []

        for cluster_centers in kmeans.cluster_centers_:
            temp = []
            for cluster_center in cluster_centers:
                temp.append(round(cluster_center, 9))
            clust_cent.append(temp)

        centers = pd.DataFrame({'Center 1': clust_cent[2],
                                'Center 2': clust_cent[1],
                                'Center 3': clust_cent[0]})
```

```
            self.w = GraphicWindow(class_table)
            self.w.show()
            self.g = GraphicWindowK(centers)
            self.g.show()

        except Exception:
            e = sys.exc_info()[1]
            print(e.args[0])
            QMessageBox.critical(self, 'Error!', "Please, before using functions - load .csv file \n"
                                 f"Detailed error description:{e}", QMessageBox.Ok)

if __name__ == "__main__":
    import sys

    app = QtWidgets.QApplication(sys.argv)
    w = Widget()
    w.show()
    sys.exit(app.exec_())
```

pandas_model.py

```
from PyQt5 import QtCore

import pandas as pd

class PandasModel(QtCore.QAbstractTableModel):
    def __init__(self, df=pd.DataFrame(), parent=None):
        QtCore.QAbstractTableModel.__init__(self, parent=parent)
        self._df = df

    def headerData(self, section, orientation, role=QtCore.Qt.DisplayRole):
        if role != QtCore.Qt.DisplayRole:
            return QtCore.QVariant()

        if orientation == QtCore.Qt.Horizontal:
            try:
                return self._df.columns.tolist()[section]
            except (IndexError,):
                return QtCore.QVariant()
        elif orientation == QtCore.Qt.Vertical:
            try:
                return self._df.index.tolist()[section]
            except (IndexError,):
                return QtCore.QVariant()

    def data(self, index, role=QtCore.Qt.DisplayRole):
        if role != QtCore.Qt.DisplayRole:
            return QtCore.QVariant()

        if not index.isValid():
```

```
            return QtCore.QVariant()

        return QtCore.QVariant(str(self._df.iloc[index.row(), index.column()]))

    def setData(self, index, value, role):
        row = self._df.index[index.row()]
        col = self._df.columns[index.column()]
        if hasattr(value, 'toPyObject'):
            value = value.toPyObject()
        else:
            dtype = self._df[col].dtype
            if dtype != object:
                value = None if value == '' else dtype.type(value)
        self._df.set_value(row, col, value)
        return True

    def rowCount(self, parent=QtCore.QModelIndex()):
        return len(self._df.index)

    def columnCount(self, parent=QtCore.QModelIndex()):
        return len(self._df.columns)

    def sort(self, column, order):
        colname = self._df.columns.tolist()[column]
        self.layoutAboutToBeChanged.emit()
        self._df.sort_values(colname, ascending=order == QtCore.Qt.AscendingOrder, inplace=True)
        self._df.reset_index(inplace=True, drop=True)
        self.layoutChanged.emit()
```

tools.py

```
import math
import sys
import numpy as np
import matplotlib.pyplot as plt
import pylab
from PyQt5 import QtWidgets
from PyQt5.QtWidgets import QMessageBox
from sklearn.cluster import KMeans

def generateData(numberOfClassEl, numberOfClasses, table):
    import random
    data = []
    for classNum in range(table.shape[0]):
        # Вибрати випадковий центр двовимірного гаусівського
        centerX, centerY = table["Ladder score"][classNum], table["upperwhisker"][classNum]
        # Виберіть випадкові вузли numberOfClassEl із RMS=0,5
        data.append([[random.gauss(centerX, 0.5), random.gauss(centerY, 0.5)], 1, table["Country
name"][classNum]])

    for classNum in range(table.shape[0]):
```

```
        # Вибрати випадковий центр двовимірного гаусівського
        centerX, centerY = table["lowerwhisker"][classNum], table["Logged GDP per
capita"][classNum]
        # Виберіть випадкові вузли numberOfClassEl із RMS=0,5
        data.append([[random.gauss(centerX, 0.5), random.gauss(centerY, 0.5)], 2, table["Country
name"][classNum]])

    for classNum in range(table.shape[0]):
        # Вибрати випадковий центр двовимірного гаусівського
        centerX, centerY = table["Ladder score in Dystopia"][classNum], table["Explained by: Log
GDP per capita"][
            classNum]
        # Виберіть випадкові вузли numberOfClassEl із RMS=0,5
        data.append([[random.gauss(centerX, 0.5), random.gauss(centerY, 0.5)], 3, table["Country
name"][classNum]])

    return data

# def generateData(numberOfClassEl, numberOfClasses, table):
#     print(table.shape[0])
#
#     data = []
#     for rowNum in range(table.shape[0]):
#         data.append([ [table["Logged GDP per capita"][rowNum], table["Healthy life
expectancy"][rowNum]], 1 ])
#         print(data)
#     return data

# Основна процедура класифікації

def euclidean_distance(a, b):
    return (a - b) ** 2

def k_means(table):
    x_l = "Log GDP per capita"
    y_l = "Generosity"
    z_l = "Social support"
    f_l = "Healthy life expectancy"
    b_l = "Freedom to make life choices"
    m_l = "Perceptions of corruption"

    x = table[x_l]
    y = table[y_l]
    z = table[z_l]
    f = table[f_l]
    b = table[b_l]
    m = table[m_l]

    data = list(zip(x, y, z, f, b, m))
```

```
    kmeans = KMeans(n_clusters=3, random_state=0, n_init="auto")
    kmeans.fit(data)

    return x_l, y_l, z_l, f_l, b_l, m_l, x, y, z, f, b, m, kmeans

def classifyKNN(table, is_square):
    x_l, y_l, z_l, f_l, b_l, m_l, x, y, z, f, b, m, kmeans = k_means(table)

    class_data = kmeans.labels_
    class_data = list(map(lambda x: x + 1, class_data))
    class_table = table[["Country name", "Generosity"]]
    class_table["Generosity"] = class_data
    class_table.rename(columns={'Generosity': 'Class'}, inplace=True)

    country_row_num = 143

    data_col_names = ["Log GDP per capita", "Social support", "Healthy life expectancy", "Freedom
to make life choices",
                      "Generosity", "Perceptions of corruption"]

    final_result_with_names = []

    country_row_num = country_row_num - 2

    for row_country_id in range(len(table["Country name"])):
        temp = []

        for data_col_name in data_col_names:
            first = table[data_col_name][row_country_id]
            second = table[data_col_name][country_row_num]
            euclidean_result = euclidean_distance(first, second)
            temp.append(euclidean_result)

        formula_result = sum(temp)

        if is_square:
            formula_result = math.sqrt(formula_result)

        country_name = "{name} ({class_name})".format(name=table["Country
name"][row_country_id] , class_name=class_table["Class"][row_country_id])
        final_result_with_names.append((country_name, formula_result))

    del final_result_with_names[country_row_num]
    final_result_with_names = sorted(final_result_with_names, key=lambda x: x[1])

    res_class = []
    res_names = []

    for id in range(7):
        res_class.append(final_result_with_names[id][1])
```

```
        res_names.append(final_result_with_names[id][0])


    fig, ax = plt.subplots()
    ax.bar(res_names, res_class)
    ax.legend(title=table["Country name"][country_row_num])
    plt.show()

def norm(table, colum):
    table[colum] = table[colum].apply(
        lambda v: round((v - table[colum].min()) / (table[colum].max() - table[colum].min()), 9))

def showDataOnMesh(nClasses, nItemsInClass, k, classify_func, table):
    # Створимо сітку вузлів, яка охоплює всі точки
    from matplotlib.colors import ListedColormap
    def generateTestMesh(trainData):
        x_min = min([trainData[i][0][0] for i in range(len(trainData))]) - 1.0
        x_max = max([trainData[i][0][0] for i in range(len(trainData))]) + 1.0
        y_min = min([trainData[i][0][1] for i in range(len(trainData))]) - 1.0
        y_max = max([trainData[i][0][1] for i in range(len(trainData))]) + 1.0
        h = 0.05
        testX, testY = np.meshgrid(np.arange(x_min, x_max, h),
                                   np.arange(y_min, y_max, h))
        return [testX, testY]

    trainData = generateData(nItemsInClass, nClasses, table)
    testMesh = generateTestMesh(trainData)
    testMeshLabels = classify_func(trainData, zip(testMesh[0].ravel(), testMesh[1].ravel()), k,
table.shape[0])
    classColormap = ListedColormap(['#FF0000', '#00FF00', '#FFFFFF'])
    testColormap = ListedColormap(['#FFAAAA', '#AAFFAA', '#AAAAAA'])

    plt.ion()
    plt.clf()
    plt.pcolormesh(testMesh[0],
                   testMesh[1],
                   np.asarray(testMeshLabels).reshape(testMesh[0].shape),
                   cmap=testColormap)
    plt.scatter([trainData[i][0][0] for i in range(len(trainData))],
                [trainData[i][0][1] for i in range(len(trainData))],
                c=[trainData[i][1] for i in range(len(trainData))],
                cmap=classColormap)

    for coords in range(len(trainData)):
        plt.text(trainData[coords][0][0], trainData[coords][0][1], trainData[coords][2],
                 fontsize=5, color='g')

    plt.show(block=False)
```