НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

КУРСОВА РОБОТА

з дисципліни «Аналіз даних в інформаційних системах»

на тему: «Аналіз впливу деяких факторів на рівень щастя населення країн. Регресійний аналіз.»

Студентів 2 курсу ІП-01 групи
Спеціальності: 121
«Інженерія програмного забезпечення»
Заранік Богдан Юрійович
Пашковський Євгеній Сергійович

«ПРИЙНЯВ» з оцінкою

_____

доц. Ліхоузова Т.А. / доц. Олійник Ю.О.

_____

___

Підпис Дата

Київ - 2022 рік

Національний технічний університет України “КПІ ім. Ігоря Сікорського”

Кафедра інформатики та програмної інженерії

Дисципліна Аналіз даних в інформаційно-управляючих системах

Спеціальність 121 "Інженерія програмного забезпечення"

Курс 2 Група ІП-01 Семестр 4

ЗАВДАННЯ

на курсову роботу студентів

Зараніка Богдана Юрійовича та Пашковського Євгенія Сергійовича

1.Тема: Аналіз впливу деяких факторів на рівень щастя населення країн.

2.Строк здачі студентом закінченої роботи 19.06.2022

3. Вхідні дані до роботи методичні вказівки до курсової робота, обрані дані з сайту

https://www.kaggle.com/datasets/virajkulkarni952/country-development-indicators

https://www.kaggle.com/datasets/mayzannilarthein44/world-happiness-report-2015-to-2022

https://www.kaggle.com/datasets/jamesvandenberg/renewable-power-generation?select=Country_Consumption_TWH.csv

https://www.kaggle.com/datasets/saleh846/causes-of-deaths-worldwide?select=age-between-5-and-14.csv

4.Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

1.Постановка задачі

2.Аналіз предметної області

3.Розробка сховища даних

4.Інтелектуальний аналіз даних

5.Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

6.Дата видачі завдання 16.04.2022

КАЛЕНДАРНИЙ ПЛАН

№п/п	Назва етапів курсової роботи	Термін виконання етапів роботи	Підписи керівника, студента
1.	Отримання теми курсової роботи	16.04.2022	
2.	Визначення зовнішніх джерел даних	20.04.2022	
3.	Пошук та вивчення літератури з питань курсової роботи	25.04.2022	
4.	Розробка моделі сховища даних	01.05.2022	
5.	Розробка ETL процесів	15.05.2022	
6.	Обґрунтування методів інтелектуального аналізу даних	20.05.2022	
7.	Застосування та порівняння ефективності методів інтелектуального аналізу даних	25.05.2022	
8.	Підготовка пояснювальної записки	15.06.2022	
9.	Здача курсової роботи на перевірку	19.06.2022	
10.	Захист курсової роботи	21.06.2022	

Студент _____ Заранік Богдан Юрійович

(підпис) (прізвище, ім’я, по батькові)

Студент _____ Пашковський Євгеній Сергійович

(підпис) (прізвище, ім’я, по батькові)

Керівник _____ доц. Ліхоузова Т.А

(підпис) (прізвище, ім’я, по батькові)

Керівник _____ доц. Олійник Ю.О.

(підпис) (прізвище, ім’я, по батькові)

"26" червня 2022 р.

АНОТАЦІЯ

Пояснювальна записка до курсової роботи: 30 сторінок, 29 рисунків, 11 посилань.

Об’єкт дослідження: інтелектуальний аналіз даних.

Предмет дослідження: створення програмного забезпечення, що дозволить аналізувати залежність рівня щастя від деяких параметрів розвитку країн, його прогнозування та тернарна класифікація країн за рівнем розвитку у залежності від вищезгаданих параметрів.

Мета роботи: проектування та реалізація сховища даних, ETL процесів та імплементація програмного забезпечення мовою Python для отримання даних зі сховища та їх подальшого аналізу, прогнозування та класифікації.

Курсова робота включає в себе: опис проектування, створення та заповнення сховища даних згідно з темою завдання за допомогою використання бібліотек мови Python та скриптів SQL, опис створення програмного забезпечення для інтелектуального аналізу даних, їх графічного представлення у вигляді графіків та гістограм та прогнозування за допомогою математичних моделей.

МОДЕЛЬ ПРОГНОЗУВАННЯ, КЛАСИФІКАЦІЯ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, ETL ПРОЦЕСИ.

ЗМІСТ

ВСТУП

Питання рівня щастя населення має першочергове значення у сучасному світі, що розвивається з величезною швидкістю. Від нього залежить рівень життя населення, темпи економічного розвитку країни, привабливість країни для спеціалістів сучасних професій із-за кордону, інвестиційна привабливість та інші.

Цей параметр є достатньо відносним, проте навіть його можна оцінити за певною шкалою. На рівень щастя населення впливають багато чинників, перш за все - економічні та соціальні.

Міжнародний індекс щастя (англ. Happy Planet Index) являє собою індекс, що відображає добробут людей та стан навколишнього середовища в різних країнах світу. Головне завдання індексу відобразити «реальний» добробут націй. Для порівняння рівня життя в різних країнах використовується значення ВВП на душу населення або ІЛР, але ці індекси не завжди можуть відобразити реальний стан речей. Зокрема порівняння значення ВВП на душу населення вважається недоречним, оскільки кінцева мета більшості людей не бути багатими, а бути щасливими та здоровими.

У нашій роботі ми задалися питанням, які саме чинники впливають на рівень щастя людей суттєво, а якими можна знехтувати.

Також, як відомо, країни поділяють на “розвинені”, “ті, що розвиваються” та “слабо розвинені”. За даними вибірок ми також маємо на меті класифікувати країни за цими трьома типами, оскільки дуже вірогідно, що параметри, що сильно впливають на рівень щастя населення країн, також будуть суттєво вливати на те, до якого класу розвиненості належить та чи інша країна. Отже, цю гіпотезу нам і потрібно перевірити шляхом створення математичної моделі класифікації даних.

В ролі системи керування сховищем даних для даної роботи буде виступати PostgreSQL, а мова програмування для реалізації застосунку – Python3.

1. ПОСТАНОВКА ЗАДАЧІ

Мета нашого дослідження - розробка ПО, що дозволяє виокремити із переліку параметрів, що впливають на рівень щастя населення, ті, що впливають суттєво, та розробити математичну модель прогнозування рівня щастя населення за даними конкретними параметрами. Також метою нашого дослідження є розробка моделі класифікації країн за рівнем розвитку.

Під час виконання курсової роботи необхідно виконати наступні завдання: Створення сховища даних типу «сніжинка».

Сховище даних повинне містити щонайменше 3 таблиць вимірів та 2 таблиць фактів.

Створення ETL процесів для завантаження даних до сховища, їх отримання зі самого сховища за допомогою запитів, а також оновлення та додавання даних до таблиць вимірів.

Реалізувати спроектоване сховище даних з використанням PostgreSQL версії 11.

Створення застосунку, що отримує вибірку даних зі створеного сховища, графічно відображає отримані дані, проводить їх інтелектуальний аналіз для отримання передбачення за допомогою різних моделей прогнозування.

Аналіз отриманих результатів, порівняння різних методів прогнозування на даній вибірці, отримання найоптимальнішого методу. Використати мову програмування Python 3 для реалізації застосунку.

Дані математичні моделі можуть суттєво допомогти економістам у перевірці своїх припущень щодо встановленого рівня щастя населення для певної країни та віднесення її до певного класу за рівнем розвиненості.

Результатом роботи алгоритму передбачення рівня щастя населення має бути певне дійсне число, що диференціює країну серед інших за вищезгаданим параметром.

Результатом роботи алгоритму класифікації країни за рівнем розвитку має бути один із трьох класів: “розвинена”, “та, що розвивається” та “слабо розвинена” - передбачуваний рівень розвитку даної країни.

2. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

На нашу думку, впливати на рівень щастя населення та рівень розвитку країни можуть такі чинники:

- ВВП на душу населення
- Рівень свободи
- Рівень довіри населення владі
- Щедрість населення
- Розповсюдженість наркотичних засобів
- Захворюваність на на деякі види захворювань
- Соціальна підтримка зі сторони держави
- Споживання електроенергії на душу населення
- Очікувана середня тривалість життя
- Площа країни проживання
- Загальна смертність на 10000 населення

Даний список параметрів, що впливають на рівень щастя населення, є неповним, оскільки на нього впливає безліч чинників, але досліджувати шукану залежніть ми будемо саме за ним, адже на нашу думку у списку присутня більшість параметрів, що так чи інакше складають оцінку рівня щастя.

У програмній системі буде реалізовано наступну функціональність, що включає в себе:

- створення ETL процесів для завантаження даних;
- створення датасету зі сховища у вигляді csv-файлу;
- графічне відображення отриманих результатів та їх аналіз.
- інтелектуальний аналіз даних;
- використання регресійних моделей прогнозування;
- використання математичних моделей класифікації даних(навчанні із вчителем);

Нами було знайдено 4 датасети.

1. World happiness report 2015-2022.

   Описує велику кількість показників, що впливають на рівень щастя населення різних країн по роках 2015-2022 та сам рівень щастя.

2. Country Electricity Consumption

   Описує параметр споживання електроенергії певної країни певного року .

3. Deaths Reasons

   Описує причини смертності населення певних країн по роках.

4. Population By Country

   Містить демографічний аналіз певних країн. Параметри такі як густина населення, кількість населення та інші.

3. РОЗРОБКА СХОВИЩА ДАНИХ

3.1. Розробка ETL процесів

Після завантаження датасетів їх потрібно ретельно підготувати до завантаження до сховища даних для подальшого зручного інтелектуального аналізу. Для цього нами було створено низку скриптів на мові Python, що перетворюють “сирі” дані у єдиний зручний формат для обробки.

На першому лістингу зображено підключення необхідних бібліотек для перетворення даних та встановлення деяких конфігурації консольного виводу.

```
import pandas as pd
pd.options.display.max_rows = 10000
pd.options.display.max_columns = 10000
pd.set_option('display.expand_frame_repr', False)
```

Рисунок 3.1 - Завантаження та імпорт бібліотек

Далі було створено функцію, яка приводить колонку рядкового типу до типу float, при цьому перетворюючи розділювач для дійсних чисел з коми на крапку.

```
def convert_column_to_float(dataset, column_label):
    dataset[column_label] = dataset[column_label]\
        .astype(str).str.replace(',', '.').astype(float)
```

Рисунок 3.2 - Функція-конвертер

3.1.1. ETL для датасету World happiness report 2015-2022

Далі потрібно перетворити деякі рядки датасету за допомогою вищезгаданої функції, попередньо завантаживши потрібний датасет до оперативної пам’яті комп’ютера. Також варто помітити, що деякі країни мають зірочку у кінці назви, тому дану неточність потрібно виправити, що і було зроблено.

```
df = pd.read_csv('../stage_zone/world-happiness-report-2015-2022-cleaned.csv',
                 sep=',', decimal='.', encoding='cp1252')
convert_column_to_float(df, 'Happiness Score')
convert_column_to_float(df, 'Economy (GDP per Capita)')
convert_column_to_float(df, 'Family (Social Support)')
convert_column_to_float(df, 'Health (Life Expectancy)')
convert_column_to_float(df, 'Freedom')
convert_column_to_float(df, 'Trust (Government Corruption)')
convert_column_to_float(df, 'Generosity')
df["Country"] = df["Country"].str.replace('*', '')
```

Рисунок 3.3 - Перетворення датасету World happiness report 2015-2022

І насамкінець видалимо непотрібну колонку та збережемо виправлений датасет.

```
df = df.drop(columns='Unnamed: 0')

df.to_csv('../main_warehouse/happiness.csv')
```

Рисунок 3.4 - Збереження датасету World happiness report 2015-2022

3.1.2. ETL для датасету Deaths Reasons.

Датасет Deaths Reasons складений із декількох частин, які потрібно об’єднати. Для цього візьмемо датасети для вікових категорій 15-49, 50-69 та 70+ років. Згрупуємо їх за складеним ключем [Country, Year] та просумуємо задля отримання датасету для вікової категорії 15+ років.

```
df1 = pd.read_csv('../stage_zone/age-between-15-and-49.csv',
                  sep=',', decimal='.', encoding='cp1252')
df2 = pd.read_csv('../stage_zone/age-between-50-and-69.csv',
                  sep=',', decimal='.', encoding='cp1252')
df3 = pd.read_csv('../stage_zone/above-age-70.csv', sep=',',
                  decimal='.', encoding='cp1252')

df = pd.concat([df1, df2, df3])
df = df.groupby(['Country', 'Year']).sum()

df.to_csv('../main_warehouse/death_reasons.csv')
```

Рисунок 3.5 - Перетворення Deaths Reasons

3.1.3. ETL для Country Consumption

Оскільки у даному датасеті колонками є рік та країни(тобто “таблиця широкого формату”), то варто для більш зручного аналізу даних перетворити її у довгий формат за допомогою ф-ції melt. Вона залишить колонку “Year”, а всі колонки із назвами країн потраплять у нову колонку із назвою “Country” як значення. Значення, що були на перехресті конкретного року та у колонці певної країни, потраплять у нову колонку із назвою “Consumption”. Наприклад, наглядно зміна колонок була такою: були колонки Year, China, England…, USA…; стали колонки Year, Country, Consumption.

```
df = pd.read_csv('../stage_zone/Country_Consumption_TWH.csv',
                 sep=',', decimal='.', encoding='cp1252')
df = df.melt(id_vars=["Year"],
             var_name="Country",
             value_name="Consumption")


df.dropna(inplace=True)
df['Year'] = df['Year'].astype(int)
df.to_csv('../main_warehouse/country_consumption.csv')
```

Рисунок 3.6 - Перетворення Country Consumption

Year	China	United States	Brazil	Belgium	Czechia	France	Ge
1990.0	874.0	1910.0	141.0	48.0	50.0	225.0	35
1991.0	848.0	1925.0	143.0	50.0	45.0	237.0	34
1992.0	877.0	1964.0	145.0	51.0	44.0	234.0	33
1993.0	929.0	1998.0	148.0	49.0	43.0	238.0	33
1994.0	973.0	2036.0	156.0	52.0	41.0	231.0	33
1995.0	1045.0	2063.0	162.0	53.0	42.0	240.0	33

Рисунок 3.7 - До перетворення

Id	Year	Country	Consumption
0	1990	China	874.0
1	1991	China	848.0
2	1992	China	877.0
3	1993	China	929.0
4	1994	China	973.0
5	1995	China	1045.0
6	1996	China	1074.0
7	1997	China	1073.0

Рисунок 3.8 - Після перетворення

3.1.4. Скрипт mainETL.py

Для зручності запуску ETL процесів було створено скрипт, що автоматично запускає усі 3 ETL скрипта послідовно.

```
#starts all ETL-files
exec(open('happiness.py').read())
exec(open('death_reasons.py').read())
exec(open('country_consumption.py').read())
```

Рисунок 3.9 - Головний ETL скрипт

3.2. Створення сховища даних

Сховище даних складається з двох фактових таблиць та трьох вимірів. Таким чином, сховище даних має тип “сніжинка”. Схема сховища даних наведена на рисунку 3.13. Опис таблиць бази даних наведені у наступній Таблиці 3.1.

Назва таблиці	Семантичне значення таблиці
happiness_report	фактова таблиця, звіт про фактори, що потенційно можуть впливати на рівень щастя та колонка із рівнем щастя
death_report	фактова таблиця, звіт про смертність населення та її причини
date	часовий вимір, представлення року
country	вимір, представлення країни
death_reason	вимір, представлення причин смертності

Таблиця 3.1 - Опис таблиць бази даних

```
CREATE SCHEMA IF NOT EXISTS stageCoursework;
CREATE SCHEMA IF NOT EXISTS ADISCoursework;
CREATE TABLE IF NOT EXISTS ADISCoursework.date(
    id bigserial PRIMARY KEY UNIQUE NOT NULL,
    year int
);

CREATE TABLE IF NOT EXISTS ADISCoursework.country(
    id bigserial PRIMARY KEY UNIQUE NOT NULL,
    name varchar(100) UNIQUE NOT NULL,
    area int,
    average_population bigint,
    net_population_change bigint
);
```

Рисунок 3.10 - Створення таблиць date і country

```
CREATE TABLE IF NOT EXISTS ADISCoursework.happiness_report(
    id bigserial PRIMARY KEY UNIQUE NOT NULL,
    date_id bigint NOT NULL,
    country_id bigint NOT NULL,
    happiness_score float4 NOT NULL,
    gdp float4,
    social_support float4,
    life_expectancy float4,
    freedom float4,
    trust float4,
    generosity float4,
    energy_consumption float4,
    total_deaths int,
    category text,
    FOREIGN KEY (date_id) REFERENCES ADISCoursework.date(id),
    FOREIGN KEY (country_id) REFERENCES ADISCoursework.country(id)
);
```

Рисунок 3.11 - Створення таблиці happiness_report

```
CREATE TABLE IF NOT EXISTS ADISCoursework.death_reason(
    id bigserial PRIMARY KEY UNIQUE NOT NULL,
    name text NOT NULL
);

CREATE TABLE IF NOT EXISTS ADISCoursework.death_report(
    id bigserial PRIMARY KEY UNIQUE NOT NULL,
    country_id int NOT NULL,
    date_id int NOT NULL,
    reason_id int NOT NULL,
    count int,
    FOREIGN KEY (country_id) REFERENCES ADISCoursework.country(id),
    FOREIGN KEY (date_id) REFERENCES ADISCoursework.date(id),
    FOREIGN KEY (reason_id) REFERENCES ADISCoursework.death_reason(id)
);
```

Рисунок 3.12 - Створення таблиць death_report та death_reason

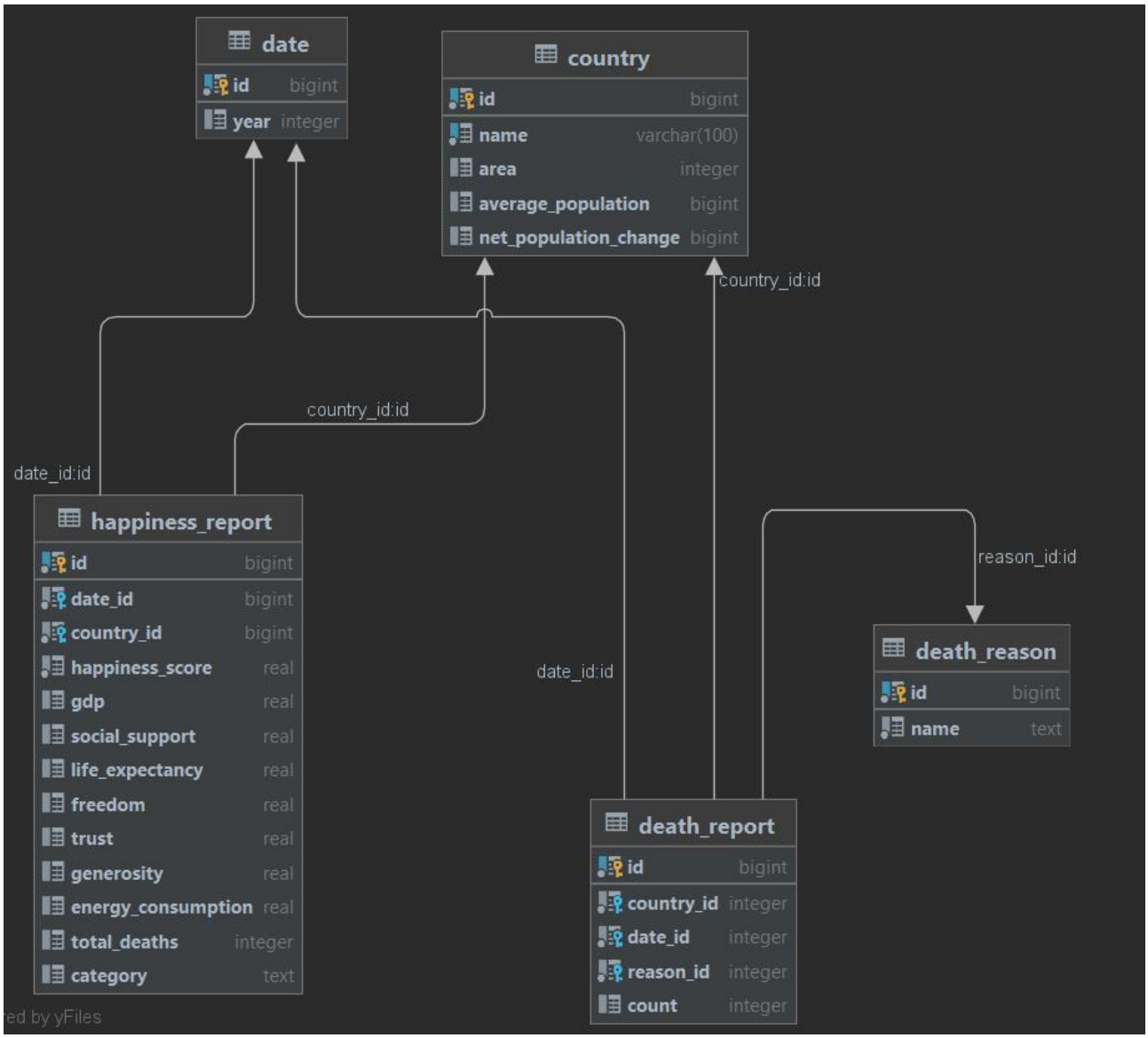

Рисунок 3.13 - Схема сховища даних

3.3. Завантаження даних до сховища

Наступним кроком завантажимо дані зі stage-зони до сховища даних, що було створено у попередньому пункті. Для цього використаємо наступні SQL-скрипти із вкладеними запитами.

Завантажимо дані до таблиці countries.

```
WITH countries AS (
    SELECT "Country" AS name FROM stagecoursework.happiness
    UNION
    SELECT name: "Country" AS name FROM stagecoursework.country_consumption
    UNION
    SELECT name: "Country" AS name FROM stagecoursework.death_reasons
    UNION
    SELECT name: "Country" AS name FROM stagecoursework.population_by_country)
INSERT INTO adiscoursework.country (name, area, average_population, net_population_change)
SELECT DISTINCT (name),
                (select "Land Area (Km²)" AS area from stagecoursework.population_by_country where name = "Country"),
                (select "Population" from stagecoursework.population_by_country where name = "Country"),
                (select "Net Change" from stagecoursework.population_by_country where name = "Country")
FROM countries;
```

Рисунок 3.14 - Завантаження даних до таблиці countries

Завантажимо дані до таблиці years.

```
WITH years AS (
    SELECT "Year" AS year FROM stagecoursework.happiness
    UNION
    SELECT year: "Year" AS year FROM stagecoursework.country_consumption
    UNION
    SELECT year: "Year" AS year FROM stagecoursework.death_reasons
)
INSERT INTO adiscoursework.date (year)
SELECT DISTINCT(year) FROM years;
```

Рисунок 3.15 - Завантаження даних до таблиці years

Завантажимо дані до таблиці happiness_report, використовуючи вкладені запити.

```
INSERT INTO adiscoursework.happiness_report(
                                        date_id,
                                        country_id,
                                        happiness_score,
                                        gdp,
                                        social_support,
                                        life_expectancy,
                                        freedom,
                                        trust,
                                        generosity,
                                        energy_consumption,
                                        category)
SELECT (SELECT id FROM adiscoursework.date WHERE "Year" = year),
       (SELECT id FROM adiscoursework.country WHERE "Country" = name),
       "Happiness Score",
       "Economy (GDP per Capita)",
       "Family (Social Support)",
       "Health (Life Expectancy)",
       "Freedom",
       "Trust (Government Corruption)",
       "Generosity",
       (SELECT "Consumption" FROM stagecoursework.country_consumption
       WHERE country_consumption."Year" = happiness."Year"
         AND country_consumption."Country" = happiness."Country"),
       (SELECT "Country Classification"
        FROM stagecoursework.country_indicators
        WHERE stagecoursework.country_indicators."Year" = happiness."Year"
          AND stagecoursework.country_indicators."Country" = happiness."Country")
FROM stagecoursework.happiness;
```

Рисунок 3.16 - Завантаження даних до таблиці happiness_report

Завантажимо дані до таблиць death_reason і death_report, використовуючи вкладені запити.

```
INSERT INTO adiscoursework.death_reason(name)
SELECT DISTINCT("Reason") FROM stagecoursework.death_reasons;

INSERT INTO adiscoursework.death_report(country_id, date_id, reason_id, count)
SELECT (SELECT id FROM adiscoursework.country WHERE "Country" = name),
       (SELECT id FROM adiscoursework.date WHERE "Year" = year),
       (SELECT id FROM adiscoursework.death_reason WHERE "Reason" = name),
       "Count"
FROM stagecoursework.death_reasons;
```

Рисунок 3.17 - Завантаження даних до таблиці death_reason і death_report

Отже, результатом роботи над цим розділом стало створення сховища даних і також його заповнення даними з датасетів.

4. ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

4.1. Обґрунтування алгоритмів для побудови регресійної моделі

Для побудови регресійної моделі ми використовуємо лінійну і поліноміальну регресії.

Регресійний аналіз – це метод моделювання даних, які вимірюються, та дослідження їх властивостей. Регресійна модель – це функція незалежної величини та коефіцієнтів з включеними випадковими змінними.

Вважають, що залежна змінна описується сумою значень деякої моделі та незалежними змінними. Відповідно до характеру розподілу залежної змінної роблять припущення, які називаються гіпотезою породження даних. Для підтвердження або спростування цієї гіпотези проводяться статистичні тести (аналіз залишків – різниця між значеннями, які спостерігаються, та значеннями, які передбаченні побудованою регресійною моделлю). При цьому вважають, що залежна змінна не містить помилок.

Нами було обрано лінійну та поліноміальну регресії.

Лінійна регресійна модель має наступний вигляд:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k + u \qquad (4.1)$$

де y – залежна змінна;

$(x_1, x_2, ..., x_n)$ – незалежні змінні;

u – випадкова похибка, розподіл якої в загальному випадку залежить від незалежних змінних, але математичне очікування якої рівне нулю.

Нелінійна регресія – окремий випадок регресійного аналізу, в якому розглянутою регресійною моделлю є нелінійна функція, що залежить від параметрів і від однієї або декількох вільних змінних. Відмінність від лінійної регресії полягає тільки в формі зв’язку та методах оцінки параметрів (формула самої регресійної функції. Що призвана оцінювати дані).

Оскільки точність методу лінійної та поліноміальної регресій є достатньою для більшості задач, що виникають у житті, а реалізація відносно простою, то наш вибір є небезпідставним.

Модель завжди є спрощенням реальності, тому вона повинна бути досить проста. З двох моделей, що приблизно однаково відповідають даним, перевагу варто віддати більш простій моделі, що містить, наприклад, менше число пояснюючих змінних.

Оцінити точність регресійної моделі можна за критеріями R^2 (коефіцієнт детермінації) і RSE (стандартна похибка залишків), що дасть нам змогу зрозуміти, чи достатньо математична модель описує життєву.

Формула для розрахунку стандартної похибки залишків:

$$RSE = \arg\min \sum_{i=1}^{n} \left| y_i - \beta_0 - \sum_{j=1}^{k} X_{ij}\beta_j \right|^2 = \arg\min_{\beta} \left\| y = X\beta \right\|^2 \quad (4.2)$$

Формула для розрахунку значення коефіцієнту детермінації:

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} \quad (4.3)$$

- де RSS - сума квадратів залишків, TSS - загальна сума квадратів відхилень.

4.2. Побудова і тренування моделі

Для побудови моделі нами було розроблено функцію regression, що одразу генерує, навчає, тестує та оцінює якість регресійної моделі.

```
def regression(x, y, max_degree):
    Xtrain, Xtest, Ytrain, Ytest = \
        train_test_split(x, y, test_size=0.3, random_state=4)
    reg = make_pipeline(PolynomialFeatures(degree=max_degree),
                        LinearRegression())
    reg.fit(Xtrain, Ytrain)
    Ypredicted = []
    for i in range(0, Xtest.__len__()):
        Ypredicted.append(reg.predict([Xtest[i]])[0])

    R2 = r2_score(Ytest, Ypredicted)
    _RSE = RSE(Ytest, Ypredicted)
    coef = reg['linearregression'].coef_
    w0 = reg['linearregression'].intercept_

    return coef, w0, R2, reg, _RSE
```

Рисунок 4.1 - Вигляд функції regression

Для побудови моделі спочатку розіб'ємо задану вибірку на навчальну та тестову частини у відношенні 3:7 відповідно. Надалі навчальну вибірку використовуватимемо для навчання моделі, а тестову - для перевірки якості моделі на нових для неї даних задля чистоти експерименту. У результаті роботи функції отримали такі змінні:

- Xtrain - предиктори навчальної вибірки
- Ytrain - результати навчальної вибірки
- Xtest - предиктори тестової вибірки
- Ytest - результати тестової вибірки

```
Xtrain, Xtest, Ytrain, Ytest = \
    train_test_split(x, y, test_size=0.3, random_state=4)
```

Рисунок 4.2 - Розбиття вибірки на навчальну та тестову

```
reg = make_pipeline(PolynomialFeatures(degree=max_degree),
                    LinearRegression())
reg.fit(Xtrain, Ytrain)
```

Рисунок 4.3 - Генерація поліному степеня max_degree та навчання моделі на даних навчальної вибірки

Для кожного X із тестової вибірки Xtest розрахуємо та збережемо передбачене значення за допомогою нашої функції регресії.

```
Ypredicted = []
for i in range(0, Xtest.__len__()):
    Ypredicted.append(reg.predict([Xtest[i]])[0])
```

Рисунок 4.4 - Розрахунок та збереження передбачених значень

4.3. Валідація моделі

Під валідацією моделі розуміється оцінка точності прогнозування за вищезгаданими критеріями R^2 та RSE.

```
R2 = r2_score(Ytest, Ypredicted)
_RSE = RSE(Ytest, Ypredicted)
coef = reg['linearregression'].coef_

return coef, R2, reg, _RSE
```

Рисунок 4.5 - Оцінка моделі

```
def RSE(y_true, y_predicted):
    """
    - y_true: Actual values
    - y_predicted: Predicted values
    """
    y_true = np.array(y_true)
    y_predicted = np.array(y_predicted)
    RSS = np.sum(np.square(y_true - y_predicted))

    rse = math.sqrt(RSS / (len(y_true) - 2))
    return rse
```

Рисунок 4.6 - Функція для оцінки моделі RSE

Далі нами було розроблено функцію, що розбиває датасет на предиктори і значення та запускає процес побудови моделей для різних степенів для подальшого порівняння між собой та вибору тої, що підходить найбільше.

```
def main_work(df):
    x = []
    for i in range(0, df.__len__()):
        tmp = df.loc[i].to_numpy()
        tmp = np.delete(tmp, df.columns.size - 1, 0)
        x.append(tmp)
    y = df['happiness_score'].to_numpy()

    for i in range(1, 5):
        coef, R2, reg, _RSE = regression(x, y, i)
        print('---------------------------------------')
        print(f'degree {i} R^2: ', R2)
        print(f'degree {i} RSE: ', _RSE)
        print('---------------------------------------')
        for col in df.columns:
            if col != 'happiness_score':
                printProjection(df, reg, col, 1000)
        plt.show()
    pass
```

Рисунок 4.7 - Функція main_work

Із результатів роботи регресії різних степенів очевидно, що поліноміальна регресія степеня 3 має найбільшу оцінку R^2 та найменшу оцінку RSE, що вказує на те, що саме ця модель є найбільш точною серед перелічених. Отже, виберемо її.

```
----------------------------------------
degree 1 R^2:  0.7383436653893025
degree 1 RSE:  0.597671674573672
----------------------------------------
----------------------------------------
degree 2 R^2:  0.7697042088428689
degree 2 RSE:  0.5607122542740632
----------------------------------------
----------------------------------------
degree 3 R^2:  0.8010696282679126
degree 3 RSE:  0.5211318230119327
----------------------------------------
----------------------------------------
degree 4 R^2:  -1.4270554310237777
degree 4 RSE:  1.8202745402855154
----------------------------------------
----------------------------------------
degree 5 R^2:  -3397011200.940698
degree 5 RSE:  68099.75299111061
----------------------------------------
```

Рисунок 4.8 - Результат роботи функції main_work

4.4. Візуалізація результатів регресійного аналізу

Для зручності оцінки адекватності моделі нами біло прийнято рішення створення функції, що зображає двовимірні проекції залежності рівня щастя від предикторів. На графіку буде зображено як емпіричні дані, так і проекцію лінії регресії. Хочеться зауважити, що зображено буде саме проекцію лінії регресії на площину, а не саму лінію, оскільки через велику кількість параметрів візуально зобразити сам графік регресії неможливо.

```
def printProjection(df, regression, argument_name, detalization):
    def fillZeros(list_length):
        list = []
        for i in range(list_length):
            list.append(0)
        return list

    xReg = []
    argument_position = df.columns.tolist().index(argument_name)
    max_value = df[argument_name].max()
    step = max_value / detalization
    arg_value = 0
    while arg_value <= max_value:
        tmp = fillZeros(df.columns.size - 1)
        tmp[argument_position] = arg_value
        xReg.append(tmp)
        arg_value += step

    plt.figure(figsize=(5, 5))
    plt.title('Regression')
    plt.xlabel(argument_name)
    plt.ylabel('happiness')
    plt.grid(linestyle='--')

    plt.plot(
        np.linspace(0, max_value, len(xReg)).reshape(-1, 1),
        regression.predict(xReg),
        color='red')
    plt.scatter(df[argument_name].to_numpy(), df['happiness_score'])
    pass
```

Рисунок 4.9 - Функція для зображення проекції

Замість усіх предикторів, крім того, що бере участь у проекції, вставляємо нулі та викликаємо функцію передбачення для регресії. Враховуючи те, що усі параметри, що є предикторами, за умовою набувають лише додатніх значень, то такий трюк є цілком припустимим.

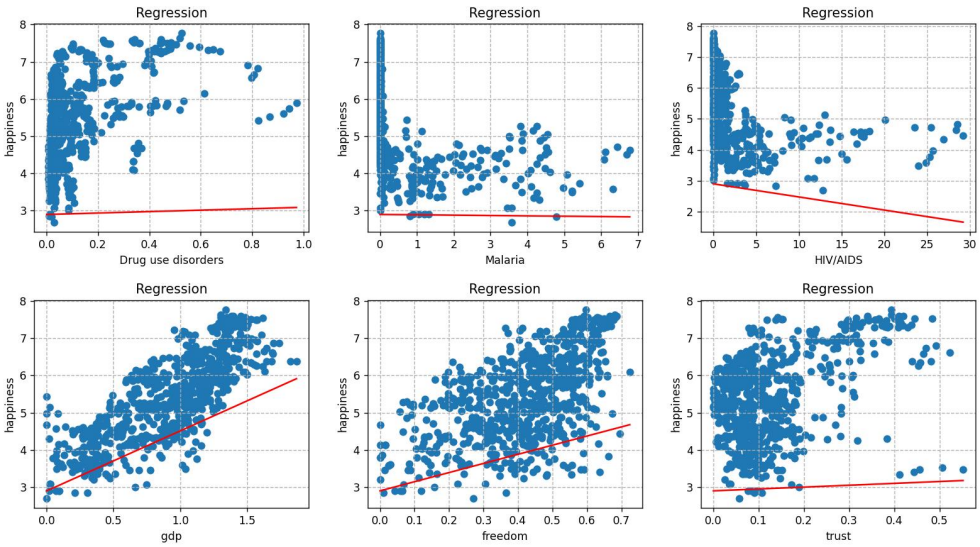

Рисунок 4.10 - Графічне відображення проекції лінійної регресії

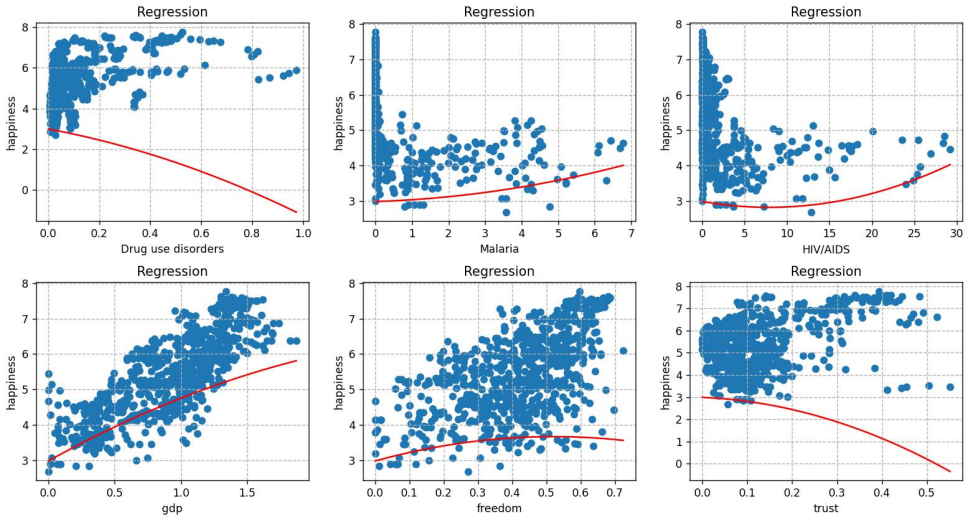

Рисунок 4.11 - Графічне відображення проекції поліноміальної регресії степеня 2

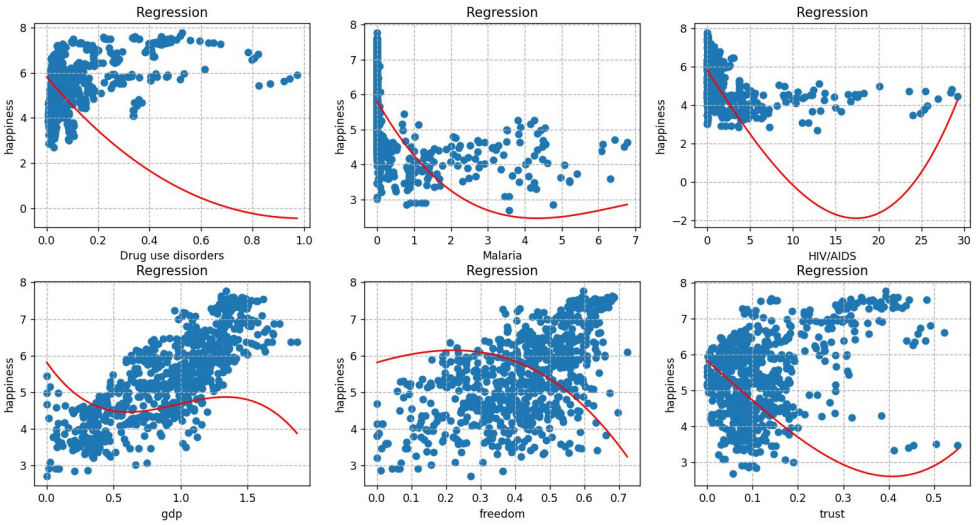

Рисунок 4.12 - Графічне відображення проекції поліноміальної регресії степеня 3

Отже, результатом роботи над цим розділом стала побудована модель регресійного аналізу, причому серед усіх протестованих вибрана найбільш точна. Досягнуто точності 80% за критерієм R^2.

5. ВИСНОВОК

В результаті виконання курсової роботи було розроблено алгоритм, який допомагає прогнозувати рівень щастя населення країн світу за вказаними соціологічними параметрами. Розглянуто основні підходи для реалізація такої моделі. Для реалізація поставленої задачі було використано мову програмування Python та різні бібліотеки: pandas, numpy, matlplot, seaborn, scikitlearn та інші.

У ході роботи було створено сховище даних, розроблено та застосовано ETL-процеси мовами SQL та Python. Усі скрипти було детально документовано у цьому звіті та проілюстровано наочними графіками.

Було розроблено код, що передбачає за значеннями параметрів рівень щастя населення країни. Було реалізовано кілька моделей регресійного аналізу передбачення:

- лінійна регресія
- поліноміальна регресія ступеня 2
- поліноміальна регресія ступеня 3
- поліноміальна регресія ступеня 4

Як показали оцінки точності та графіки, найбільш прийнятною виявилася модель поліноміальної регресії степеня 3. На основі детального опису та проведеного аналізу предметної області інтелектуального аналізу даних для визначення рівня щастя населення було отримано результати з високою точністю передбачення. Підтвердженням даних висновків є результати точності, яка складає до 0.72-0.75 для найбільш оптимального варіанту регресії на тестовому наборі даних. Результати досліджень показують, що дана модель може застосовуватися за призначенням і приносити користь, наприклад, соціологам чи економістам, у завдання яких як раз входять завдання такого плану. Отже, поставлені задачі були виконані.

ПЕРЕЛІК ПОСИЛАНЬ

1. Лекційні матеріали
2. Machine Learning. Coursera. Author: Andrew Ng
3. https://matplotlib.org /stable/tutorials/index
4. https://seaborn.pydata.org/generated/seaborn.heatmap.html
5. https://pandas.pydata.org/docs/
6. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ

https://github.com/GeniusDP/CursachDataAnalysis - public repository with resources and code.