МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра

Інформатики та програмної інженерії

(повна назва кафедри, циклової комісії)

**КУРСОВА РОБОТА**

з дисципліни «Аналіз даних в інформаційних системах»

(назва дисципліни)

на тему: «Прогнозування щастя країни за показниками терористичних актів та клімату»

Студента 2 курсу, групи ІП-15

Спеціальності 121 «Інженерія програмного забезпечення»

Мєшкова Андрія Ігоровича

«ПРИЙНЯВ» з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Підпис Дата

Київ - 2023 рік

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

(назва вищого навчального закладу)

Кафедра інформатики та програмної інженерії

Дисципліна Аналіз даних в інформаційних системах

Спеціальність 121 «Інженерія програмного забезпечення»

Курс 2 Група ІП-15 Семестр 4

**ЗАВДАННЯ**

на курсову роботу студента

Мєшкова А.І.

(прізвище, ім’я, по батькові)

1. Тема роботи Прогнозування щастя країни за показниками терористичних актів та клімату

2. Строк здачі студентом закінченої роботи 08.06.2023

3. Вхідні дані до роботи методичні вказівки до курсової роботи, дані з сайтів

https://www.kaggle.com/datasets/ajaypalsinghlo/world-happiness-%20report-2021?select=world-happiness-report.csv

<https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data?select=GlobalLandTemperaturesByCountry.csv>

https://www.kaggle.com/datasets/START-UMD/gtd

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

Постановка задачі

Аналіз предметної області

Розробка сховища даних

Інтелектуальний аналіз даних

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

6. Дата видачі завдання 16.04.2023

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Назва етапів курсової роботи | Термін виконання етапів роботи | Підписи керівника, студента |
| 1. | Отримання теми курсової роботи | 30.03.23 |  |
| 2. | Визначення зовнішніх джерел даних | 02.04.23 |  |
| 3. | Пошук та вивчення літератури з питань курсової роботи | 02.05.23 |  |
| 4. | Розробка моделі сховища даних | 19.05.23 |  |
| 6. | Обґрунтування методів аналізу даних | 20.05.23 |  |
| 5. | Застосування та порівняння ефективності методів інтелектуального аналізу | 06.06.23 |  |
| 6. | Підготовка пояснювальної записки | 07.06.23 |  |
| 7. | Здача курсової роботи на перевірку | 08.06.23 |  |
| 8. | Захист курсової роботи | 09.06.23 |  |

Студент Мєшков А.І.

(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)

"16" березня 2023 р.

АНОТАЦІЯ

Пояснювальна записка до курсової роботи: 41 сторінка, 25 рисунків, 2 таблиці, 8 посилань.

Об’єкт дослідження: суспільно-економічні фактори та кліматичні умови країн.

Предмет дослідження: взаємозв'язок між показниками терористичних актів, кліматичними умовами та рівнем щастя країн.

Мета роботи: виявити вплив наявності терористичних актів та кліматичних умов на рівень щастя країн і розробити прогностичну модель для прогнозування щастя країн на основі цих факторів.

Дана курсова робота включає в себе: постановку задачі, аналіз предметної області, створення та заповнення сховища даних, аналіз обраних методів для прогнозування та їх порівняння.

СХОВИЩЕ ДАНИХ, МОДЕЛЬ ПРОГНОЗУВАННЯ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, ЛІНІЙНА РЕГРЕСІЯ, ПОЛІНОМІАЛЬНА РЕГРЕСІЯ.

Зміст

[Вступ 5](#_Toc137067029)

[1 Постановка задачі 6](#_Toc137067030)

[2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 7](#_Toc137067031)

[3 розробка СХОВИЩА ДАНИХ 8](#_Toc137067032)

[3.1. Визначення даних 8](#_Toc137067033)

[3.2. Модель Stage зони для ETL процесів 10](#_Toc137067034)

[3.3. Модель основного сховища за типом «зірка» 11](#_Toc137067035)

[3.4. ETL засоби 13](#_Toc137067036)

[3.5. Занесення даних до основного сховища 16](#_Toc137067037)

[3.6. Підготовка даних до виконання задачі 18](#_Toc137067038)

[3.7. Поділ даних 20](#_Toc137067039)

[4 ІНТЕЛЕКтуальний аналіз даних 21](#_Toc137067040)

[4.1. Обґрунтування вибору методів інтелектуального аналізу даних 21](#_Toc137067041)

[4.2. Підготовка даних до інтелектуального аналізу 22](#_Toc137067042)

[4.3. Тестування методів 24](#_Toc137067043)

[4.4. Порівняння методів 26](#_Toc137067044)

[Висновки 29](#_Toc137067045)

[Перелік посилань 30](#_Toc137067046)

[Додаток А Тексти програмного коду 31](#_Toc137067047)

Вступ

Прогнозування щастя країни є важливою та актуальною проблемою, яка вимагає глибокого аналізу різноманітних факторів. Одним із ключових аспектів, які можуть впливати на щастя населення, є рівень терористичних актів. Терористична діяльність має потенціал спричинити страх, тривогу та нестабільність, що негативно впливає на життя громадян та соціальну ситуацію в країні. Крім того, інший фактор, який може мати велике значення для рівня щастя населення, - це кліматичні умови. Клімат є важливою складовою як фізичного, так і психологічного благополуччя людей. Екстремальні погодні явища, висока вологість чи паводки можуть викликати стрес та незручності для населення, а сприятливі кліматичні умови можуть позитивно впливати на загальний настрій та самопочуття людей.

У рамках даної курсової роботи були проаналізовані дані про погодні умови в країнах, а саме середня температура повітря та невизначеність середньої температури, та інформація про наявність актів тероризмом з можливою довготривалістю, на основі отриманих даних було використано декілька методів для прогнозування щастя країни, для оцінки щастя було обрано життєву драбину.

В рамках даної курсової роботи було розроблено сховище даних на основі фізичної моделі бази даних, функціонал якої було розроблено та заповнено за допомогою SQL скриптів та програмне забезпечення, що отримує дані з неї, проводить інтелектуальний аналіз даних та прогнозує результат.

В ролі системи керування сховищем даних для даної роботи буде виступати MySQL[1], а мова програмування для реалізації застосунку – Python3[2].

# Постановка задачі

Під час виконання курсової роботи потрібно виконати наступні завдання:

Аналіз предметної області.

Створення сховища даних для аналізу.

Робота з датасетом: завантаження, дослідження його структури та виправлення наявних помилок.

Вибір методів для прогнозування та обґрунтування даного вибору.

Аналіз отриманих результатів кожного з методів та порівняння отриманих результатів ефективності.

Створення застосунку, що отримує дані з бази даних та проводить інтелектуальний аналіз способом лінійної регресії та поліноміальної регресії, що тестує моделі прогнозування. Обрати найоптимальніший метод для прогнозування щастя країни.

Вхідними даними будуть середня температура повітря, невизначеність середньої температури, наявність терористичного акту та розширення, тривання події більш ніж 24 години.

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Щастя країни є важливою темою для бізнесу з кількох причин.

По-перше, це допомагає бізнес-організаціям управляти ризиками. Нестабільна політична ситуація та зміни в кліматичних умовах можуть створювати ризики для бізнесу. Прогнозування щастя країни на основі таких факторів, як терористичні акти і кліматичні зміни, дозволяє бізнесу оцінити ці ризики та прийняти відповідні заходи для їх управління.

По-друге, моя тема може бути важливим фактором при прийнятті інвестиційних рішень. Багато бізнес-організацій беруть до уваги соціально-економічні фактори, коли розглядають інвестиційні можливості. Визначення щастя країни може служити важливим показником для оцінки потенційної доходності та стабільності інвестиційних проектів.

По-третє, дані можуть бути корисними для маркетингових стратегій та розуміння споживчої поведінки. Рівень щастя та задоволення населення можуть впливати на споживчі попити та вподобання. Бізнес-організації можуть використовувати такі дані, щоб краще розуміти настрої споживачів та адаптувати свої маркетингові стратегії.

Нарешті, прогнозування може сприяти розвитку корпоративної соціальної відповідальності. Багато компаній активно займаються ініціативами соціального покращення в регіонах, де вони працюють. Прогноз може служити орієнтиром для визначення напрямків цих ініціатив та програм, спрямованих на поліпшення якості життя та благополуччя населення.

У програмному забезпеченні буде реалізовано наступну функціональність:

* завантаження та дослідження структури даних;
* використання декількох моделей прогнозування даних;
* прогнозування за обраними характеристиками клімату та тероризму;
* відображення отриманих результатів та їх аналіз;
* порівняння використаних методів.

# розробка СХОВИЩА ДАНИХ

## Визначення даних

Для виконання курсової роботи було обрано 3 джерела відкритих даних на сайті <https://www.kaggle.com//>. А саме:

* Звіт про щастя у світі: https://www.kaggle.com/datasets/ajaypalsinghlo/world-happiness-report-2021?select=world-happiness-report.csv
* Зміна клімату: дані про температуру поверхні Землі: https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data?select=GlobalLandTemperaturesByCountry.csv
* Тероризм: https://www.kaggle.com/datasets/START-UMD/gtd

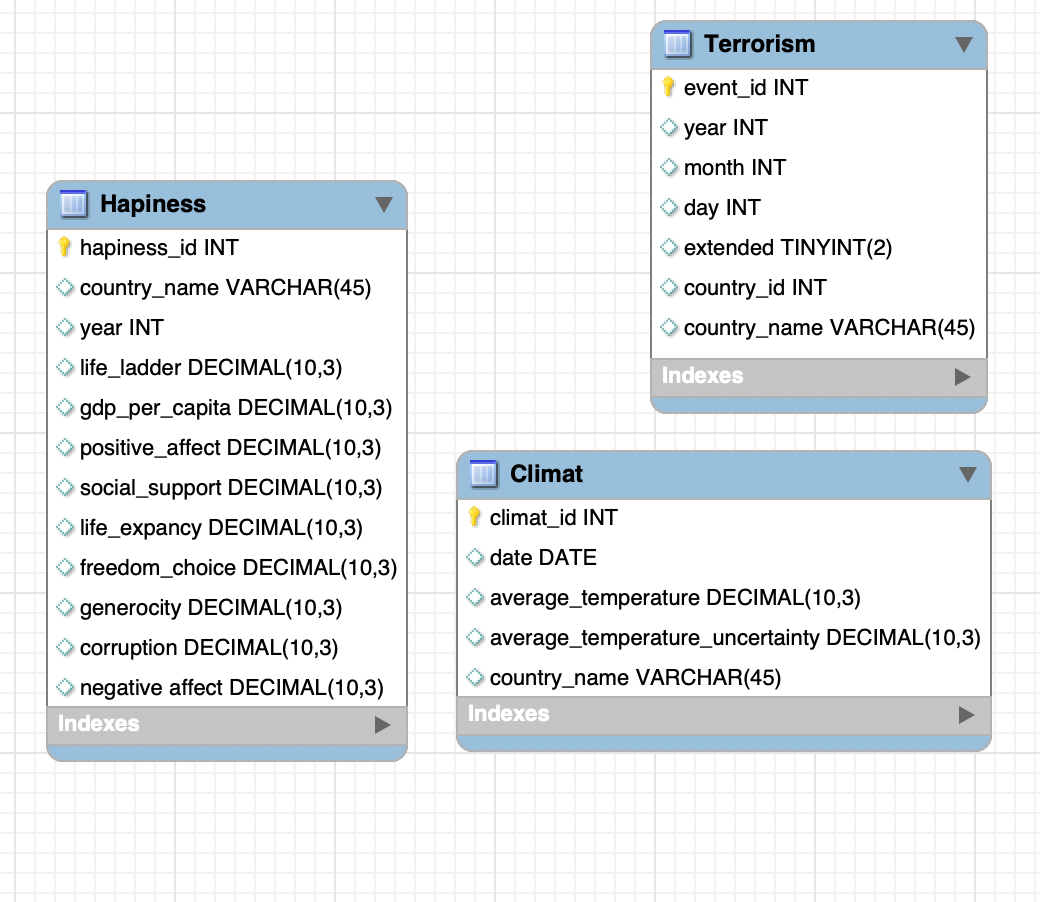
Нижче наведені поля для кожного з файлів, які безпосередньо використовувались у подальшій побудові бізнес-процесів:

*Таблиця 3.1 - поля вхідних файлів*

|  |  |  |
| --- | --- | --- |
| world-happiness-report.csv | country\_name | Назва країни |
| year | Рік |
| life\_ladder | Життєва драбина |
| gdp\_per\_capita | ВВП на душу населення |
| social\_support | Соціальна допомога |
| life\_expancy | Очікувана здорова тривалість життя при народженні |
| freedom\_choice | Свобода робити життєвий вибір |
| generocity | Щедрість |
| corruption | Сприйняття корупції |
| positive\_affect | Позитивний ефект |
| negative affect | Негативний ефект |
| GlobalLandTemperaturesBy-Country.csv | date | Дата |
| average\_temperature | Середня температура |
| average\_temperature\_uncertainty | Невизначеність середньої температури |
| country\_name | Назва країни |
| globalterrorismdb\_0718-dist.csv | event\_id | Номер події |
| year | Рік |
| month | Місяць |
| day | День |
| extended | Розширення, тривання події більш ніж 24 години |
| country\_id | Номер країни |
| country\_name | Назва країни |

## Модель Stage зони для ETL процесів

В результаті розробки була спроектована схема stage-зони, яка зображена на рисунку 2.1. Дана модель відображає таблиці для даних із вхідних джерел.



*Рисунок 3.1 – Stage зона для ETL процесів*

Скрипти створення Stage зони знаходяться у додатку А, StageZone.sql.

Опис таблиць stage зони:

1. Hapiness – призначена для зберігання інформації про щастя країн в різні роки.
2. Terrorism – призначена для зберігання інформації про терористичні акти в різних країнах в різні роки.
3. Climat – призначена для зберігання інформації про зміну температури в країнах.

## Модель основного сховища за типом «зірка»

У процесі розробки моделі сховища даних було створено одну таблицю фактів та 4 таблиці вимірів:

-dim\_climat – таблиця виміру клімату за температурою;

-dim\_terrorism – таблиця виміру терористичних атак;

-dim\_date – таблиця виміру дати;

-dim\_country – таблиця виміру країни;

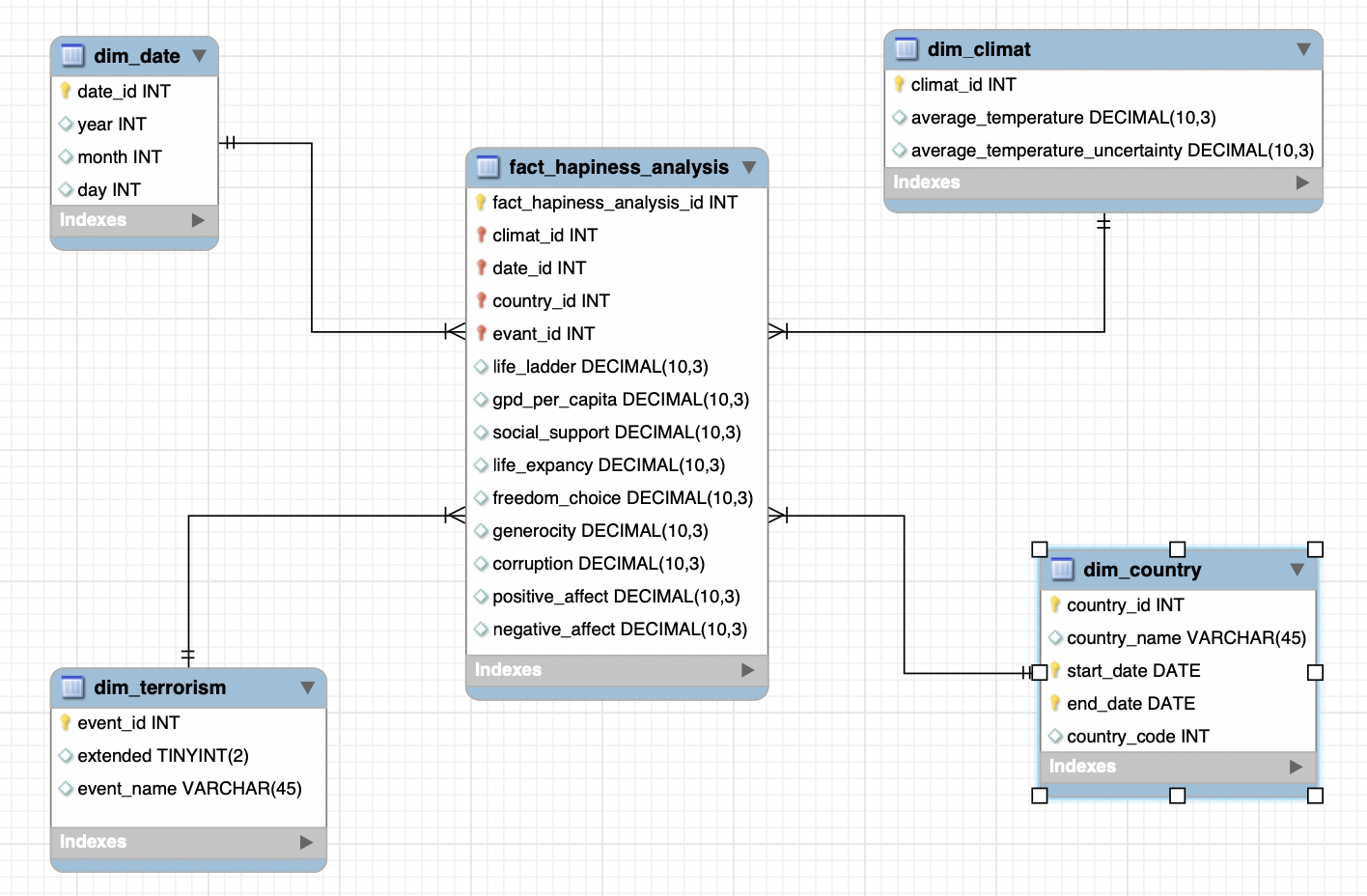
-fact\_hapiness\_analysis – таблиця фактів щастя в точці часу та простору;

*Таблиця 3.2 – Таблиця атрибутів таблиць сховища даних*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назва таблиці** | **Назва атрибуту** | **Тип даних** | **Первинний ключ** |
| dim\_climat | climat\_id | INT | climat\_id |
| average\_temperature | DECIMAL(10,3) |
| average\_temperature\_ uncertainty | DECIMAL(10,3) |
| dim\_terrorism | event\_id | INT | event\_id |
| event\_name | VARCHAR(45) |
| extended | TINYINT(2) |
| dim\_date | date\_id | INT | date\_id |
| year | INT |
| month | INT |
| day | INT |
| dim\_country | country\_id | INT | country\_id |
| country\_code | INT |
| country\_name | VARCHAR(45) |
| fact\_hapiness\_ analysis | hapiness\_analysis\_id | INT | hapiness\_analysis\_id |
| happiness\_id | INT |
| climat\_id | INT |
| event\_id | INT |
| date\_id | INT |
| country\_id | INT |
| life\_ladder | DECIMAL(10,3) |
| gdp\_per\_capita | DECIMAL(10,3) |
| social\_support | DECIMAL(10,3) |
| life\_expancy | DECIMAL(10,3) |
| freedom\_choice | DECIMAL(10,3) |
| generocity | DECIMAL(10,3) |
| corruption | DECIMAL(10,3) |
| positive\_affect | DECIMAL(10,3) |
| negative affect | DECIMAL(10,3) |

*Продовження Таблиці 3.2*

В результаті була спроектована схема сховища даних, яка зображена на рисунку 3.2. Дана модель дозволяє описувати відповідні бізнес-процеси згідно предметній області. Для представлення даних була вибрана багатовимірна модель зі схемою “зірка”.



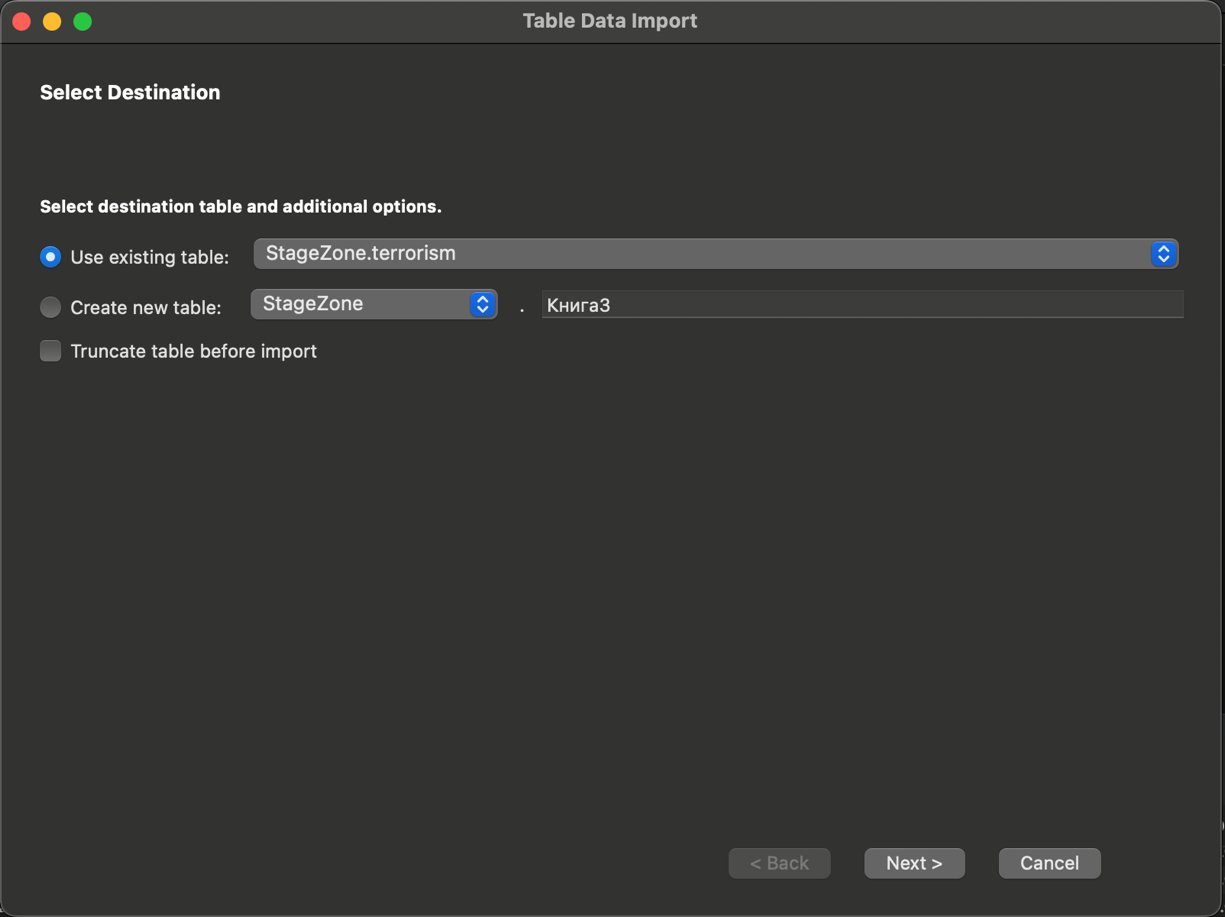
*Рисунок 3.2 – Сховище за типом «зірка»*

Скрипти створення основного сховища за типом «зірка» знаходяться у додатку А, DataWarehouse.sql.

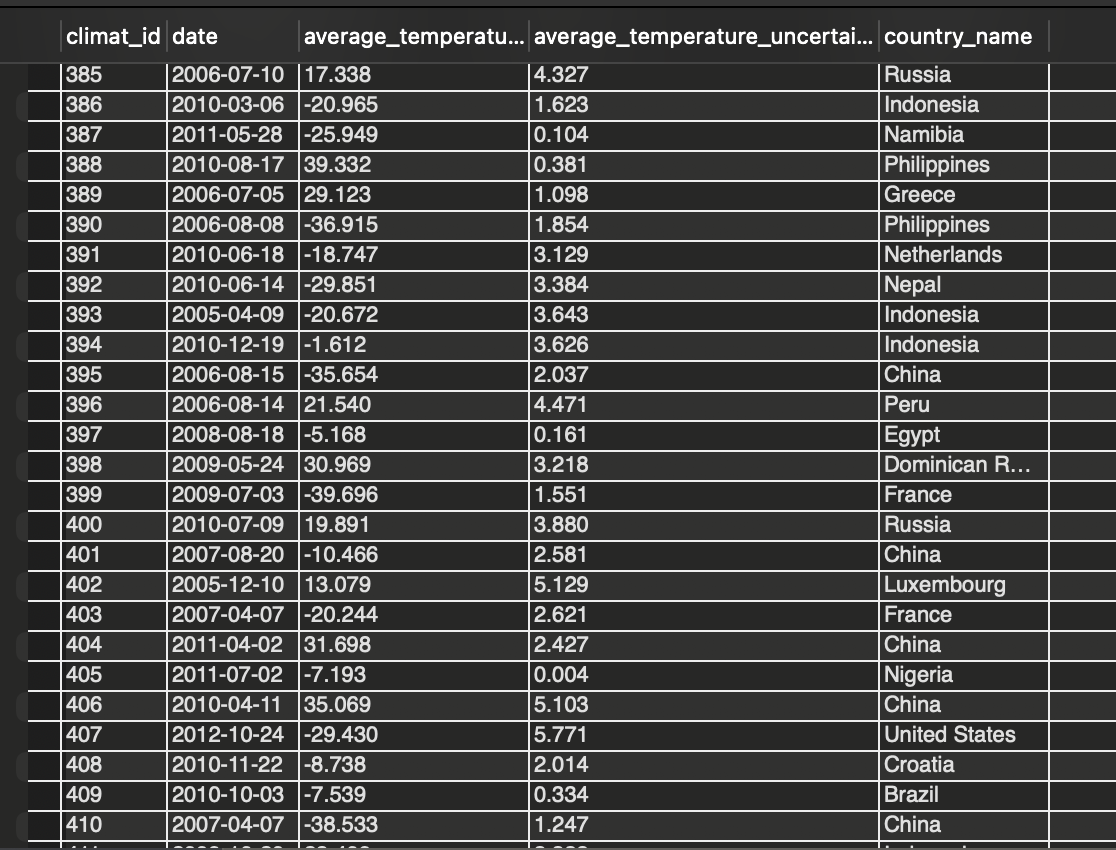
## ETL засоби

Скрипти ETL-процедур знаходяться у додатку А, ETL.sql

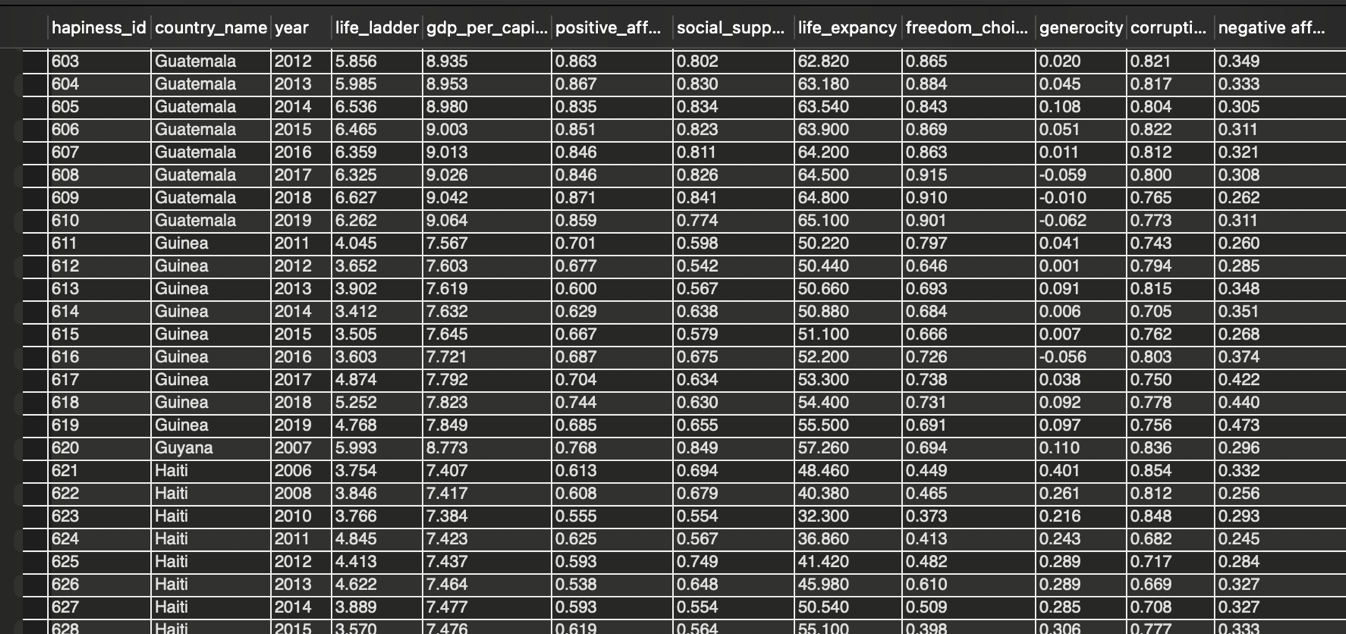
Спочатку ми здійснюємо завантаження даних до Stage зони.(рис. 3.3-3.6)



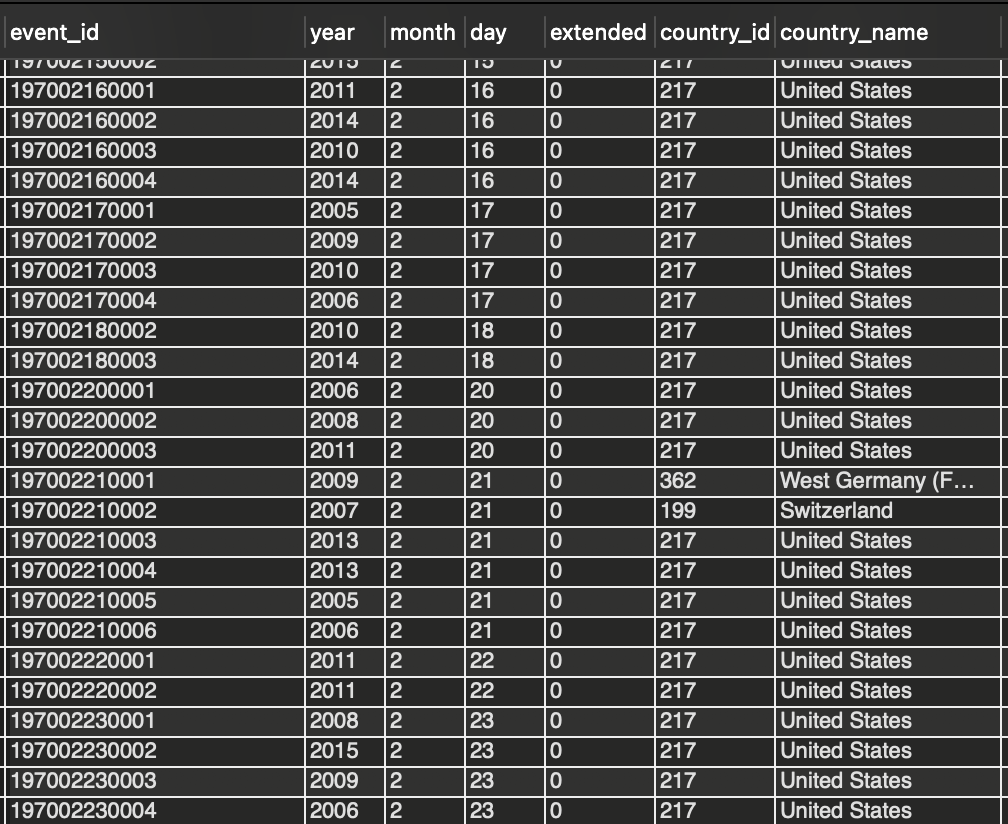
*Рисунок 3.3 – Імпорт даних*



*Рисунок 3.4 – Таблиця Climat*



*Рисунок 3.5 – Таблиця Hapiness*



*Рисунок 3.6 – Таблиця Terrorism*

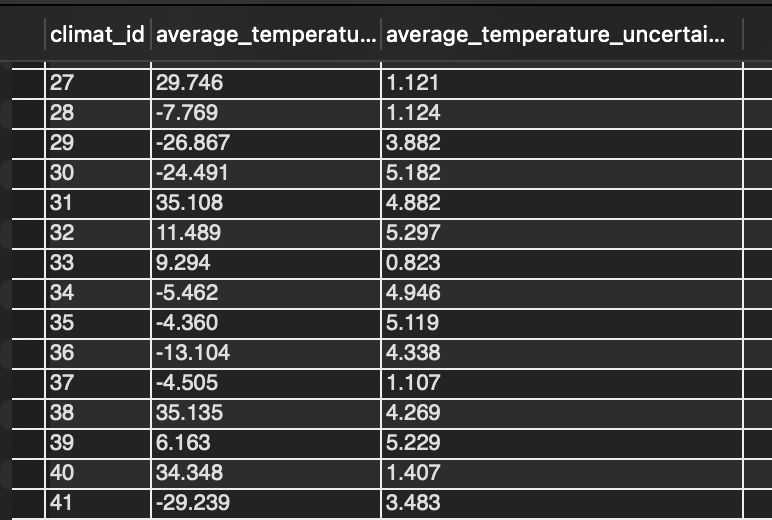
Cтворюємо набір процедур/функцій для перетворення та завантаження даних до основного сховища.

## Занесення даних до основного сховища

За допомогую скриптів дані були перенесені у основне сховище для подальшого аналізу даних. (рис. 3.7-3.11)



*Рисунок 3.7 – Таблиця dim\_country*



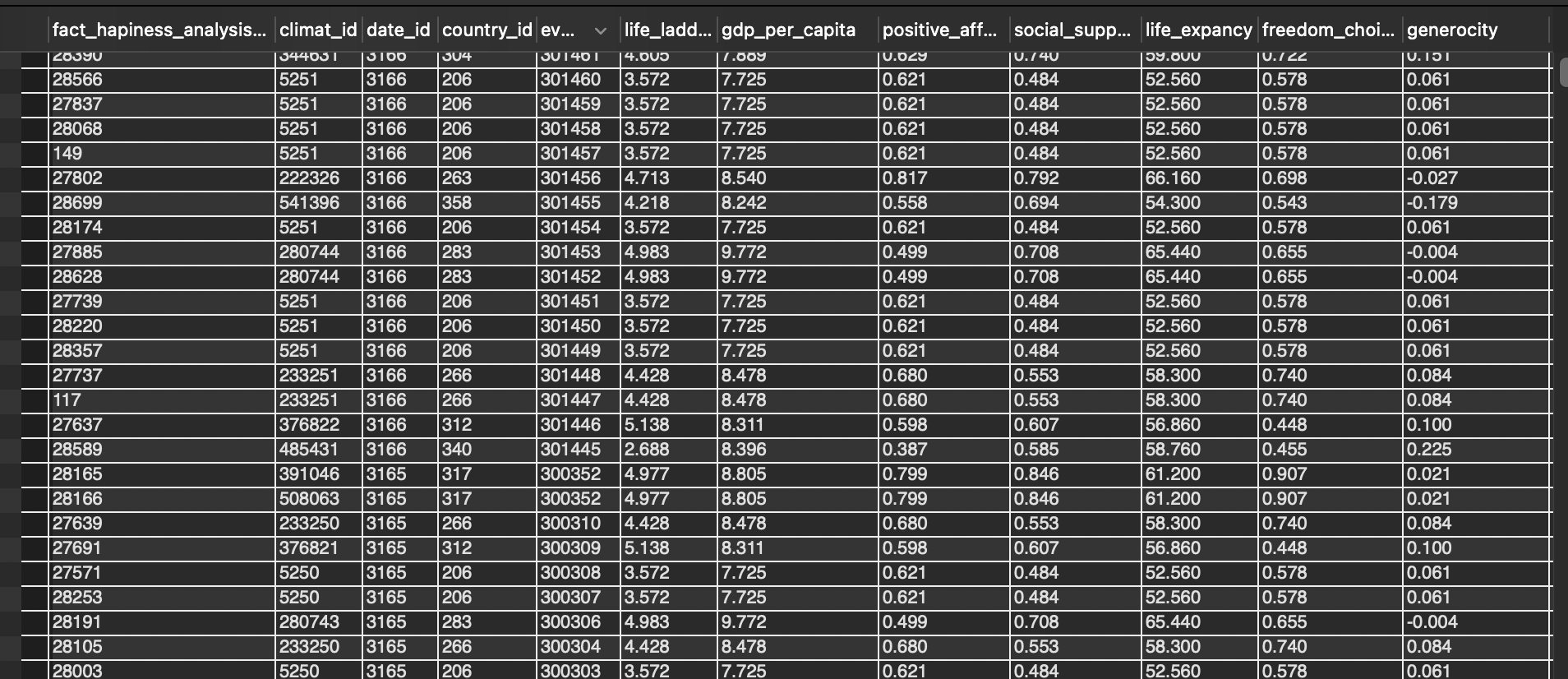
*Рисунок 3.8 – Таблиця dim\_climat*



*Рисунок 3.9 – Таблиця dim\_date*



*Рисунок 3.10 – Таблиця terrorism*



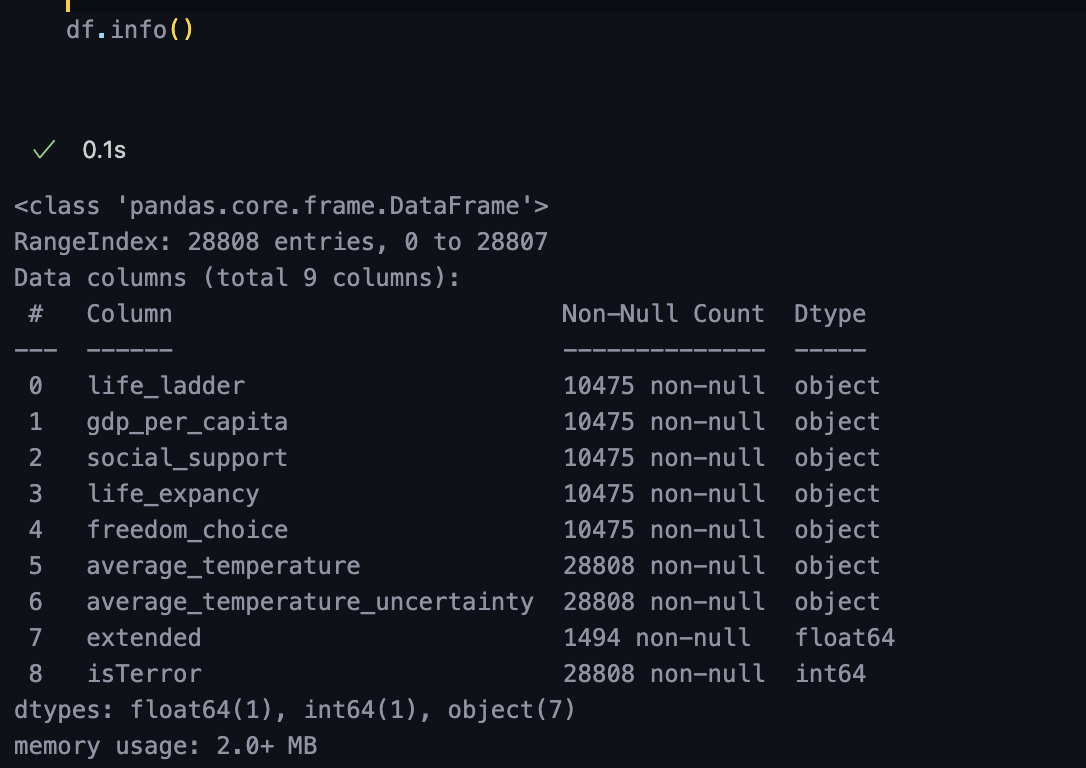
*Рисунок 3.11 – Таблиця fact\_hapiness\_analysis*

## Підготовка даних до виконання задачі

Було створено сховище даних. Для подальшого аналізу, ми повинні підготувати саме ті наді, які будуть брати участь у прогнозуванні. Для вхідних даних ми будемо використовувати інформацію: average\_temperature average\_temperature\_uncertainty, event\_name, extended. Для результату буде використано life\_ladder. Так як щастя – субʼєктивне поняття, воно може вимірюватись різними факторами, як одним і декількома. Для альтернативи можна би було використати gdp\_per\_capita, social\_support, life\_expancy, freedom\_choice, generocity, але для нашої моделі було використано саме ladder.

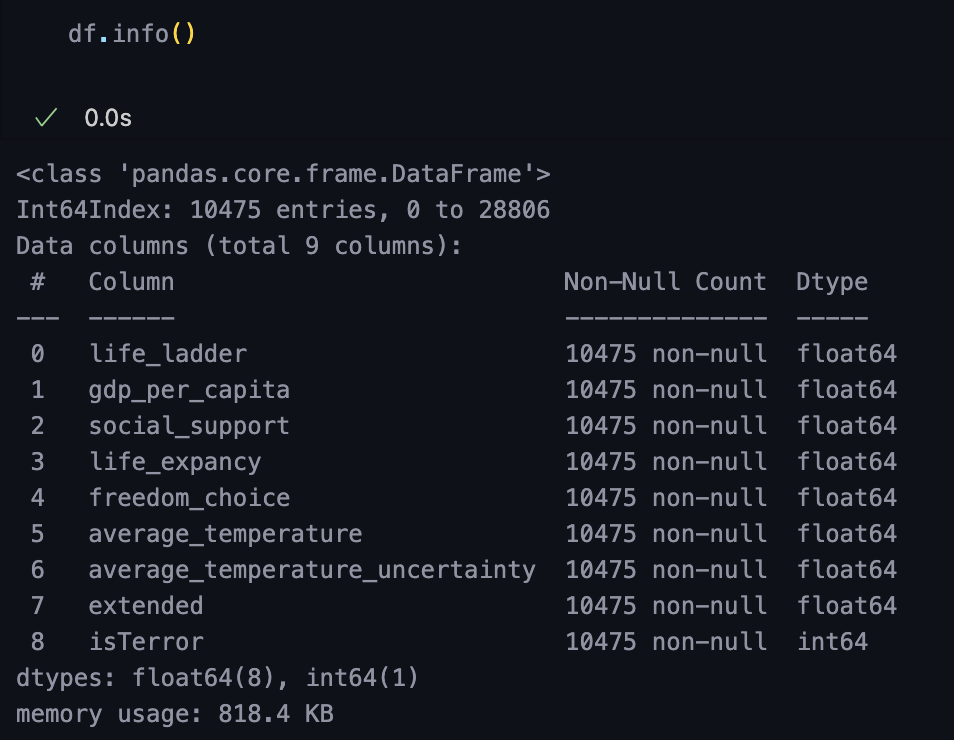
Для звʼязку між MySQL та Python було використано бібліотеку Pymysql. Сторивши звʼязок ми можеми отримати потрібну інформацію.

Використавши бібліотеку Pandas, ми зможемо легше аналізувати стркутуру. Одразу був створенний новий ствопець isTerror, який показує наявність тероризму, він буде замість event\_name, який ми видаляємо. Проаналізуємо структуру(рис 3.12)



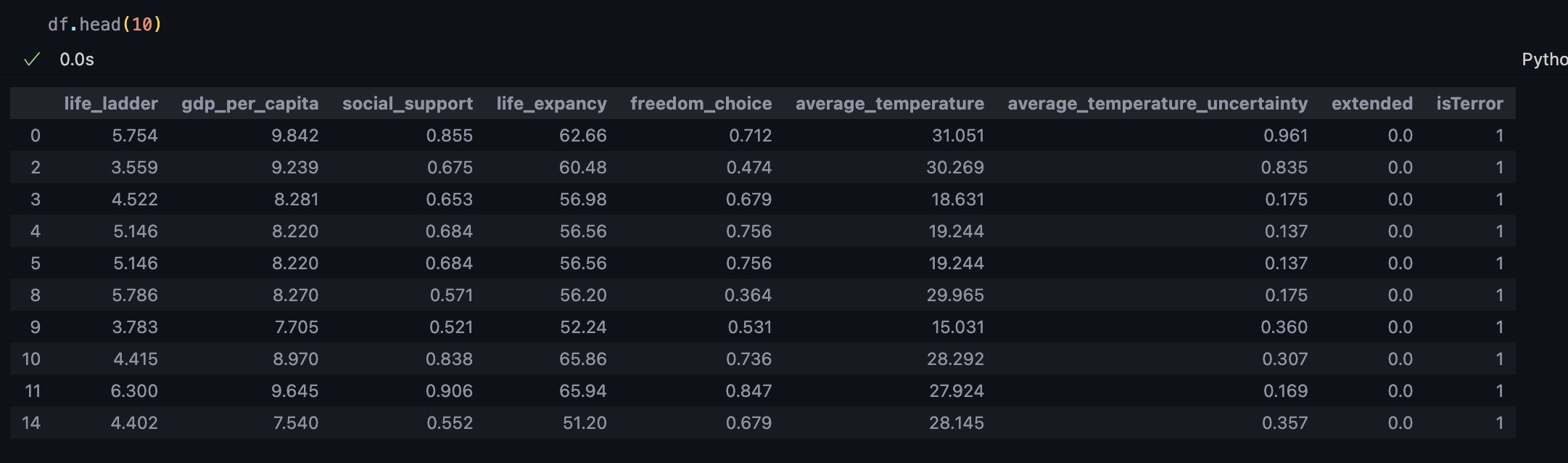
*Рисунок 3.12 – Загальна інформація про датафрейм*

Як ми можемо побачити дані мають пропущені значення, тому потрібно це виправити. Стовпець extended має менше всього значень, що означає, що підтвердження довготривалого акту тероризму нема, тому усі пропущені значення приймуть значення 0. Стовпці, які характеризують щастя, мають достатньо для аналізу даних, тому нульові значення можемо видалити, бо замінивши їх на середнє наприклад, нам складніше буде виявити залежності. (рис 3.13)



*Рисунок 3.13 – Виправлені дані*

Таким чином ми скоректували наші дані та підготували до математичних дій. Перевіримо вигляд наших даних (рис 3.14).



*Рисунок 3.14 – Датафрейм для подальшої роботи*

## Поділ даних

Останнім кроком ми ділимо наші дані на тестову та тренувальну вибірки для подальшої роботи з методами регресії (рис 3.15).



*Рисунок 3.15 – Поділ інформації на вхідні дані та результат*

Для уникнення перенавчання, ми використовували метод розділення набору даних на тренувальний і тестовий. Відсоток розподілу був 80% на тренувальний набір і 20% на тестовий набір. Це дозволяє нам отримати більш об'єктивне уявлення про ефективність наших методів на етапі тестування. Тестування проводиться на невидимих раніше даних, що дозволяє нам краще оцінити, наскільки наші методи вірно працюють для вирішення поставленої задачі.

# ІНТЕЛЕКтуальний аналіз даних

## Обґрунтування вибору методів інтелектуального аналізу даних

Мною було обрано два методи для прогнозування а в подальшому і порівняні їх – це методи лінійна регресія[3] та поліноміальна регресія[4].

Лінійна регресія є одним з найпростіших і найбільш зрозумілих методів аналізу даних. Її основна ідея полягає у встановленні лінійної залежності між вхідними змінними і цільовою змінною. Це дозволяє з легкістю інтерпретувати результати та розуміти вплив кожної змінної на цільову змінну.

Поліноміальна регресія є розширенням лінійної регресії, яке дозволяє моделювати нелінійні залежності. Використовуючи поліноміальні функції, можна захопити складніші взаємозв'язки між змінними. Це забезпечує більшу гнучкість в моделюванні та дозволяє краще враховувати різноманітні закономірності, що можуть існувати в даних.

Обидва методи також мають добру масштабованість і ефективно працюють навіть з великими обсягами даних. Вони широко використовуються в багатьох галузях, включаючи економіку, фінанси, соціальні науки та інші.

Незважаючи на багато переваг, лінійна регресія та поліноміальна регресія також мають деякі обмеження і мінуси, які варто враховувати.

Лінійна регресія передбачає лінійну залежність між вхідними змінними і цільовою змінною. Це може бути недостатньо для точного опису складних нелінійних зв'язків, що можуть існувати в даних. Такі методи можуть бути непридатними для моделювання складних і нелінійних структур даних.

Якщо модель лінійної регресії або поліноміальної регресії занадто точно підлаштується під навчальні дані, вона може перенавчитися і втратити здатність узагальнювати на нові, невидимі дані. Це може призвести до поганої прогнозної здатності моделі на нових наборах даних.

При використанні поліноміальної регресії вибір правильної степені полінома є важливим. Неправильний вибір може призвести до недооцінки або переоцінки впливу змінних, що може призвести до неточних результатів.

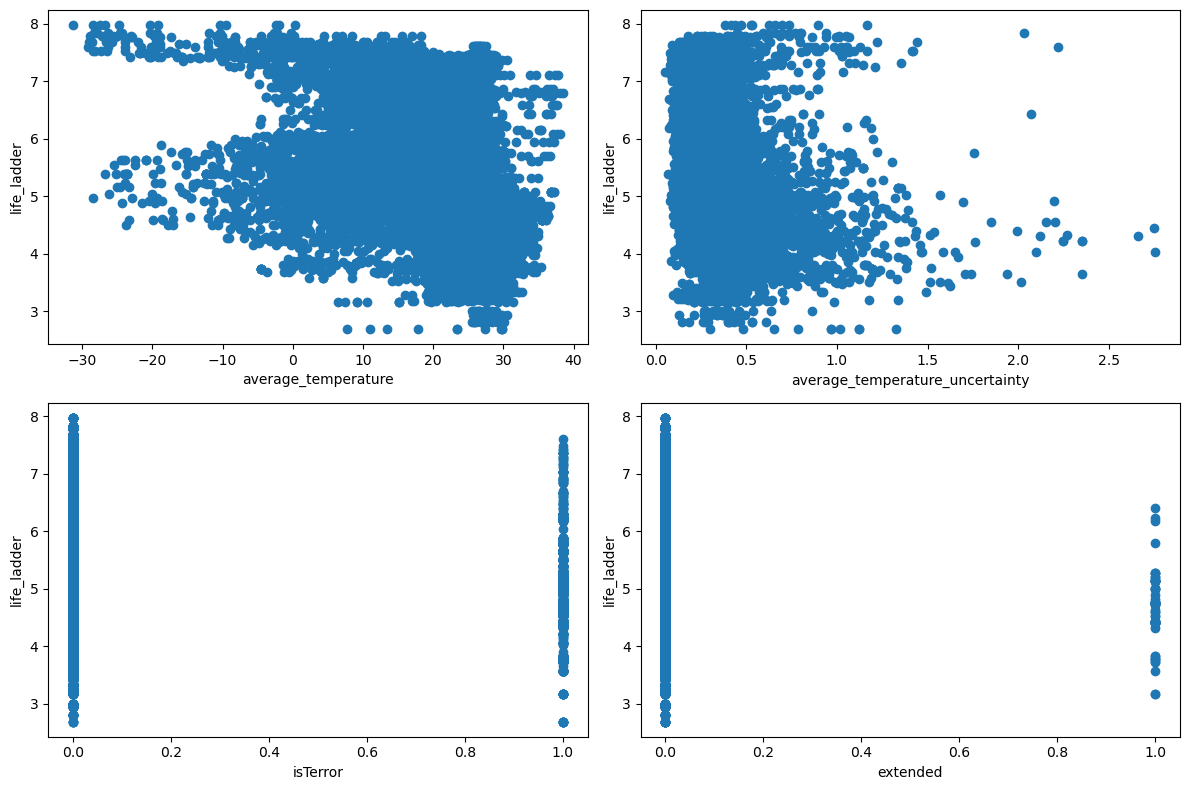
Лінійна регресія та поліноміальна регресія можуть бути чутливими до наявності викидів або аномальних значень в даних. Викиди можуть вплинути на параметри моделі і зробити її менш точною і надійною.

Лінійна регресія та поліноміальна регресія базуються на певних припущеннях про розподіл даних, лінійну залежність і відсутність мультиколінеарності. Якщо ці припущення порушуються, це може призвести до неточних результатів.

Ураховуючи ці мінуси, перед використанням лінійної регресії та поліноміальної регресії важливо ретельно аналізувати дані, перевіряти відповідність моделей припущенням та враховувати їх обмеження.

## Підготовка даних до інтелектуального аналізу

Розглянемо кожний з зібраних факторів для того, щоб визначити їх вплив на результат.(рис. 4.1). Код програми прогнозування знаходиться у додатку А, CWork.py



*Рисунок 4.1 – Графіки залежності результату від кожного предиктора*

Ми одразу бачимо, що сильної залежності нема, проте вже можна виявити закономірності. Наприклад, ми бачимо менші оцінки щастя при дуже великих температурах – засухи, бідні країни Ближнього та Середнього Сходу або Африки. Також видно, що у випадках актів тероризму щастя менше.

Також для використання регресії нам потрібне зробити перевірку на мультиколінеарність (рис. 4.2).



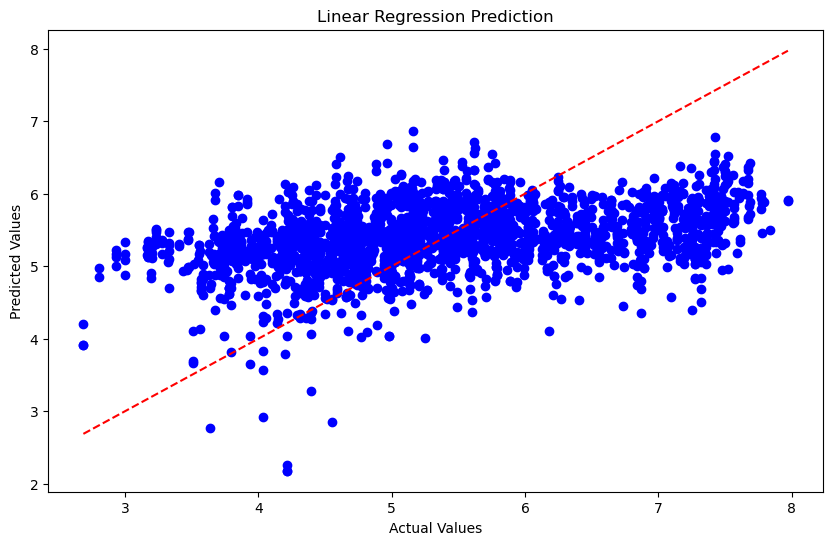
*Рисунок 4.2 – Теплова карта матриці кореляції*

Як ми бачимо з матриці кореляції, мультиколінеарність відсутня. Також ми знову бачимо негативну кореляцію предикторів з результатом, що означає що звʼязок існує і він обернений. Тобто усі вхідні дані можна використовувати для аналізу.

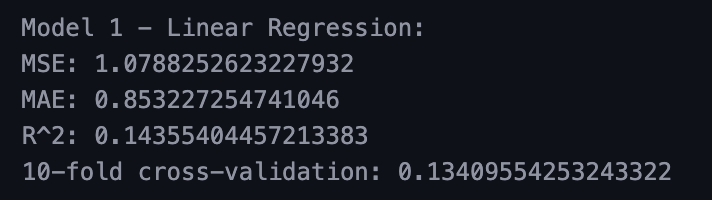
## Тестування методів

Ми тренує моделі на тренувальних значеннях. Далі на тестових вхідних даних ми робимо прогнозовані результати. Потім оцінюємо результати, в даній роботі використовуються метрики: середньоквадратична помилка, середня абсолютна помилка, коефіцієнт детермінації та 10-кратна перехресна перевірка(коефіцієнт детермінації).

Подивимося результати лінійної регресії.(рис. 4.3-4.4)



*Рисунок 4.3 – Відношення правдивого та прогнозованого значень Лінійної регресії*



*Рисунок 4.4 – Метрика першої моделі*

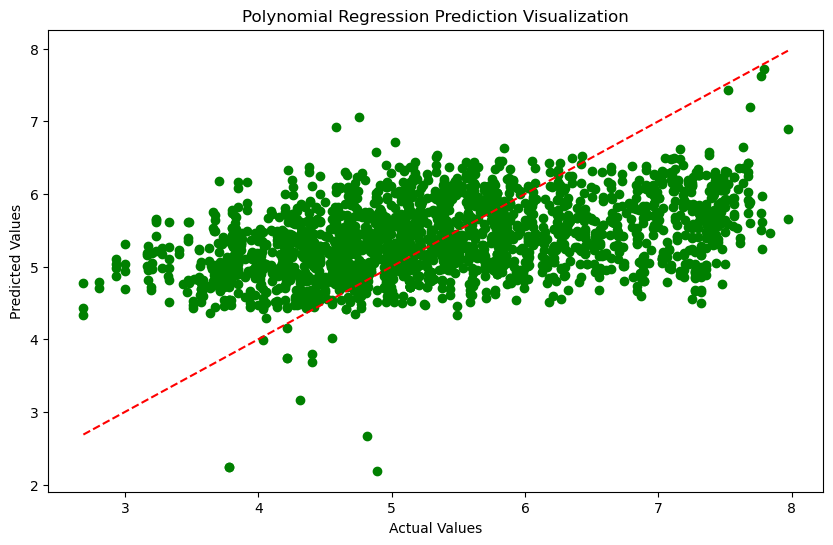
Як ми можемо бачити, то точність невелика. Середньоквадратична помилка більше 1, важливо зазначити, що інтерпретація значень MSE залежить від конкретного контексту та діапазону цільової змінної. У деяких випадках MSE вище 1 може бути прийнятним, якщо цільова змінна має великий діапазон або високу дисперсію. Тому можна припускати, що метод не вловлює закономірність, або розкид значень результату доволі великий, де друге в нашому випадку більш схоже на правду.

Для роботи Поліноміальної регресії, потрібно знайти степінь полінома, який дає кращий результат(рис. 4.5).

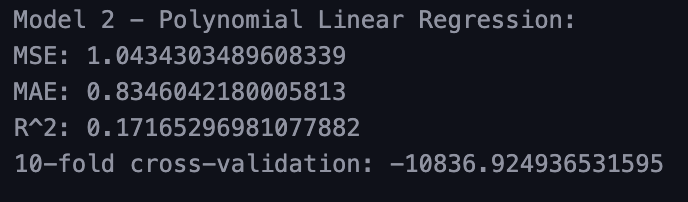


*Рисунок 4.5 – Знаходимо кращий степінь поліному*

Кращій степінь – це 4, тому можемо тренувати нашу модель. Проаналізуємо результати Поліноміальної регресії.(рис. 4.6-4.7)



*Рисунок 4.6 – Відношення правдивого та прогнозованого значень Поліноміальної регресії*

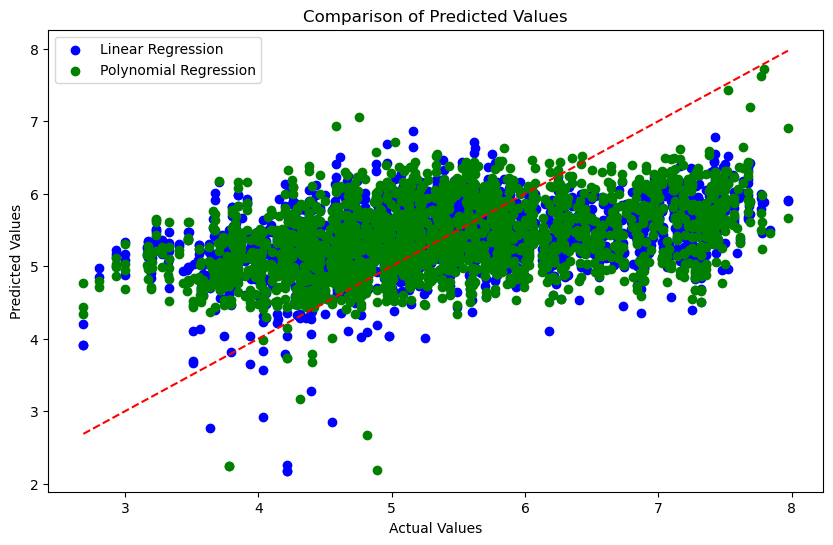


*Рисунок 4.7 – Метрика першої моделі*

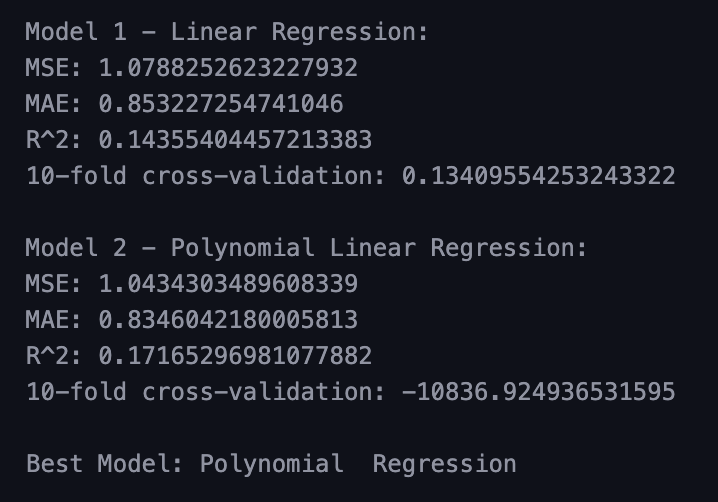
10-кратний показник перехресної перевірки -10836,924936531595 вказує на низьку продуктивність або потенційну проблему з моделлю. Важливо зазначити, що від’ємне значення означає, що модель працює гірше, ніж модель постійної базової лінії. Але аналізуючи іншу метрику можна виявити мінімальні покращення, тому можна перейти до порівняння методів.

## Порівняння методів

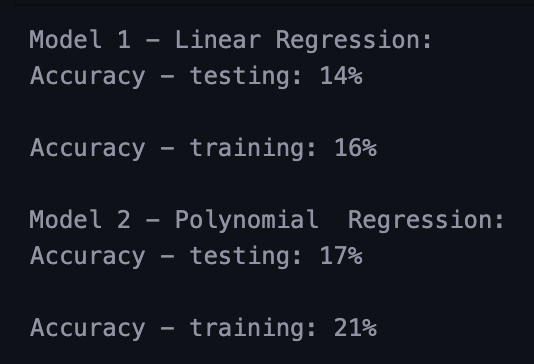
Порівняємо результати методів графічно та за допомогою метрики та виявимо кращий метод.(рис. 4.8-4.10)



*Рисунок 4.8 – Порівняння двох методів*

**

*Рисунок 4.9 – Виявлення кращого методу*

**

*Рисунок 4.10 – Точність методів на тестових та тренувальних даних*

Отже, маємо що максимальної точності прогнозування в даних, на яких тренувалися ці методи, досягла поліноміальна регресія – це 21%, порівнюючи його з лінійною регресією, що досягла 16%. Як можна побачити обидва методи виявили дуже низьку точність та велику похибку, але вже можна виявити закономірності, тому зробимо висновки.

Висновки

В результаті виконання курсової роботи було оброблено та проаналізовано дані про щастя країн, деякі кліматичні показники та показники тероризму, завантажено дані до сховища даних. Для реалізації даних задач було використано MySQL та мову Python3.

З аналізу прогнозу можна зробити висновки, що дана модель не дає точних результатів. За результатами можна виявити, що прямого звʼязку між факторами немає, проте можна припустити існування посередньої залежності. Тобто наші вхідні дані впливають на якийсь фактор, від якого результат вже залежить сильніше. Проте вже можна свідчити, що навіть температура повітря впливає на щастя країни.

Отже, поставлені задачі були виконані, для подальшого використання моделі потребується виявити опосередкований звʼязок, який вже допоможе з точністю прогнозування та зробить прийнятною для використання.

Перелік посилань

1. Документація РСКБД MySQL. [Електронний ресурс] Доступ за посиланням:

<https://docs.oracle.com/cd/E17952_01/index.html>

1. Документація мови програмування Python. [Електронний ресурс] Доступ за посиланням:

<https://docs.python.org/3/>

1. Лінійна регресія [Електронний ресурс] Доступ за посиланням:

<https://en.wikipedia.org/wiki/Linear_regression>

1. Поліноміальна регресія [Електронний ресурс] Доступ за посиланням:

<https://en.wikipedia.org/wiki/Polynomial_regression>

1. Бібліотека Pandas. [Електронний ресурс] Доступ за посиланням:

<https://pandas.pydata.org/docs/>

1. Бібліотека Matplotlib. [Електронний ресурс] Доступ за посиланням:

<https://matplotlib.org/stable/>

1. Бібліотека Sklearn. [Електронний ресурс] Доступ за посиланням:

<https://scikit-learn.org/stable/user_guide.html>

1. Дані для аналізу Kaggle [Електронний ресурс] Доступ за посиланням:

<https://www.kaggle.com/>

Додаток А Тексти програмного коду

*Тексти програмного коду прогнозування щастя країни за показниками терористичних актів та клімату*

(Найменування програми)

*Жорсткий диск*

(Вид носія даних)

(Обсяг програми)

Студента групи ІП-15 2 курсу

Мєшкова А.І.

StageZone.sql

-- -----------------------------------------------------

-- Table Hapiness

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS Hapiness (

`hapiness\_id` INT NOT NULL AUTO\_INCREMENT,

`country\_name` VARCHAR(45) NULL,

`year` INT NULL,

`life\_ladder` DECIMAL(10,3) NULL,

`gdp\_per\_capita` DECIMAL(10,3) NULL,

`positive\_affect` DECIMAL(10,3) NULL,

`social\_support` DECIMAL(10,3) NULL,

`life\_expancy` DECIMAL(10,3) NULL,

`freedom\_choice` DECIMAL(10,3) NULL,

`generocity` DECIMAL(10,3) NULL,

`corruption` DECIMAL(10,3) NULL,

`negative affect` DECIMAL(10,3) NULL,

PRIMARY KEY (`hapiness\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table Climat

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS Climat (

`climat\_id` INT NOT NULL AUTO\_INCREMENT,

`date` DATE NULL,

`average\_temperature` DECIMAL(10,3) NULL,

`average\_temperature\_uncertainty` DECIMAL(10,3) NULL,

`country\_name` VARCHAR(45) NULL,

PRIMARY KEY (`climat\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table Terrorism

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS Terrorism (

`event\_id` BIGINT NOT NULL,

`year` INT NULL,

`month` INT NULL,

`day` INT NULL,

`extended` TINYINT(2) NULL,

`country\_id` INT NULL,

`country\_name` VARCHAR(45) NULL,

PRIMARY KEY (`event\_id`))

ENGINE = InnoDB;

DataWarehouse.sql

-- -----------------------------------------------------

-- Table dim\_climat

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS dim\_climat (

`climat\_id` INT NOT NULL AUTO\_INCREMENT,

`average\_temperature` DECIMAL(10,3) NULL,

`average\_temperature\_uncertainty` DECIMAL(10,3) NULL,

PRIMARY KEY (`climat\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table dim\_terrorism

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS dim\_terrorism (

`event\_id` INT NOT NULL AUTO\_INCREMENT,

`event\_name` VARCHAR(45) NULL,

`extended` TINYINT(2) NULL,

PRIMARY KEY (`event\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table dim\_date

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS dim\_date (

`date\_id` INT NOT NULL AUTO\_INCREMENT,

`year` INT NULL,

`month` INT NULL,

`day` INT NULL,

PRIMARY KEY (`date\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table dim\_country

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS dim\_country (

`country\_id` INT NOT NULL AUTO\_INCREMENT,

`country\_code` INT NULL,

`country\_name` VARCHAR(45) NULL,

PRIMARY KEY (`country\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table fact\_hapiness\_analysis

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS fact\_hapiness\_analysis (

`fact\_hapiness\_analysis\_id` INT NOT NULL AUTO\_INCREMENT,

`climat\_id` INT NULL,

`date\_id` INT NOT NULL,

`country\_id` INT NOT NULL,

`event\_id` INT NULL,

`life\_ladder` DECIMAL(10,3) NULL,

`gdp\_per\_capita` DECIMAL(10,3) NULL,

`positive\_affect` DECIMAL(10,3) NULL,

`social\_support` DECIMAL(10,3) NULL,

`life\_expancy` DECIMAL(10,3) NULL,

`freedom\_choice` DECIMAL(10,3) NULL,

`generocity` DECIMAL(10,3) NULL,

`corruption` DECIMAL(10,3) NULL,

`negative\_affect` DECIMAL(10,3) NULL,

PRIMARY KEY (`fact\_hapiness\_analysis\_id`),

INDEX `fk1\_idx` (`climat\_id` ASC) VISIBLE,

INDEX `fk2\_idx` (`date\_id` ASC) VISIBLE,

INDEX `fk3\_idx` (`country\_id` ASC) VISIBLE,

INDEX `fk4\_idx` (`event\_id` ASC) VISIBLE,

CONSTRAINT `fk1`

FOREIGN KEY (`climat\_id`)

REFERENCES dim\_climat (`climat\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk2`

FOREIGN KEY (`date\_id`)

REFERENCES dim\_date (`date\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk3`

FOREIGN KEY (`country\_id`)

REFERENCES dim\_country (`country\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk4`

FOREIGN KEY (`event\_id`)

REFERENCES dim\_terrorism (`event\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

ETL.sql

-- ----------------------------------------------------

-- Table dim\_climat

-- -----------------------------------------------------

INSERT INTO DataWarehouse.dim\_climat (average\_temperature, average\_temperature\_uncertainty)

SELECT

ROUND(average\_temperature, 3) as average\_temperature,

ROUND(average\_temperature\_uncertainty, 3) as average\_temperature\_uncertainty

FROM StageZone.Climat;

-- ----------------------------------------------------

-- Table dim\_terrorism

-- -----------------------------------------------------

INSERT INTO DataWarehouse.dim\_terrorism (event\_name, extended)

SELECT event\_id, extended

FROM StageZone.Terrorism;

-- ----------------------------------------------------

-- Table dim\_country

-- -----------------------------------------------------

INSERT INTO DataWarehouse.dim\_country (country\_code, country\_name)

SELECT country\_id, country\_name FROM StageZone.Terrorism

UNION

SELECT NULL, country\_name FROM StageZone.Hapiness

UNION

SELECT NULL, country\_name FROM StageZone.Climat

WHERE NOT EXISTS (

SELECT \* FROM DataWarehouse.dim\_country

WHERE DataWarehouse.dim\_country.country\_name = country\_name

);

-- ----------------------------------------------------

-- Table dim\_date

-- -----------------------------------------------------

INSERT IGNORE INTO DataWarehouse.dim\_date (year, month, day)

SELECT DISTINCT YEAR(date) AS year, MONTH(date) AS month, DAY(date) AS day

FROM StageZone.Climat

UNION

SELECT DISTINCT year, NULL, NULL FROM StageZone.Hapiness

UNION

SELECT DISTINCT year, month, day FROM StageZone.Terrorism

WHERE NOT EXISTS (

SELECT \* FROM DataWarehouse.dim\_date

WHERE (DataWarehouse.dim\_date.year = year AND DataWarehouse.dim\_date.month = month AND DataWarehouse.dim\_date.day = day)

);

-- ----------------------------------------------------

-- Table fact\_hapiness\_analysis

-- -----------------------------------------------------

INSERT INTO fact\_hapiness\_analysis

(climat\_id, date\_id, country\_id, event\_id, life\_ladder, gdp\_per\_capita, positive\_affect, social\_support, life\_expancy, freedom\_choice, generocity, corruption, negative\_affect)

select

CL.climat\_id,

D.date\_id,

CO.country\_id,

T.event\_id,

SH.life\_ladder,

SH.gdp\_per\_capita,

SH.positive\_affect,

SH.social\_support,

SH.life\_expancy,

SH.freedom\_choice,

SH.generocity,

SH.corruption,

SH.negative\_affect

from (

select \*

from StageZone.Climat

where YEAR(date) > 2004

) SCL

inner join dim\_climat CL ON CL.average\_temperature = SCL.average\_temperature and CL.average\_temperature\_uncertainty = SCL.average\_temperature\_uncertainty

left join StageZone.Terrorism ST on ST.country\_name = SCL.country\_name and ST.year = YEAR(SCL.date) and

ST.month = MONTH(SCL.date) and ST.day = DAY(SCL.date)

left join dim\_terrorism T on ST.extended = T.extended and ST.event\_name = T.event\_name

left join StageZone.Hapiness SH on YEAR(SCL.date) = SH.year and SH.country\_name = SCL.country\_name

left join dim\_date D on YEAR(SCL.date) = D.year and MONTH(SCL.date) = D.month and

DAY(SCL.date) = D.day

left join (select \* from dim\_country where country\_code is null) CO on CO.country\_name = SCL.country\_name;

CWork.py

import pymysql

#Connection with Database

connection = pymysql.connect(

host="127.0.0.1",

user="root",

password="",

database="DataWarehouse"

)

cursor = connection.cursor()

 #Fetching data

cursor = connection.cursor()

query ="""

SELECT f.life\_ladder, f.gdp\_per\_capita, f.social\_support, f.life\_expancy, f.freedom\_choice,

c.average\_temperature, c.average\_temperature\_uncertainty, t.event\_name, t.extended

FROM fact\_hapiness\_analysis AS f

JOIN dim\_climat AS c ON f.climat\_id = c.climat\_id

LEFT JOIN dim\_terrorism AS t ON f.event\_id = t.event\_id

JOIN dim\_country AS ct ON f.country\_id = ct.country\_id

"""

cursor.execute(query)

data = cursor.fetchall()

import pandas as pd

#Preparing data

columns = ['life\_ladder', 'gdp\_per\_capita', 'social\_support', 'life\_expancy', 'freedom\_choice',

'average\_temperature', 'average\_temperature\_uncertainty',

'event\_name', 'extended']

df = pd.DataFrame(data, columns=columns)

#Adding isTerror instead of event\_name

df['isTerror'] = df['event\_name'].apply(lambda x: 1 if pd.notnull(x) else 0)

df = df.drop(columns=['event\_name'])

#Analyze structure

df.info()

import numpy as np

#Fixing data

df['extended'] = df['extended'].fillna(0)

noNanData = list(df.columns)[0:5]

for i in noNanData:

df = df.dropna()

numeric\_columns = ['life\_ladder', 'gdp\_per\_capita', 'social\_support', 'life\_expancy',

'freedom\_choice', 'average\_temperature', 'average\_temperature\_uncertainty']

df[numeric\_columns] = df[numeric\_columns].apply(pd.to\_numeric, errors='coerce')

#Analyze structure after fixing

df.info()

df.head(10)

import matplotlib.pyplot as plt

#Visual of relations

fig, axs = plt.subplots(2, 2, figsize=(12, 8))

axs[0, 0].plot(df['average\_temperature'], df['life\_ladder'], 'o')

axs[0, 0].set\_xlabel('average\_temperature')

axs[0, 0].set\_ylabel('life\_ladder')

axs[0, 1].plot(df['average\_temperature\_uncertainty'], df['life\_ladder'], 'o')

axs[0, 1].set\_xlabel('average\_temperature\_uncertainty')

axs[0, 1].set\_ylabel('life\_ladder')

axs[1, 0].plot(df['isTerror'], df['life\_ladder'], 'o')

axs[1, 0].set\_xlabel('isTerror')

axs[1, 0].set\_ylabel('life\_ladder')

axs[1, 1].plot(df['extended'], df['life\_ladder'], 'o')

axs[1, 1].set\_xlabel('extended')

axs[1, 1].set\_ylabel('life\_ladder')

plt.tight\_layout()

plt.show()

#Dividing data

from sklearn.model\_selection import train\_test\_split

X = df[['average\_temperature', 'average\_temperature\_uncertainty', 'isTerror', 'extended']]

y = df['life\_ladder']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

import seaborn as sns

#Calculate the correlation matrix

data = pd.concat([y, X], axis=1)

correlation\_matrix = data.corr()

#Create a heatmap

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Matrix')

plt.show()

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import PolynomialFeatures

from sklearn.metrics import mean\_squared\_error, r2\_score, mean\_absolute\_error

from sklearn.model\_selection import cross\_val\_score

#Training models

model1 = LinearRegression()

model1.fit(X\_train, y\_train)

y\_pred1 = model1.predict(X\_test)

mae1 = mean\_absolute\_error(y\_test, y\_pred1)

mse1 = mean\_squared\_error(y\_test, y\_pred1)

r2\_1 = r2\_score(y\_test, y\_pred1)

scores1 = cross\_val\_score(model1, X, y, cv=10, scoring='r2')

r2\_1\_cross\_val = scores1.mean()

poly = PolynomialFeatures(degree=4)

X\_poly = poly.fit\_transform(X)

X\_train\_poly = poly.fit\_transform(X\_train)

X\_test\_poly = poly.transform(X\_test)

model2 = LinearRegression()

model2.fit(X\_train\_poly, y\_train)

y\_pred2 = model2.predict(X\_test\_poly)

mae2 = mean\_absolute\_error(y\_test, y\_pred2)

mse2 = mean\_squared\_error(y\_test, y\_pred2)

r2\_2 = r2\_score(y\_test, y\_pred2)

scores2 = cross\_val\_score(model2, X\_poly, y, cv=10, scoring='r2')

r2\_2\_cross\_val = scores2.mean()

# Compare and select the best model

if r2\_1 > r2\_2 :

best\_model = model1

best\_model\_name = 'Linear Regression'

else:

best\_model = model2

best\_model\_name = 'Polynomial Regression'

#Finding degree for PR

def find\_best\_degree(X\_train, y\_train, X\_test, y\_test, degrees):

best\_degree = 1

best\_score = -np.inf

for degree in range(1, degrees+1):

poly = PolynomialFeatures(degree=degree)

X\_train\_poly = poly.fit\_transform(X\_train)

X\_test\_poly = poly.transform(X\_test)

model = LinearRegression()

model.fit(X\_train\_poly, y\_train)

score = model.score(X\_test\_poly, y\_test)

if score > best\_score:

best\_score = score

best\_degree = degree

return best\_degree, best\_score

best\_degree, best\_score = find\_best\_degree(X\_train, y\_train, X\_test, y\_test, degrees=10)

print("Best degree:", best\_degree)

print("Best score:", best\_score)

# Visualize the Linear Regression predictions

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred1, color='blue')

plt.plot([min(y\_test), max(y\_test)], [min(y\_test), max(y\_test)], color='red', linestyle='--')

plt.xlabel('Actual Values')

plt.ylabel('Predicted Values')

plt.title('Linear Regression Prediction')

plt.show()

# Visualize the Polynomial Regression predictions

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred2, color='green')

plt.plot([min(y\_test), max(y\_test)], [min(y\_test), max(y\_test)], color='red', linestyle='--')

plt.xlabel('Actual Values')

plt.ylabel('Predicted Values')

plt.title('Polynomial Regression Prediction Visualization')

plt.show()

# Plotting the predictions and actual both methods values

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred1, color='blue', label='Linear Regression')

plt.scatter(y\_test, y\_pred2, color='green', label='Polynomial Regression')

plt.plot([min(y\_test), max(y\_test)], [min(y\_test), max(y\_test)], color='red', linestyle='--')

plt.xlabel('Actual Values')

plt.ylabel('Predicted Values')

plt.title('Comparison of Predicted Values')

plt.legend()

plt.show()

#Metrics

print(f"Model 1 - Linear Regression:")

print(f"MSE: {mse1}")

print(f"MAE: {mae1}")

print(f"R^2: {r2\_1}")

print(f"10-fold cross-validation: {r2\_1\_cross\_val}")

print()

print(f"Model 2 - Polynomial Linear Regression:")

print(f"MSE: {mse2}")

print(f"MAE: {mae2}")

print(f"R^2: {r2\_2}")

print(f"10-fold cross-validation: {r2\_2\_cross\_val}")

print()

# Print the best model

print(f"Best Model: {best\_model\_name}")

# Print the accuracy for each model

print(f"Model 1 - Linear Regression:")

print(f"Accuracy - testing: {int(model1.score(X\_test, y\_test)\*100)}%")

print()

print(f"Accuracy - training: {int(model1.score(X\_train, y\_train)\*100)}%")

print()

print(f"Model 2 - Polynomial Regression:")

print(f"Accuracy - testing: {int(model2.score(X\_test\_poly, y\_test)\*100)}%")

print()

print(f"Accuracy - training: {int(model2.score(X\_train\_poly, y\_train)\*100)}%")

print()

# Close the cursor and connection

cursor.close()

connection.close()