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.»

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: 27 сторінок, 26 рисунки, 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

[АНОТАЦІЯ 3](#_Toc14477)

[ВСТУП 5](#_Toc6229)

[1. ПОСТАНОВКА ЗАДАЧІ 6](#_Toc18657)

[2. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 7](#_Toc3711)

[3. РОЗРОБКА СХОВИЩА ДАНИХ 9](#_Toc8979)

[3.1. Розробка ETL процесів 9](#_Toc13183)

[3.1.1. ETL для датасету World happiness report 2015-2022 9](#_Toc23665)

[3.1.2. ETL для датасету Deaths Reasons. 10](#_Toc17647)

[3.1.3. ETL для Country Consumption 11](#_Toc19367)

[3.1.4. Скрипт mainETL.py 12](#_Toc30809)

[3.2. Створення сховища даних 13](#_Toc27489)

[3.3. Завантаження даних та імпортування важливих бібліотек 13](#_Toc13279)

[3.3 Feature Engineering 15](#_Toc3798)

[3.4 Очищення даних 17](#_Toc21770)

[3.5 Візуалізація даних 18](#_Toc24533)

[Побудуємо гістограму, яка показує кількість статей відповідно до кількості слів в ній. 19](#_Toc16179)

[3.6 Tokenization and padding 20](#_Toc6664)

[1. ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ 23](#_Toc30250)

[4.1 Обґрунтування вибору алгоритмів для побудови моделі 23](#_Toc21900)

[4.2 Побудова моделі 23](#_Toc28414)

[4.3 Тренування моделі 24](#_Toc785)

[4.4 LSTM performance 25](#_Toc20498)

[4.5 Logistic Regression performance 26](#_Toc3604)

[4.6 Порівняння ефективності методів LSTM та Logistic Regression 26](#_Toc25323)

[ПЕРЕЛІК ПОСИЛАНЬ 28](#_Toc32258)

[ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ 29](#_Toc19799)

## 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.

## 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. 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.

## 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.

1. Country Electricity Consumption

Описує параметр споживання електроенергії певної країни певного року .

1. Deaths Reasons

Описує причини смертності населення певних країн по роках.

1. Population By Country

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

## РОЗРОБКА СХОВИЩА ДАНИХ

### 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.

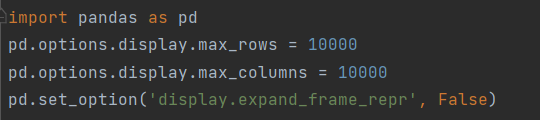


Рисунок 3.1 - Завантаження та імпорт бібліотек

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

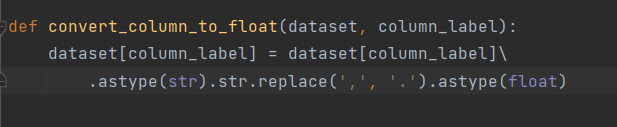


Рисунок 3.2 - Функція-конвертер

#### 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.

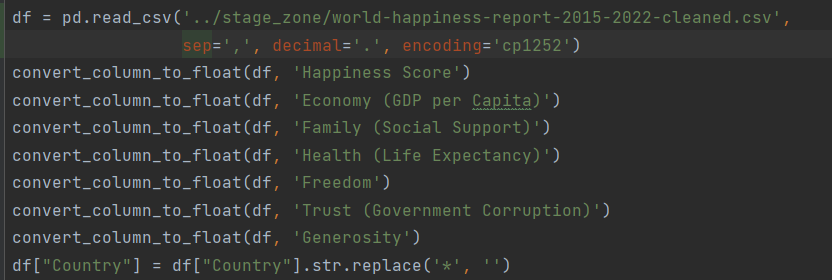


Рисунок 3.3 - Перетворення датасету World happiness report 2015-2022

І насамкінець видалимо непотрібну колонку та збережемо виправлений датасет.

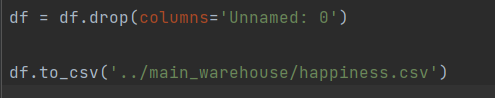


Рисунок 3.4 - Збереження датасету World happiness report 2015-2022

#### 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.

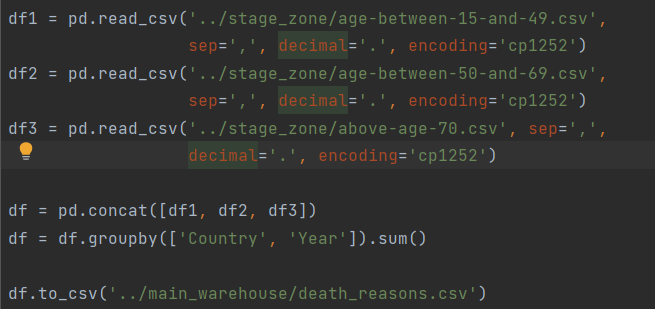


Рисунок 3.5 - Перетворення Deaths Reasons

#### 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.

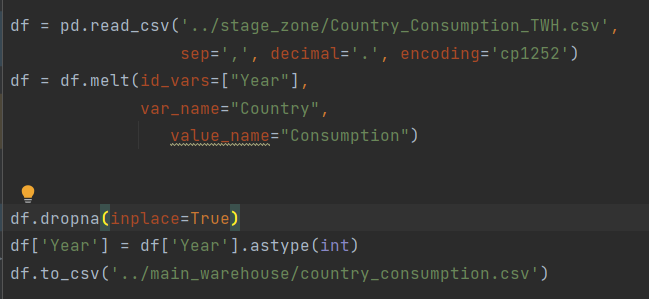


Рисунок 3.5 - Перетворення Country Consumption

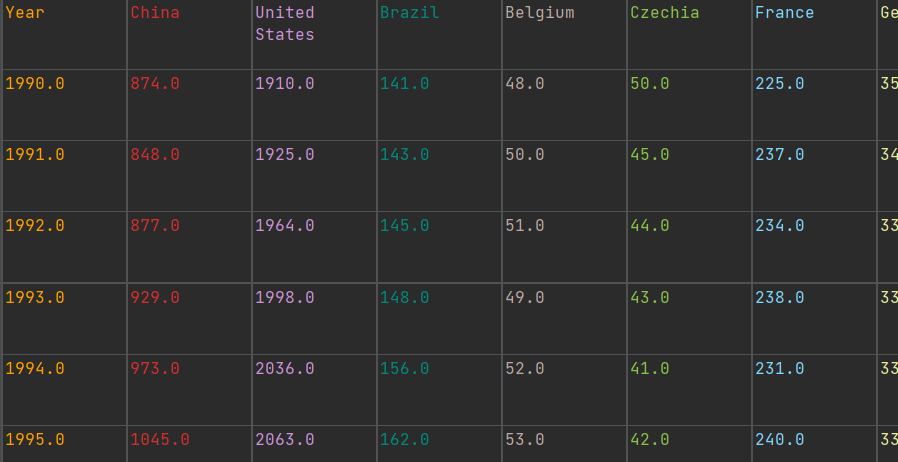


Рисунок 3.6 - До перетворення

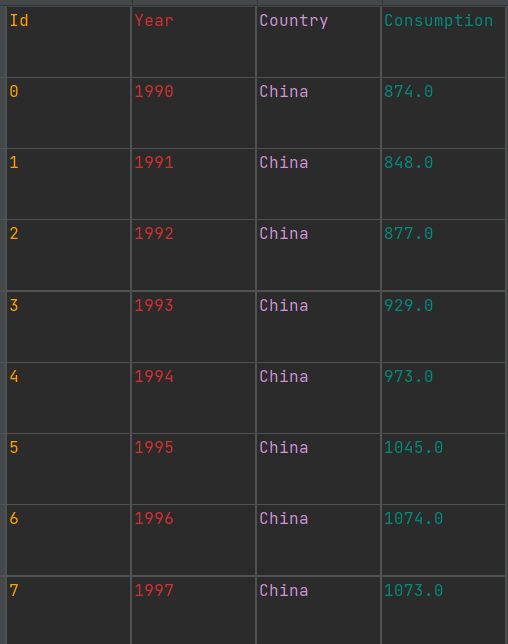


Рисунок 3.7 - Після перетворення

#### Скрипт mainETL.py

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

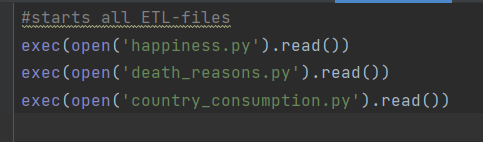
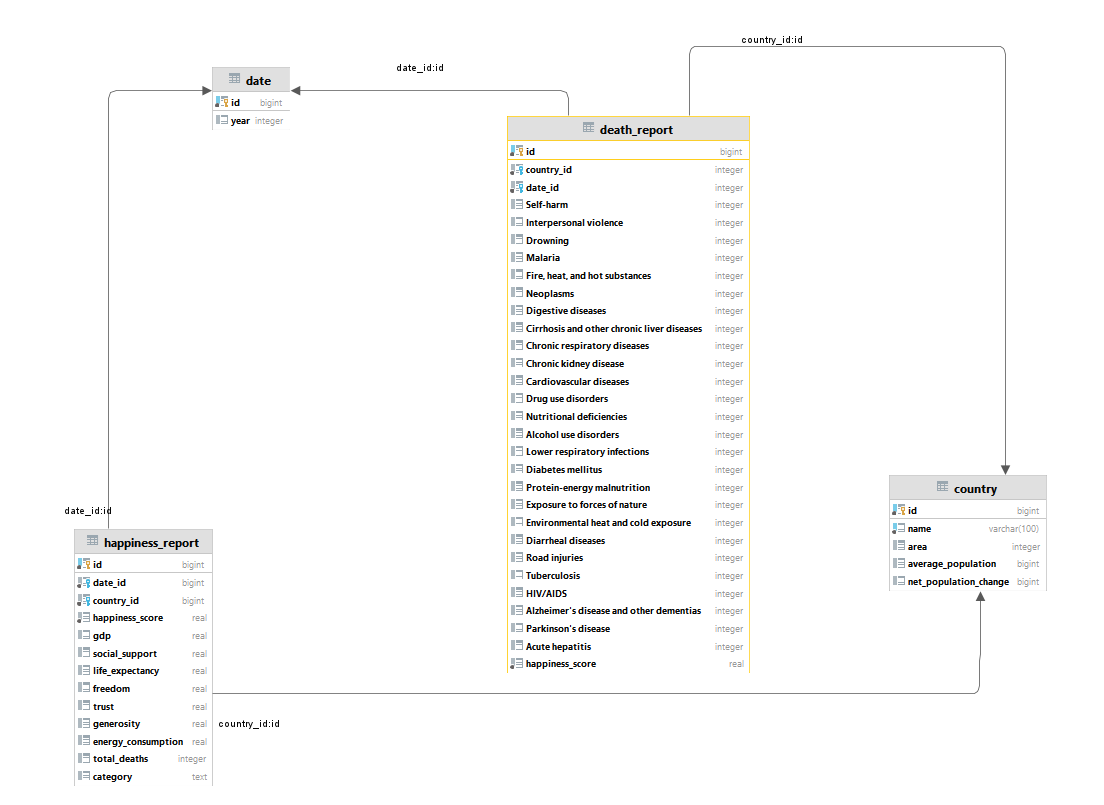


Рисунок 3.8 - Головний ETL скрипт

### Створення сховища даних



### Завантаження даних та імпортування важливих бібліотек

Перш за все, імпортуємо усі бібліотеки, що потрібні нам для аналізу даних. Далі вкажемо деякі конфігурації консольного виводу.

2ETL python

1Create warehouse and Load data

\*Views

\*Data for analysis preparation(multicollinearity, Illnesses correlations)

1Regression

2Classification

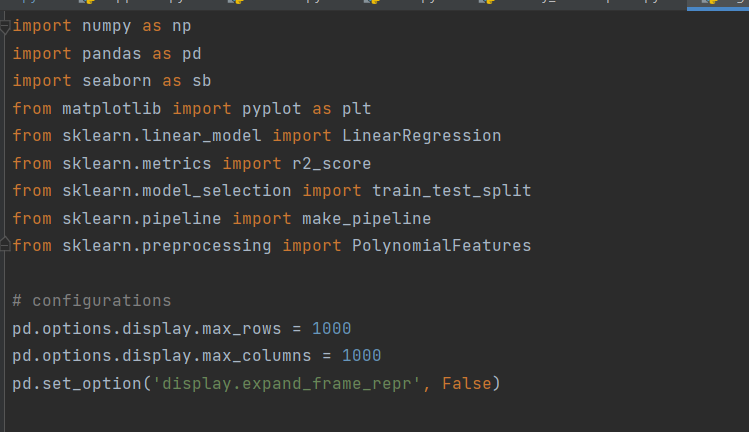


Рисунок 3.1 - Завантаження та імпорт бібліотек

Використовуючи pandas завантажимо 2 .csv файли: один з яких складається з фейкових новин, а інший з - правдивих.

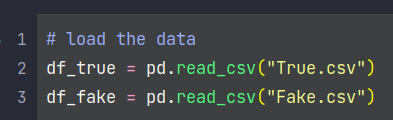


Рисунок 3.2 - Завантаження .csv файлів в датафрейми

Загальна інформація про обидва фрейми:

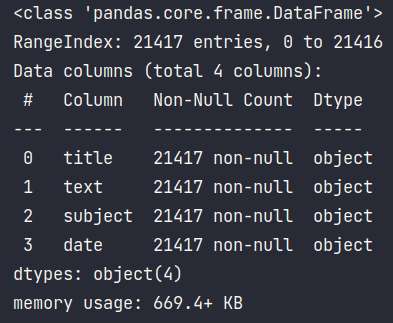


Рисунок 3.3 - Інформація про True dataset

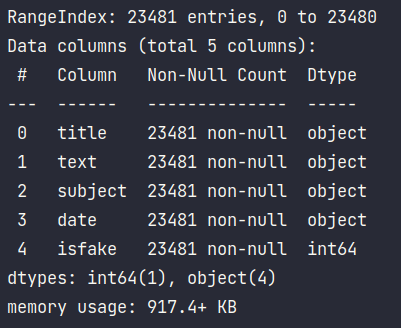


Рисунок 3.4 - Інформація про Fake dataset

### 3.3 Feature Engineering

Зробимо перевірку на існування невалідних значень і у разі виявлень такий рядків, видалимо їх.

Підготуємо два датасети до зливання, додаючи колонку isfake до обох з відповідним значення (1 - true, 0 - fake).

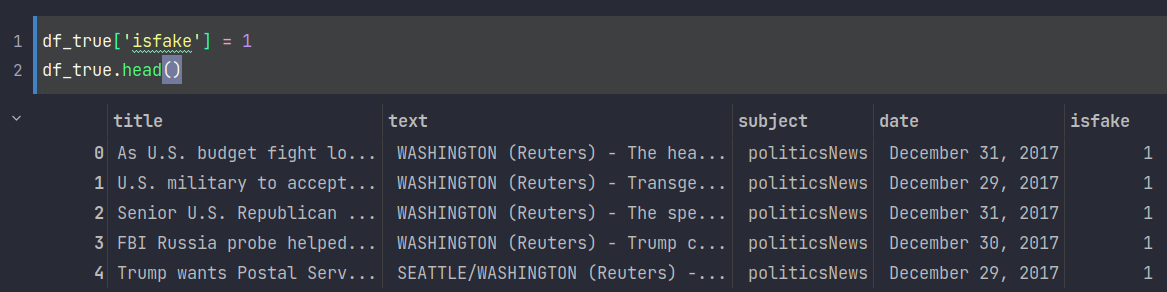


Рисунок 3.5 - Додавання допоміжної колонки ‘isfake’

Об’єднаємо два датасети в один загальний, при цьому дотримуючись правильної індексації.

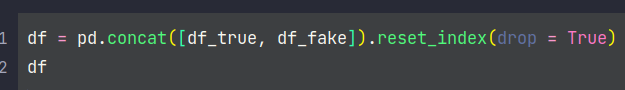


Рисунок 3.6 - Злиття Fake та True датасетів

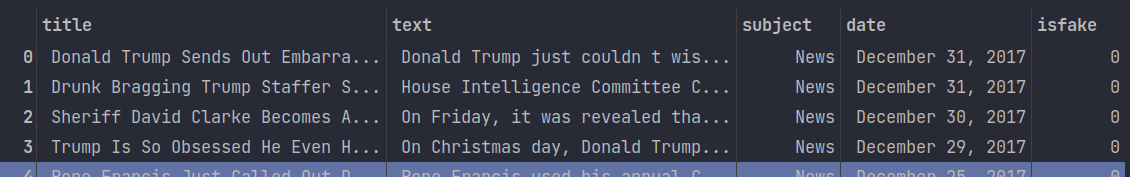


Рисунок 3.7 - Приклад True dataset

Проаналізуємо утворену таблицю. Можна помітити, що колонка ‘date’ є непотрібною, тому можемо видалити її.

Також об’єднаємо колонки text та title до original - саме цю колонку ми будемо використовувати для Tokenization і тренування нашої моделі.

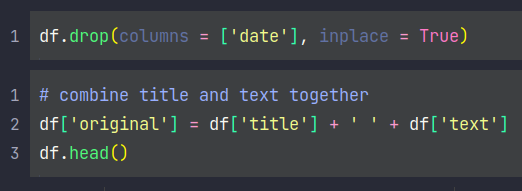


Рисунок 3.8 - Видалення ‘date’ колонки та утворення ‘original’

### 3.4 Очищення даних

У попередньому кроці ми створили колонку original, яку і плануємо використовувати для тренування моделі.

Для очищення даних на потрібно:

1. Видалити всі stop words.
2. Видалити всі слова, які мають довжину меншу за 4.
3. Видалити всі непотрібні спеціальні символи.

Для отримання списку stop words ми використаємо open source Python library Natural Language Toolkit (ntlk). Після аналізу декількох статей ми вирішили, що було б добре розширити наш список існуючих стоп слів, додавши from, subject, re, edu, use.

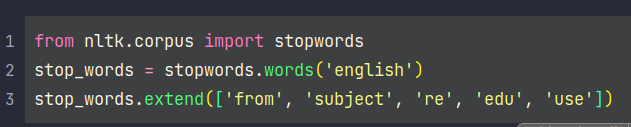


Рисунок 3.9 - Завантаження та розширення стоп-слів

Далі ми створимо функцію, яка на вході отримує рядок (статтю), а на виході ми отримуватимемо масив всіх слів, які задовольняють наші вимоги. Для того, щоб отримати список всіх слів з рядка, враховуючи всі спецсимволи, ми використаємо ***gensim.utils.simple\_preprocess()***. Дану функцію застосовуємо до кожного значення в колонці original.

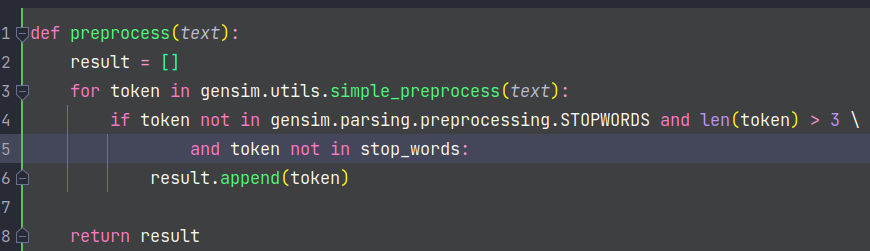


Рисунок 3.10 - Функція для очищення тексту

Тобто в результаті ми матимемо список слів для кожної статі як показано на малюнку нижче, який ми помістимо в нову колонку. Після цього також було б доречно створити ще одну колонку в якому міститимуться всі слова, об’єднані в один рядок.



Рисунок 3.11 - Очищена стаття у вигляді масиву

### 3.5 Візуалізація даних

Для кращого розуміння, давайте зробимо деяку візуалізацію нашого датасету. Ми знаємо, що наш датасет містить колонку з subject, тож побудуємо гістограму, яка показує скільки статей відносяться до певного subject.

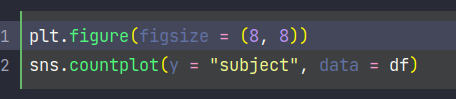


Рисунок 3.12 - Побудова гістограми для subject

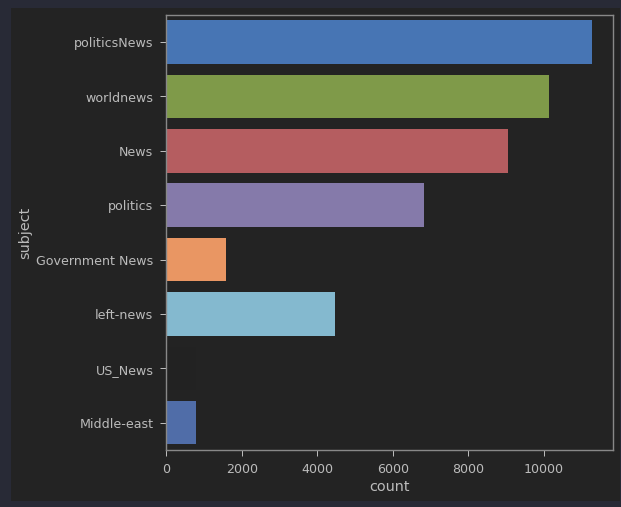


Рисунок 3.13 - Відношення к-сті статей для кожного subject

### Побудуємо гістограму, яка показує кількість статей відповідно до кількості слів в ній.

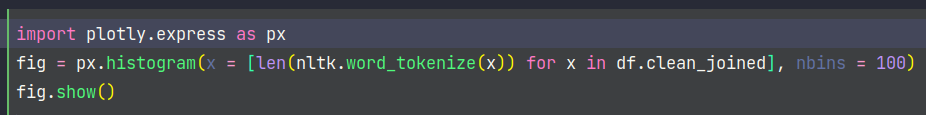


Рисунок 3.13 - Побудова гістограми відношення кількості слів до статей з цією кількістю

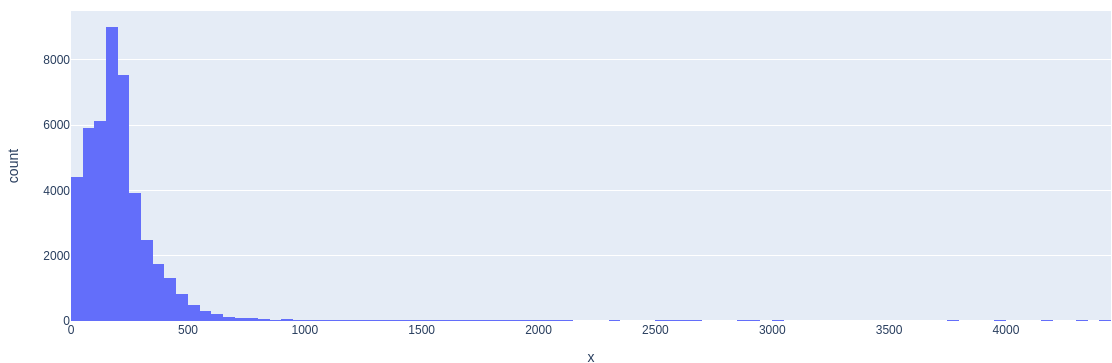


Рисунок 3.14 - Результат в-ння кількості слів до статей з цією кількістю

### 3.6 Tokenization and padding

Перш за все розподілимо наш датесет в train and test.

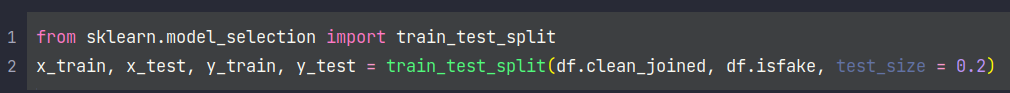


Рисунок 3.15 - Поділ датасету на train та test

Підготуємо дані для tokenization. Для цього знайдемо всі унікальні слова для наших статей.

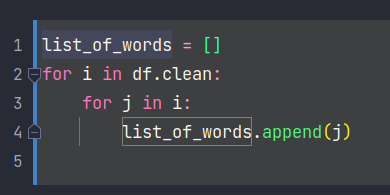


Рисунок 3.16 - Додаємо в масив всі слова

В результаті ми матимемо 9276947 слів і після додавання їх до set, ми отримаємо результат, що ми маємо 108704 унікальних слів.

Знайдемо максимальну кількість слів в одній статті серед усіх. Для цього скористаємося також nltk.

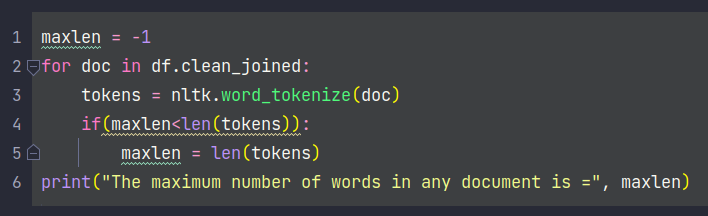


Рисунок 3.17 - Знаходимо розмір максимальної кількості слів в статті

В результаті отримаємо відповідь у 4405 слів.

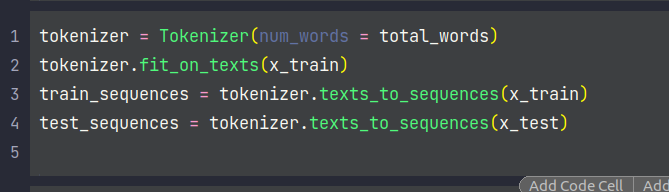
Сам процей Tokenization - це процес, який допомагає нам розбити рядок на менші частинки. У нашому ж випадку, цей процес tokenization допомагає нам перетворити текстове представлення статті в масив чисел. Розбиття рядків на індивідуальні токени допомагає нам простіше класифікувати і перекладати різні тексти. 

Рисунок 3.18 - Tokenization

Процес padding допомагає нам сформувати tokenized масиви однакового розміру, додаючи до кожного з них ту кількість нулів, якої бракує для формування масиву з максимальним розміром.

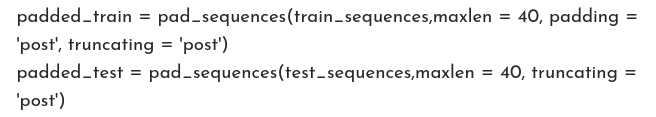


Рисунок 3.19 - Padding

В результаті ми отримаємо наступний результат для кожної статті:

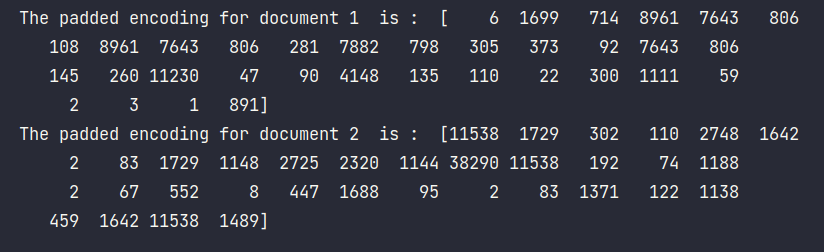


Рисунок 3.20 - Результат tokenization and padding

## 

## ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

### 4.1 Обґрунтування вибору алгоритмів для побудови моделі

Recurrent Neural Network - клас штучних нейронних мереж, у яких зв’язки між вершинами формують направлений або ненаправлений граф у тимчасовій послідовності. Це дає можливість проявляти тимчасову динамічну поведінку. Так як RNN походить від feedforward neural networks, Recurrent Neural Network може використовувати свою пам’ять для обробки змінних довжини введеної інформації.

Long Short-Term Memory - це тип Recurrent Neural Network, який працює на практиці краще за RNN через більш складну і оптимізовану операцію оновлення та через backpropagation dynamics.

Ми використовуємо LSTM замість RNN, тому що LSTM вирішує gradient descent problem. Тобто не призведе до того, що нейронна мережа навчатиметься дуже повільно, а в деяких випадках взагалі не навчатиметься.

### 4.2 Побудова моделі

Для побудови моделі ми використаємо Sequential model, яка підходить для простого стеку шарів, де кожен шар має рівно один input tensor та один output tensor.

Embedding layer перетворює цілі додатні числа (індекси) у dense vectors фіксованого розміру. Ми використовуємо для побудови алгоритму optimazer “Adam”. Це stochastic gradient descent, що грунтується на адаптивній оцінці першого та другого порядку.

Також ми додамо дві activation functions:

* relu activation function - повертає максимальне значення входу та нуля.
* sigmoid activation function - повертає значення близьке до нуля.

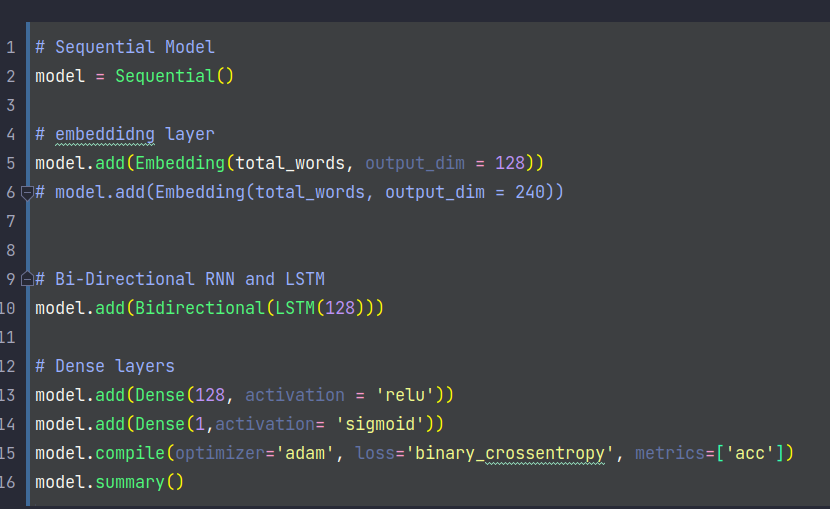
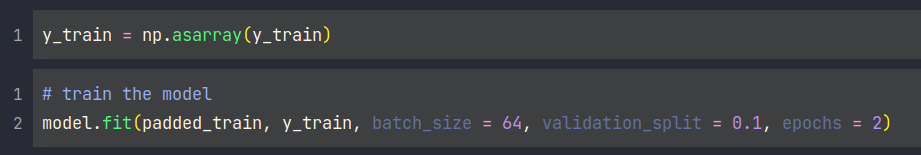
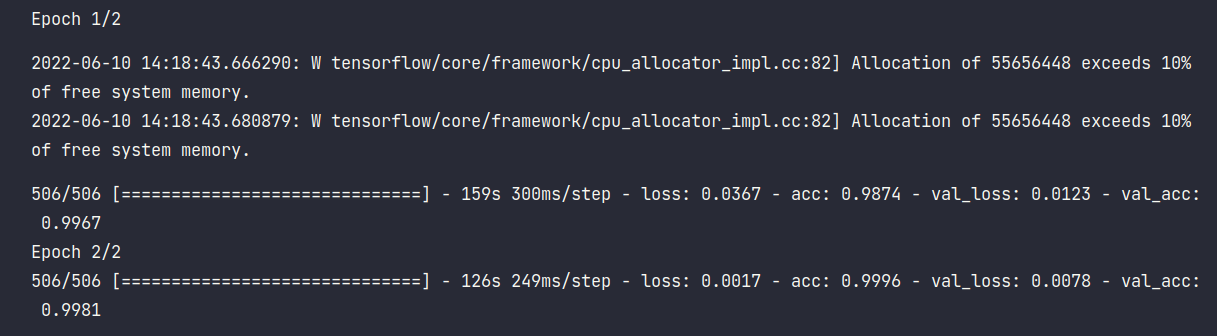


Рисунок 4.1 - LSTM model

### 4.3 Тренування моделі

Наступним кроком ми натренуємо нашу модель використовуючи train набір даних.

Рисунок 4.2 - Train LSTM model

Рисунок 4.3 - Result of training LSTM model

### 4.4 LSTM performance

Перевіримо точність нашого алгоритму на тестовому наборі даних. Для цього створимо масив з передбачень відповідно до ймовірності того, що стаття фейкова: якщо ймовірність більша за 0.5, то це правдива стаття, менша - фейк.

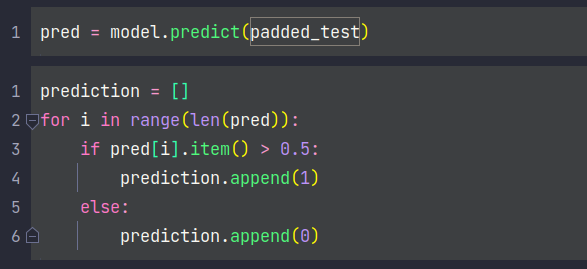


Рисунок 4.4 - Make a prediction

Тепер скористаємося sklearn.metrics можливостями для визначення точності алгоритму.

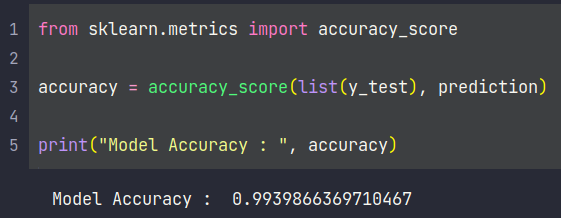


Рисунок 4.5 - Calculate accuracy score

### 4.5 Logistic Regression performance

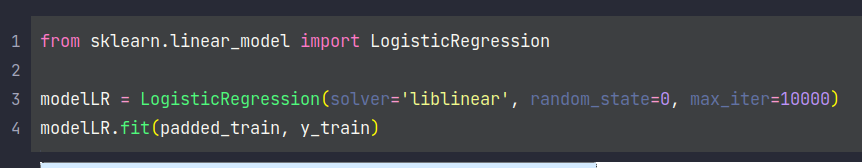
Застосуємо Logistic Regression для того самого набору даних і знайдемо точність. 

Рисунок 4.6 - Logistic Regression

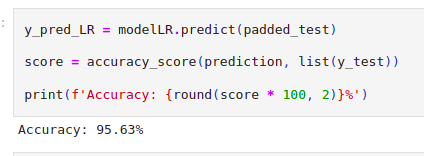


Рисунок 4.6 - Calculate accuracy score for Logistic Regression

Бачимо, що в даному випадку алгоритм відпрацював з ефективністю наближеною до 96%.

### 4.6 Порівняння ефективності методів LSTM та Logistic Regression

Зробивши порівняння двох натренованих моделей можна зробити висновок, що підхід з використанням LSTM є трішки ефективнішим в порівнняні з Logistic Regression.

Можливі варіант покращення Logistic Regression:

* Визначити Grid Search Parameters;
* Cross Validation;

**ВИСНОВКИ**

В результаті виконання курсової роботи було розроблено алгоритм, який допомагає виявити чи новина фекова. Розглянуто основні підходи для реалізація такоє моделі. Для реалізація поставленої задачі було використано мову програмування Python та різні бібліотеки: pandas, numpy, matlplot, seaborn, nltk, gensim, tensorflow.keras та інші.

На основі детального опису та проведеного аналізу предметної області інтелектуального аналізу даних для визначення фейкових новин було отримано результати з високою точністю передбачення таких новин. Підтвердженням даних висновків є результати точності виявлення фейкових новин, яка складає 0.9939866369710467 для тестового набору даних. Результати досліджень показують, що дана можель може застосовуватися за призначенням і приносити користь.

Отже, поставлені задачі були виконані, а також планується створення датасету з російськими новинами та фековими статтями, які можна буде використовувати для тренування розробленої моделі. Дана можель може допомогти розробити телеграм-бот, який допоможе фільтрувати новині стрічки.

## 

## ПЕРЕЛІК ПОСИЛАНЬ

1. <https://stackoverflow.com/questions/51956000/what-does-keras-tokenizer-method-exactly-do>
2. <https://machinelearningknowledge.ai/keras-tokenizer-tutorial-with-examples-for-fit_on_texts-texts_to_sequences-texts_to_matrix-sequences_to_matrix/#>
3. <https://keras.io/api/preprocessing/>
4. <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
5. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
6. Лекційні матеріали
7. Machine Learning. Coursera. Author: Andrew Ng
8. <https://keras.io/api/layers/activations/>
9. <https://www.tensorflow.org/guide/keras/sequential_model>
10. <https://en.wikipedia.org/wiki/Vanishing_gradient_problem>
11. <https://en.wikipedia.org/wiki/Backpropagation>

## ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ

<https://github.com/Dichik/FakeNewsML> - public repository with resources and code.