

МІЖКРАЇНОВИЙ ПРОГНОЗ ЕФЕКТИВНОСТІ ЕНЕРГЕТИЧНИХ СИСТЕМ¹**Чигрин О.Ю.,**

доктор економічних наук, доцент,
Сумський державний університет,
e-mail: o.chygryn@biem.sumdu.edu.ua
ORCID <http://orcid.org/0000-0002-4007-3728>

Койбічук В. В.,

кандидат економічних наук, доцент,
Сумський державний університет
e-mail: v.koibichuk@uabs.sumdu.edu.ua
<http://orcid.org/0000-0002-3540-7922>

Кочережченко Р. Д.,

студент спеціальності «Економіка»,
Сумський державний університет,
e-mail: r.kocherezhchenko@student.sumdu.edu.ua
ORCID: 0000-0001-7269-4177

На сьогодні автоматизований аналіз даних є продуктивним, потужним інструментом для прийняття важливих рішень щодо вдосконалення підходів оцінювання ефективності бізнесу, є основою та підґрунтям для формування думки про стан підприємства, країни, її економіки та майже будь-якої системи, ключові детермінанти котрої можна представити у вигляді кількісних та якісних даних. Разом з тим в умовах постійного росту кількості інформації в онлайн середовищі, багато класичних методів опинились під питанням ефективності для використання великих даних та нечітких груп даних таких, як відео, звук, фото та інші. Тому для оптимального вирішення завдань, що можуть дати певні гарантії для якості моделі, можна обрати більш новітній інструментарій та методи, що використовують інструменти нейромережевого моделювання та автоматизації цих процесів. Питання ефективності використання електроенергії завжди стояло константою як для України, так і світу. Метою статті є розроблення моделі для прогнозу ефективності енергетичних систем 122 країн світу (країн Європейського союзу, Північної та Південної Америки, Африки, Азії, країн з високим рівнем доходу, країн з рівнем доходу нижче середнього, країн з низьким рівнем доходу), що ґрунтується на основі значень часового ряду «чистий імпорт електроенергії як частка попиту» за період з 2000 року по 2021 рік. У статті даний детальний опис та інструкції для відтворення моделі, пояснення для глибокого розуміння результату роботи моделі, також наведено детальний опис дослідження якості моделі, метрик, що можуть допомогти в аналізі якості моделі. Проведена робота надає приклад в форматі фото кодів, які було використано для роботи с даними, реалізації метрик, створення моделі. Для реалізації моделі були використані інструменти та інфраструктура мови програмування Python, що дозволило створити модель без занурення в інженерні деталі побудови подібних моделей. В результаті дослідження було отримано модель, яка в середньому дає похибку в 6.9 відсотка, що є адекватним результатом для прогнозних моделей. В результаті було визначено, що модель може бути базисом для інших прогнозних моделей, це дає основу для побудови прогнозного дослідження.

Ключові слова: чистий імпорт електроенергії як частка попиту, нейронечіткі технології, середня абсолютна помилка, ансамблевий метод машинного навчання, автоматизація, прогноз.

DOI: 10.21272/1817-9215.2022.4-19

ВСТУП

У сучасних організованих системах якість прийняття рішень цілком і повністю залежить від якості наданої інформації. Визначаючи масштаби користування електроенергією у світі, питання щодо оцінки та аналізу ефективності використання електроенергії стоїть особливо критично в умовах нестачі цього ресурсу. Прогноз містить в собі інформацію, що буде, але й дозволяє поставити запитання чому він такий, яку інформацію відображають цифри. Подібні знання також стають у нагоді при порівнянні результатів аналізу між суміжними сутностями, адже це може надати ті самі відповіді «чому?» та «як?». Чому в певній країні результати кращі за інші, як

¹ Це дослідження фінансується за рахунок грантів Міністерства освіти і науки України № 0122U000788, 0120U102002.

досягти такого результату як призери в енергоефективності. Прогноз це частка та невід’ємний інструмент такого дослідження, в той час як аналіз дозволяє отримати ретроспективну інформацію про стан, прогноз надає можливі дані про майбутнє, прогноз може стимулювати прийняття рішень для покращення становища або підтвердити ефективність рішень прийнятих в минулому.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Велика кількість публікацій вітчизняних та зарубіжних науковців-теоретиків та науковців-практиків присвячена дослідженням ефективності енергетичних систем. Лише за останні п’ять років (за період з 2017 року по 2022 рік) було знайдено 148 905 документів, що індексуються базою даних Скопус (рис. 1).

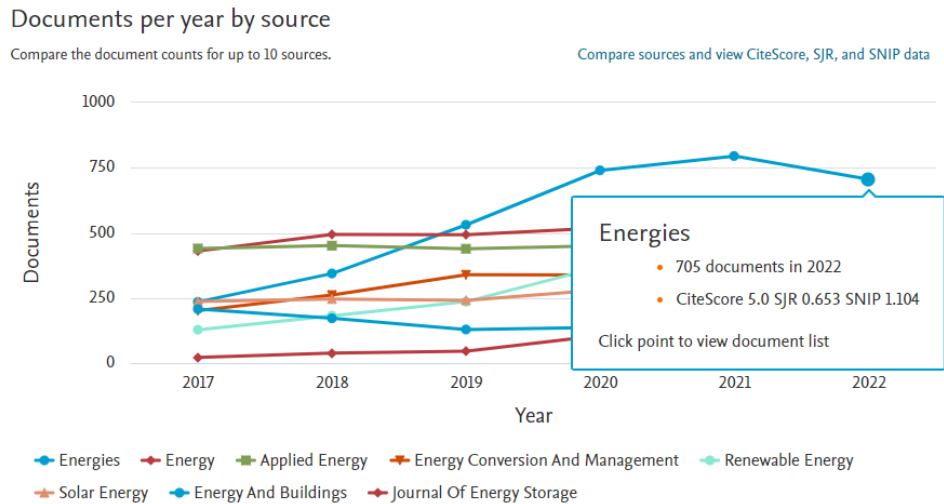


Рисунок 1 – Публікаційна активність науковців всього у журналах, обліковуваних БД Скопус за тематикою «ефективність енергетичних систем» за період 2017 – 2022 рр.

Окремо слід відзначити працю [1], в якій автори описують методологію розумного аналізу ключових показників ефективності (КРІ), продуктивності якості та безпеки енергетичних систем, реалізація якої здійснена за допомогою виконання двох комплексних етапів – розрахунку базових КРІ, динамічного оновлення всіх КРІ та порівняння їх з базовими. Результатом запропонованої методології є посилення захисту енергетичних систем, зниження витрат, посилення управління енергією. Також високої уваги заслуговує дослідження науковців [2], які визначають оптимальний розподіл спінінгових резервів (оборотних запасів) у взаємопов’язаних енергетичних системах із реагуванням на попит за допомогою двовимірної моделі прогнозування вітру та використовують інноваційний ймовірнісний метод, що враховує в енергетичній системі непередбачувані ситуації, пов’язані з відключенням генератора та невизначеностями прогнозних помилок навантаження на мережу та використання відновлюваної енергії. Модель застосована для двозонної перевірки надійності мережі та аналізу впливу міжзонального виробництва електроенергії.

ПОСТАНОВКА ЗАВДАННЯ

Метою роботи є розроблення прогнозу на основі нейронечітких технологій, а саме: створення моделі «Випадковий Ліс»; формування програмного коду для реалізації метрик якості; демонстрація прикладів коду для обробки та трансформації даних та порівняння результатів прогнозу між країнами.

МЕТОДИ ДОСЛІДЖЕННЯ

У процесі дослідження використані нейронечіткі та статистичні методи для аналізу та прогнозу часових рядів. Для автоматизації розрахунків використанні такі інструменти як: мова програмування Python з бібліотеками Numpy, Pandas, Sklearn для обробки та прогнозу, такий набір забезпечує необхідними методами для мануального та автоматичного аналізу та побудови моделей.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для отримання інформації з даних є багато методів та підходів, серед них можна виділити певні групи за певними критеріями, як математична база, клас задач для яких був створений метод, тобто за метою метода. Для реалізації є інструменти, які можна використовувати аби покращити якість та швидкість прогнозу. Знаючи це можна виділити основні компоненти при роботі з даними: дані, методи, результати. З першими двома все зрозуміло, саме словосполучення «робота над даними» приховує ці компоненти, але не результат, можна сказати що результат – це аналіз або прогноз, але результат – це досягнення цілей, що визначаються метою дослідження [3]. Потрібно розібрати ключові компоненти та розглянути, як вони відносяться до роботи. Дані – цей термін має багато значень, але всі вони зводяться до основи – сукупність значень. Проте дані не завжди можуть бути корисними, структурованими чи містити якусь потрібну інформацію, наприклад результатом роботи псевдогенератора випадкових чисел (ПГВЧ) також є дані, що самі по собі є безглуздими, але при застосуванні стохастичних методів для обробки сигналів, стають корисними [4]. Отже, в якості вхідного параметра для розроблення прогнозу ефективності енергетичних систем є часовий ряд за проміжок 2000-2021 роки для 122 індикаторів, що відповідають окремим країнам, материкам та групам країн [5] (додаток А, табл. А.1). Фрагмент вхідної вибірки подано на рисунку 2.

Entity	Code	Year	Net electricity imports as a share of demand (%)	
0	Afghanistan	AFG	2000	17.543859
1	Afghanistan	AFG	2001	14.492754
2	Afghanistan	AFG	2002	12.658228
3	Afghanistan	AFG	2003	9.615385
4	Afghanistan	AFG	2004	10.101010
...
5048	Zimbabwe	ZWE	2016	21.337946
5049	Zimbabwe	ZWE	2017	23.294859
5050	Zimbabwe	ZWE	2018	10.049261
5051	Zimbabwe	ZWE	2019	8.937198
5052	Zimbabwe	ZWE	2020	9.083469

*Рисунок 2 – Структура даних
Джерело: побудовано авторами на основі [5]*

Для отримання прогнозу ефективності енергосистем використано модель «Випадкового лісу» (Random Forest, RF). Слід розглянути цей метод машинного навчання більше детально. Перш за все, модель RF – це модель, що базується на деревах прийняття рішень. Дерева рішень – це засіб для прийняття рішень, що традиційно використовується як базовий алгоритм у машинному навчанні. Структурою служать так звані «гілки» та «листя». На гілках дерева записані ознаки, від яких залежить цільова функція, в «листях» записані значення цільової функції, а в інших вузлах – ознаки, якими відрізняються випадки. Щоб класифікувати новий випадок, треба спуститись по дереву до листка і знайти відповідне значення [6] Такі дерева використовують і окремо від машинного навчання, наприклад, як послідовну структуру-скрипт для аналізу даних для дуже малих вхідних даних, які може обробити людина. Ціль полягає в тому, щоб створити модель, що на основі змінних

на вході дає модельоване значення цільової змінної. Зазвичай, дерево рішень складається з трьох типів вузлів: вузли рішення; імовірнісні вузли та замикаючі. Прикладом такого дерева наведено на рисунку 3.

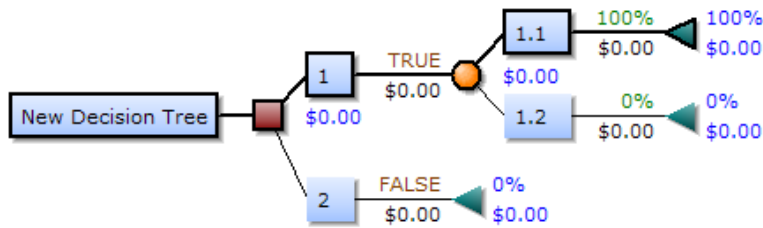


Рисунок 3 – Приклад дерева
Джерело: побудовано авторами

В графі (рис. 2) квадратні блоки – це вузли рішення, круглі – імовірнісні вузли і кінцеві, замикаючі – круглі. Дерева можна розподілити за класом задач на: дерева для класифікації, якщо кінцева змінна має класовий характер, та дерева для регресії, якщо цільова змінна є числом. Це впливає на критерії розподілу змінних. Існують методи, які дозволяють використовувати дерева рішень, створюючи ансамблі моделей. Ансамбль моделей у статистиці та машинному навчанні використовує кілька навчених алгоритмів з метою отримання кращої ефективності прогнозування. Серед таких можна виділити: «Бегінг» – це найбільш раннє дерево рішень, буде кілька дерев рішень, неодноразово інтерполюючи дані із заміною, і дерева рішень для прогнозу консенсусу; RF – використовує ряд дерев рішень, з метою поліпшення значення класифікації. Саме через механізм ансамблю в RF, і буде використаний цей метод, тому що тільки так в умовах реальних даних можна отримати якісний прогноз. Ці методи потребують набагато більше уваги для глибокого розуміння, але для отримання результатів прогнозу потрібно розуміти лише інтерфейс для роботи з моделлю, такими як: вхідні дані та гіперпараметри моделі [7, 8]. Результати – це певні метрики, які слугують критеріями якості поєднання даних та методів, або нові дані, що описують, наприклад, результати кластеризації, виявлення аномалій чи зменшення розмірності даних. Варто зауважити такі метрики залежать від вхідних даних та методів, це прямо впливає на якість прогнозу, наприклад, якщо взяти занадто малу вибірку або неправильний метод то результати будуть лише для того аби показати, що не підходить. Отже, існує багато метрик для оцінки якості моделей, прогнозу, класифікації, в даному дослідженні для побудованої моделі RF використана абсолютна середня помилка (mean absolute error, MAE), що дозволяє оцінити якість розробленого прогнозу, на основі розрахунку похибки прогнозу на один крок вперед [9]:

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (1)$$

Крім того, для оцінки якості моделі можна використати й метрику середньоквадратичної помилки (mean square error, MSE) [7]:

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (2)$$

В класичних статистичних методах для перевірки моделей на точність допускають значення абсолютної помилки до 10% [10], проте, чим ближче значення метрик (1, 2) до нуля, тим краще модель.

Отже, виділимо наступні кроки для розроблення прогнозу ефективності енергетичних систем країн: обрати інструменти; виділити метрики якості; здійснити загальне дослідження даних на можливість їх застосування у прогнозній моделі; фомування вибірки змінних; розподіл даних на тренувальну та валідаційну частину; тренування моделі.

За результатом покрокового виконання цього алгоритму буде отримано робочу прогнозу модель. В якості інструментів для створення прогнозу були обрані технології, які являються оптимальним варіантом для вирішення таких задач. Мова програмування Python – мова для вирішення загальних задач, не прив’язана до якогось конкретного домену, базовий функціонал добре розширюється бібліотеками [11]. Бібліотеки Numpy [12] та Pandas [13], створенні для векторизації даних, організації в зручні структури та швидкості обчислень завдяки паралелізації обчислень. Для роботи з моделями машинного навчання – програмна бібліотека Sklearn [14], а в якості оточення для полегшення роботи зі змінними було обрано Jupyter Notebook [15]. Для встановлення пакетів варто виконати код локального встановлення необхідних пакетів (рис. 3). Код демонструє команди, що можуть знадобитися, якщо необхідні інструменти-пакети не встановлені локально. Також всі команди, зображені на рис. 3, можна виконати окремо в терміналі чи іншому командному середовищі, яке залежить від системи. Дякуючи архітектурі мови Python, не знадобиться розробляти окремі версії для кожної операційної системи: Windows, OSX, Linux. Python сам подбає про належне виконання коду.

```
▷ %pip install numpy
  %pip install pandas
  %pip install matplotlib
  %pip install seaborn
  %pip install sklearn
[1] ✓ 12.5s
```

Рисунок 4 – Код локального встановлення необхідних пакетів
Джерело: побудовано авторами

Після локального встановлення їх можна використовувати будь де, в залежності від задач та завдань тематики дослідження. Наступний крок – ініціалізація в проєкті (рис. 5.). Код демонструє загальний синтаксис команд локальної ініціалізації пакетів в проєкті. Без цих команд середовище виконання коду не зрозуміє, що потрібно робити.

```
▷ import numpy as np
  import pandas as pd
  import matplotlib.pyplot as plt
  import seaborn as sns
  from sklearn.impute import SimpleImputer
  from sklearn.ensemble import RandomForestRegressor
[658]
```

Рисунок 5 – Код імпорту пакетів в проєкт
Джерело: побудовано авторами

Для отримання результатів визначення якості прогнозу варто розробити функції для отримання MAE (1) та MSE метрик (2), також тут є приклади реалізації інших метрик, які теж можуть послугувати як метрики для вдосконалення якості моделі (рис. 6). Для полегшення роботи с кодом кожен метрику можна винести в чисту функцію, яка приймає деякі аргументи, в контексті прогнозної моделі це вектори прогнозних значень та реальних та результат.

```

def mape(y_true, y_pred):
    ape = np.abs((y_true - y_pred) / y_true)
    # ape[~np.isfinite(ape)] = 0. # VERY questionable
    # ape[~np.isfinite(ape)] = 1. # pessimist estimate
    return np.mean(ape)

def wmape(y_true, y_pred):
    return np.sum(np.abs(y_true - y_pred)) / np.sum(np.abs(y_true))

def mse(y_true, y_pred):
    return np.sum((y_true - y_pred)**2) / y_true.shape[0]

def rmse(y_true, y_pred):
    return np.sqrt(mse(y_true, y_pred))

def mae(y_true, y_pred):
    return np.sum(np.abs(y_true - y_pred)) / y_true.shape[0]

```

Рисунок 6 – Код функцій метрик
Джерело: побудовано авторами

Для виведення статистики та зручності тестуванні якості моделі доцільно використати код (рис. 7):

```

def estimate(y_true, y_pred):
    print("-----Metrics-----\n")
    print(f"MSE: {mse(y_true, y_pred)}")
    print(f"MAE: {mae(y_true, y_pred)}")

```

Рисунок 7 – Код функції-обгортки
Джерело: побудовано авторами

Тепер, маючи все необхідне, можна приступити до аналізу. Для аналізу потрібні дані, їх необхідно імпортувати в вхідного файлу (рис. 8).

```

df_f = pd.read_excel("../final.xlsx")
df_f_with_year = df_f
df_f = df_f.set_index("Entity")

```

Рисунок 8 – Код імпорту датасету
Джерело: побудовано авторами

Для того, аби мати уявлення як виглядають дані, можна викликати head-метод та отримати табличну репрезентацію даних (рис. 9). Синтаксис методів *object.method(args)* активно використовується при роботі з бібліотеками. В коді показаному на рис. 9, зображений приклад, де об'єкт – df_f, а метод – це head.

Entity	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	...	2012	2013	2014	20
Afghanistan	17.543859	14.492754	12.658228	9.615385	10.101010	9.708738	33.858269	39.102566	50.675674	55.023926	...	77.721519	76.532768	76.180702	76.2096
Africa	0.893194	0.787330	0.488582	0.430911	0.201433	0.203085	0.538536	0.824676	0.748418	0.837663	...	1.140127	1.266437	1.119095	1.0919
Albania	17.331022	31.934307	36.068375	15.540541	7.881773	6.368330	9.951060	49.736378	39.004814	21.212122	...	34.986225	25.000000	39.409500	19.2043
Algeria	-0.293871	-0.200160	-0.076982	0.035945	0.034014	0.250156	0.240964	0.028588	-0.132345	-0.082919	...	-0.092696	-0.160057	-0.315562	-0.0464
Argentina	1.410731	1.994108	6.713416	5.399313	3.499395	3.704057	2.074360	6.415668	4.557930	5.029803	...	5.583682	5.567054	6.969182	6.1052

Рисунок 9 – Код Для виведення перших записів
Джерело: побудовано авторами

З допомогою геттера `shape` можна отримати розмірність даних (рис. 10). Синтаксиси геттерів подібні до синтаксисів методів, але без знаків «(» та «)» в кінці.

```
df_f.shape
[112] ✓ 0.5s
... (122, 22)
```

Рисунок 10 – Код для отримання розмірності даних
Джерело: побудовано авторами

Як зображено на рис. 11, це 122 рядки, що відповідають країнам дослідження [5], та 22 колонки, або роки. За допомогою `describe` метода можна отримати базову статистику по рокам (рис. 10). Метод `describe` повертає по колонкам таку базову інформацію, як: `count` (кількість записів), `mean` (середнє), `std` (стандартне відхилення), `min` (мінімальне значення), `25%` (ніжній кватиль), `50%` (медіана), `75%` (верхній кватиль), `max` (максимальне значення). Варто зауважити, що змінити результат виклику метода можна за допомогою аргументів, що передаються в нього, які зазначені в інструменті Pandas [13].

```
df_f.describe()
✓ 0.1s
```

	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	...
count	121.000000	121.000000	121.000000	121.000000	121.000000	122.000000	122.000000	122.000000	122.000000	122.000000	...
mean	-6.989960	-3.586532	-3.714321	-3.309325	-2.954945	-0.801356	1.432305	-0.009147	-0.495058	1.473086	...
std	91.762046	74.815615	75.380701	77.536108	74.983660	66.324637	61.503236	70.831704	77.567839	58.589826	...
min	-844.206726	-683.071533	-701.006714	-741.871887	-703.281250	-638.338196	-606.233398	-558.292114	-603.061218	-480.683044	...
25%	-1.376327	-1.802651	-1.830449	-2.387640	-1.841295	-1.957879	-1.684286	-1.448078	-1.267766	-1.852661	...
50%	0.446177	0.278552	0.449856	0.202114	0.209179	0.446152	0.341821	0.440371	0.310553	0.315584	...
75%	8.051949	10.382949	9.237875	9.615385	9.090909	9.526470	10.041922	8.992248	9.941420	13.092566	...
max	100.000000	98.578201	94.285713	88.135597	87.878784	85.885887	90.379005	89.106148	90.610321	89.438210	...

8 rows x 22 columns

Рисунок 11 – Код для отримання базової статистики
Джерело: побудовано авторами

Якщо транспонувати датасет за допомогою геттера «`T`», то можна отримати інформацію і по країнам (рис. 12).

```
df_f.T.describe()
[11] ✓ 0.3s
...
```

	Afghanistan	Africa	Albania	Algeria	Argentina	Armenia	Asia	Austria	Azerbaijan	Bahrain	...	Ukraine
count	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	...	22.000000
mean	54.713774	0.660250	22.437385	-0.131046	5.304652	-13.491646	0.084810	6.712323	-0.717938	0.138866	...	-3.344323
std	30.564648	0.405402	17.453278	0.179315	1.863028	7.631070	0.059327	4.558640	4.470266	0.429311	...	1.434817
min	9.615385	-0.335828	-14.350453	-0.477796	1.410731	-26.622295	-0.001543	-2.342681	-6.872205	-0.893697	...	-6.137300
25%	21.622462	0.501070	11.348430	-0.247285	3.917526	-18.656562	0.032272	3.577289	-4.160154	0.000000	...	-4.524283
50%	68.403612	0.767874	23.569774	-0.146201	5.844444	-15.467885	0.102023	7.870209	-1.660561	0.000000	...	-3.175311
75%	78.049513	0.905252	35.797837	-0.054048	6.711737	-7.482719	0.129642	10.059705	2.462368	0.489680	...	-2.313919
max	101.367732	1.266437	49.736378	0.250156	7.467691	-0.366972	0.182308	13.989710	7.490809	0.795136	...	-0.842224

8 rows x 122 columns

Рисунок 12 – Код для транспонування даних
Джерело: побудовано авторами

Для моделі такий тип даних не підходить, тому необхідно використовуючи їх перетрансформувати в форму, що буде підходити для прогнозу моделі (рис. 13). Ця функція корисна для переведення DataFrame у формат, де один або більше стовпців є змінними-ідентифікаторами (“`id_vars`”), тоді як усі інші стовпці, що вважаються

вимірюваними змінними (“value_vars”), «не повертаються» до осі рядка, залишаючи лише два неідентифікаторних стовпці, ‘змінна’ і ‘значення’.

```
df_melt = df_f_with_year.melt(
    id_vars=["Entity"],
    var_name="Year",
    value_name="Net electricity imports as a share of demand (%)")
df_melt.head()
```

	Entity	Year	Net electricity imports as a share of demand (%)
0	Afghanistan	2000	17.543859
1	Africa	2000	0.893194
2	Albania	2000	17.331022
3	Algeria	2000	-0.293871
4	Argentina	2000	1.410731

Рисунок 13 – Код для форматування даних для потреби моделі
Джерело: побудовано авторами

Тепер, маючи метрики та підготовлені дані, можна почати розробляти модель. Першим кроком буде розбиття існуючих даних на дві підгрупи – тренувальні та валідаційні. Потрібно це для того, щоб модель для прогнозу не була занадто налаштована для існуючих даних, не перенавчилася. Це зробить модель більш універсальною і навіть, якщо зміниться формат даних, то модель все ще буде якісно давати прогноз. Для розбиття можна використовувати свої рішення або бібліотеки, для прикладу подана реалізація без бібліотек (рис. 14).

```
df_train = df_melt[df_melt["Year"] < 2010].copy()
df_valid = df_melt[df_melt["Year"] >= 2010].copy()
```

Рисунок 14 – Код для розподілу даних
Джерело: побудовано авторами

В даному прикладі (рис. 14) вибірки розподілені за часовим параметром, тренувальні – це всі записи, рік яких нижче за 2010, в той час, як записи більше за 2009, йдуть в валідаційні. Такий розподіл даних може здатися незвичним, але для часових рядів це дозволить уникнути перепідгонки даних. Для побудови моделі потрібні додаткові змінні, які покращать ефективність роботи з моделлю та самої моделі. Серед таких змінних є: значення наступного року; середнє за 4 минулі роки; різниця між значенням цільової змінної для поточного та минулого року; значення минулого року. Для початку треба встановити значення наступного року, або «target next year» змінну (рис. 15).

```
df_train["target next year"] = df_train.groupby(
    "Entity")["Net electricity imports as a share of demand (%)"].shift(-1)
df_train
```

	Entity	Year	Net electricity imports as a share of demand (%)	target next year
0	Afghanistan	2000	17.543859	14.492754
1	Africa	2000	0.893194	0.787330
2	Albania	2000	17.331022	31.934307
3	Algeria	2000	-0.293871	-0.200160
4	Argentina	2000	1.410731	1.994108
...
1215	Uzbekistan	2009	-0.112233	NaN
1216	Venezuela	2009	-0.318061	NaN
1217	Vietnam	2009	4.471350	NaN
1218	Zambia	2009	-6.250000	NaN
1219	Zimbabwe	2009	15.256257	NaN

Рисунок 15 – Додавання змінної значення наступного року
Джерело: побудовано авторами

Наприклад, для першої країни, Афганістану, дані виглядають так (рис. 16):

```
df_train[df_train["Entity"] == "Afghanistan"]
```

	Entity	Year	Net electricity imports as a share of demand (%)	target next year
0	Afghanistan	2000	17.543859	14.492754
122	Afghanistan	2001	14.492754	12.658228
244	Afghanistan	2002	12.658228	9.615385
366	Afghanistan	2003	9.615385	10.101010
488	Afghanistan	2004	10.101010	9.708738
610	Afghanistan	2005	9.708738	33.858269
732	Afghanistan	2006	33.858269	39.102566
854	Afghanistan	2007	39.102566	50.675674
976	Afghanistan	2008	50.675674	55.023926
1098	Afghanistan	2009	55.023926	NaN

Рисунок 16 – Приклад зі змінною значення наступного року
Джерело: побудовано авторами

Маючи значення наступного року для кожного запису, можна і встановити дані минулих років (17).

```
df_train["lag"] = df_train.groupby("Entity")["Net electricity imports as a share of demand (%)"].shift(1)
df_valid["lag"] = df_valid.groupby("Entity")["Net electricity imports as a share of demand (%)"].shift(1)
df_train
```

Рисунок 17 – Додавання значень минулих років
Джерело: побудовано авторами

Тепер дані виглядають так, як показано на рисунку 18.

```
df_train[df_train["Entity"] == "Afghanistan"]
```

	Entity	Year	Net electricity imports as a share of demand (%)	target next year	lag
0	Afghanistan	2000	17.543859	14.492754	NaN
122	Afghanistan	2001	14.492754	12.658228	17.543859
244	Afghanistan	2002	12.658228	9.615385	14.492754
366	Afghanistan	2003	9.615385	10.101010	12.658228
488	Afghanistan	2004	10.101010	9.708738	9.615385
610	Afghanistan	2005	9.708738	33.858269	10.101010
732	Afghanistan	2006	33.858269	39.102566	9.708738
854	Afghanistan	2007	39.102566	50.675674	33.858269
976	Afghanistan	2008	50.675674	55.023926	39.102566

Рисунок 18 – Приклад зі змінними даних минулих років
Джерело: побудовано авторами

Наступним кроком буде встановлення різниці між значенням цільової змінної для поточного та минулого року. (рис. 19):

```
df_train["diff_net"] = df_train.groupby("Entity")["Net electricity imports as a share of demand (%)"].diff(1)
df_valid["diff_net"] = df_valid.groupby("Entity")["Net electricity imports as a share of demand (%)"].diff(1)
df_train
```

Рисунок 19 – Додавання змінної різниці
Джерело: побудовано авторами

Додатковий стовпчик зі змінною матиме вигляд, як показано на рисунку 20.

```
df_train[df_train["Entity"] == "Afghanistan"]
```

Entity	Year	Net electricity imports as a share of demand (%)	target next year	lag	diff_net	
0	Afghanistan	2000	17.543859	14.492754	NaN	NaN
122	Afghanistan	2001	14.492754	12.658228	17.543859	-3.051105
244	Afghanistan	2002	12.658228	9.615385	14.492754	-1.834526
366	Afghanistan	2003	9.615385	10.101010	12.658228	-3.042843
488	Afghanistan	2004	10.101010	9.708738	9.615385	0.485625
610	Afghanistan	2005	9.708738	33.858269	10.101010	-0.392272
732	Afghanistan	2006	33.858269	39.102566	9.708738	24.149530
854	Afghanistan	2007	39.102566	50.675674	33.858269	5.244297
976	Afghanistan	2008	50.675674	55.023926	39.102566	11.573109

Рисунок 20 – Приклад даних з різницею
Джерело: побудовано авторами

Залишилось лише додавання середнього до даних (рис. 21):

```
df_train["mean_net_4"] = df_train.groupby("Entity")["Net electricity imports as a share of demand (%)"].rolling(4).mean().reset_index(level=0, drop=True)
df_valid["mean_net_4"] = df_valid.groupby("Entity")["Net electricity imports as a share of demand (%)"].rolling(4).mean().reset_index(level=0, drop=True)
```

Рисунок 21 – Додавання середнього значення до даних
Джерело: побудовано авторами

Тепер дані мають всі необхідні змінні для розроблення найкращого прогнозного результату. Першим етапом для тренування моделі є етап вибору змінних. В великих даних цей процес можливо би потребував навіть інших моделей для виявлення аномалій чи зменшення розмірності, але у випадку з часовими рядами такого формату такої проблеми не стоїть. Серед існуючих змінних для тренування будуть обрані змінні, що подані на рисунку 22.

```
features = ["Net electricity imports as a share of demand (%)", "lag", "diff_net", "mean_net_4"]
```

Рисунок 22 – Змінні для тренування моделі
Джерело: побудовано авторами

Перший варіант моделі (рис. 23) відповідає її тренуванню.

```
imputer = SimpleImputer()
Xtr = imputer.fit_transform(df_train[features])
ytr = df_train['target next year']

mdl = RandomForestRegressor(n_estimators=100, random_state=0, n_jobs=6)
mdl.fit(Xtr, ytr)
```

Рисунок 23 – Перший варіант моделі
Джерело: побудовано авторами

Тут слід звернути увагу на 2 ключових момента – це перетворення змінних за допомогою моделі SimpleImputer, та тренування моделі RF за допомогою RandomForestRegressor, він приймає 3 параметри (рис. 23), а саме: n_estimators – кількість дерев в ансамблі, в цій моделі 100, 100 дерев рішень, random_state – випадковий початковий стан моделей, для моделі було обрано 0, та n_jobs – кількість паралельних процесів, в нашому випадку 6. Для покращення якості їх зазвичай підбирають, опираючись на результати метрик, але зараз це не має особливого значення, така підгонка може призвести до занадто перенавченої моделі. Маючи натреновану модель вже можна перевірити її результативність (рис. 24).

```

Xval = imputer.transform(df_valid[features])
yval = df_valid['target next year']

p = mdl.predict(Xval)

estimate(yval, p)

-----Metrics-----
MSE: 159.17396006776428
MAE: 4.76289489475359

```

Рисунок 24 – Тест першого варіанту моделі
Джерело: побудовано авторами

Середня абсолютна похибка дала результат в 4.76 відсотка, в таких межах це може вважатися цілком адекватним результатом [10]. Наступним етапом є розширення даних аби отримати реальні прогнози, в датасеті (додаток 1) вже є зміна «target next year», що містить інформацію про стан цільової змінної в наступній часовій позначці. Тобто тепер потрібно розширити дані, створивши зміну, що містить стан цільової змінної в році після наступного «net_next_next_year» (рис. 25).

```

df_train['net_next_next_year'] = df_train.groupby("Entity")["Net electricity imports as a share of demand (%)"].shift(-2)
df_valid['net_next_next_year'] = df_valid.groupby("Entity")["Net electricity imports as a share of demand (%)"].shift(-2)

```

Рисунок 25 – Задавання змінної для прогнозу
Джерело: побудовано авторами

Тепер дані можна переглянути, наприклад, для України, скориставшись наступною командою, код команди наведено на рисунку 26.

```

df_train[df_train['Entity'] == "Ukraine"].head()

```

Entity	Year	Net electricity imports as a share of demand (%)	target next year	lag	diff net	mean_net_4	net_next_next_year
112	Ukraine	2000	-2.299606	-1.802651	NaN	NaN	-1.830449
234	Ukraine	2001	-1.802651	-1.830449	-2.299606	0.496955	-2.818670
356	Ukraine	2002	-1.830449	-2.818670	-1.802651	-0.027798	-3.016583
478	Ukraine	2003	-2.818670	-3.016583	-1.830449	-0.988221	-4.702636
600	Ukraine	2004	-3.016583	-4.702636	-2.818670	-0.197913	-5.711473

Рисунок 26 – Приклад даних зі змінною для прогнозу
Джерело: побудовано авторами

Тепер можна зробити прогноз на 2022 рік. Для цього потрібно взяти дані за 2021 рік (рис. 27) та використати модель (рис. 28).

```

new_examples = df_valid[df_valid['Year'] == 2021].copy()

```

Entity	Year	Net electricity imports as a share of demand (%)	target next year	lag	diff net	mean_net_4	net_next_next_year
2562	Afghanistan	2021	101.367732	NaN	86.291740	15.075992	87.727631
2563	Africa	2021	0.615815	NaN	-0.150996	0.766812	0.180729
2564	Albania	2021	22.139548	NaN	30.039526	-7.899978	17.984234
2565	Algeria	2021	-0.262993	NaN	-0.192678	-0.070315	-0.190881
2566	Argentina	2021	7.050346	NaN	3.306655	3.743691	6.059770
2679	Uzbekistan	2021	-1.065879	NaN	2.214334	-3.280213	0.654838
2680	Venezuela	2021	-0.824846	NaN	-0.845892	0.021046	-0.892280
2681	Vietnam	2021	1.631233	NaN	0.602119	1.029113	0.894295
2682	Zambia	2021	-1.029064	NaN	-5.131579	4.102514	-4.614627
2683	Zimbabwe	2021	1.456240	NaN	9.083469	-7.627229	7.381542

Рисунок 27 – Виділення змінних для 2021 року
Джерело: побудовано авторами

```
p = mdl.predict(new_examples[features])
p
✓ 0.7s
```

Рисунок 28 – Прогноз для 2022 року
Джерело: побудовано авторами

Результат використання цієї моделі слід розподілити за новими змінними (рис. 29).

```
new_examples['p_net_next_year'] = p[:, 0]
new_examples['p_net_next_next_year'] = p[:, 1]
✓ 0.5s
```

Рисунок 29 – Розподіл результату прогнозу за змінними
Джерело: побудовано авторами

Отже, тепер можна подивитися на результат (рис. 30).

```
new_examples.head()
```

Entity	Year	Net electricity imports as a share of demand (%)	target next year	lag	diff_net	mean_net_4	net_next_next_year	p_net_next_year	p_net_next_next_year
2562	Afghanistan	2021	101.367732	NaN	86.291740	15.075992	87.727631	NaN	93.075273
2563	Africa	2021	0.615815	NaN	-0.150996	0.766812	0.180729	NaN	1.577378
2564	Albania	2021	22.139548	NaN	30.039526	-7.899978	17.984234	NaN	24.750111
2565	Algeria	2021	-0.262993	NaN	-0.192678	-0.070315	-0.190881	NaN	0.148154
2566	Argentina	2021	7.050346	NaN	3.306655	3.743691	6.059770	NaN	7.022594

Рисунок 30 – Результат прогнозу для 2022 року
Джерело: побудовано авторами

Потрібно якось оцінити результат роботи натренованої моделі, маючи прогноз на 2022 рік цього неможливо зробити, хіба що трохи почекати, коли дані будуть опубліковані у відкритих даних та порівняти із обчисленими. Але можна зробити прогноз на 2021 рік і порівняти з реальним. Як і в минулому прикладі з 2021 роком потрібно взяти 2020 рік (рис. 31).

```
new_examples = df_valid[df_valid["Year"] == 2020].copy()
new_examples
```

Entity	Year	Net electricity imports as a share of demand (%)	target next year	lag	diff_net	mean_net_4	net_next_next_year
2440	Afghanistan	2020	86.291740	101.367732	82.244560	4.047180	81.986038
2441	Africa	2020	-0.150996	0.615815	-0.335828	0.184832	0.254683
2442	Albania	2020	30.039526	22.139548	31.668858	-1.629332	22.240734
2443	Algeria	2020	-0.192678	-0.262993	-0.182887	-0.009791	-0.244582
2444	Argentina	2020	3.306655	7.050346	7.467691	-4.161036	6.056113
...
2557	Uzbekistan	2020	2.214334	-1.065879	2.131468	0.082866	0.642660
2558	Venezuela	2020	-0.845892	-0.824846	-0.832217	-0.013675	-0.952293
2559	Vietnam	2020	0.602119	1.631233	0.580504	0.021615	0.578427
2560	Zambia	2020	-5.131579	-1.029064	-5.121471	-0.010108	-4.910537
2561	Zimbabwe	2020	9.083469	1.456240	8.937198	0.146272	12.841197

122 rows x 8 columns

Рис 31 – Виділення даних для 2020 року
Джерело: побудовано авторами

Далі потрібно використати модель (рис. 32) та розподілити за змінними (рис. 33):

```
p = mdl.predict(new_examples[features])
p
✓ 0.9s
```

Рисунок 32 – Прогноз для 2021 року
Джерело: побудовано авторами

```
new_examples['p_net_next_year'] = p[:, 0]
new_examples['p_net_next_next_year'] = p[:, 1]
```

Рисунок 33 – Розподіл по змінним для 2021 року
Джерело: побудовано авторами

Наступним кроком є перенесення даних до валідаційного датасету (рис. 34).

```
df_valid["2021 forecast"] = new_examples['p_net_next_year']
df_valid["2022 forecast"] = new_examples['p_net_next_next_year']
```

Рисунок 34 – Задавання змінних для валідаційного датасету
Джерело: побудовано авторами

Для зручності можна винести всі необхідні дані в окрему змінну (рис. 34) та отримати дані в такому форматі, як на рисунку 36:

```
year_2020_entity = year_2020.set_index("Entity")
year_2020_entity.rename(columns={"target next year": "Real 2021", "2021 forecast": "Forecast 2021", "2022 forecast": "Forecast 2022"}, inplace=True)
df_analysis = pd.DataFrame(year_2020_entity[["Real 2021", "Forecast 2021", "Forecast 2022"]])
```

Рисунок 35 – Групування даних
Джерело: побудовано авторами

df_analysis

Entity	Real 2021	Forecast 2021	Forecast 2022
Afghanistan	101.367732	78.901820	78.910221
Africa	0.615815	-0.134376	-0.283255
Albania	22.139548	31.485789	32.549805
Algeria	-0.262993	-0.090210	0.096786
Argentina	7.050346	3.511555	3.715910
...
Uzbekistan	-1.065879	1.900377	2.259483
Venezuela	-0.824846	-0.746258	-0.755886
Vietnam	1.631233	0.650440	0.800668
Zambia	-1.029064	-6.094319	-5.256528
Zimbabwe	1.456240	8.881800	15.728382

122 rows x 3 columns

Рисунок 36 – Результат групування даних
Джерело: побудовано авторами

Тепер можна проаналізувати, наскільки якісно модель прогнозує за допомогою порівняння справжніх даних у 2021 році та спрогнозованих у 2021 (рис. 37).

```
estimate(df_analysis["Real 2021"], df_analysis["Forecast 2021"])
```

```
-----Metrics-----
MSE: 245.32813447206678
MAE: 6.903912902961688
```

Рисунок 37 – Перевірка якості аналізу
Джерело: побудовано авторами

Як зображено на рисунку 36, середня похибка має значення 6.90, це означає, що модель статистично точна. Також можна подивитися дані по окремим країнам, наприклад для України (рис. 38) або для Великобританії (рис. 39)

```
df_analysis.loc["Ukraine"]
[106] ✓ 0.6s
... Real 2021      -3.402549
Forecast 2021   -1.903594
Forecast 2022   -1.848633
Name: Ukraine, dtype: float64
```

Рисунок 38 – Приклад для України
Джерело: побудовано авторами

```
df_analysis.loc["United Kingdom"]
[107] ✓ 0.5s
... Real 2021      5.548429
Forecast 2021     5.202708
Forecast 2022     3.800140
Name: United Kingdom, dtype: float64
```

Рисунок 39 – Приклад для Великобританії
Джерело: побудовано авторами

Результат роботи моделі можна зберегти в окремий файл для подальшого використання (рис. 39, 40) (додаток А).

```
df_analysis.to_excel("../results.xlsx")
[102] ✓ 0.1s
```

Рисунок 40 – Експорт фінальних даних до Excel файлу
Джерело: побудовано авторами

Результатом роботи цього коду є створений Excel файл (рис. 41, додаток А).

	A	B	C	D
	Entity	Real 2021	Forecast 2021	Forecast 2022
2	Afghanistan	101,3677319	78,9018203	78,91022072
3	Africa	0,615815388	-0,134376342	-0,283254626
4	Albania	22,1395483	31,48578932	32,54980474
5	Algeria	-0,262992796	-0,090210201	0,096786349
6	Argentina	7,050346274	3,51155369	3,715909887
7	Armenia	-21,26177173	-18,73483104	-17,69221209
8	Asia	0,109319049	0,09946077	0,072120986
9	Austria	10,56030401	3,030698997	3,645036194
10	Azerbaijan	-6,872205482	-4,430006459	-3,910766713
11	Bahrain	0,393502553	0,676405385	0,427751689
12	Belarus	1,109258817	-0,790138824	-1,349126579
13	Belgium	8,667151342	-1,409287944	-1,557186819
14	Belize	24,34774933	35,4206653	35,90488944
15	Benin	81,93539767	72,84595375	74,15013077
16	Bhutan	-202,4144807	-104,8900703	-109,6713392
17	Bosnia and Herzegovina	-26,59467852	-30,64949692	-28,22031137
18	Botswana	38,99984215	35,2262738	31,32863322
19	Brazil	3,948747286	6,665622674	7,153584299
20	Bulgaria	-18,61601426	-5,259437479	-2,366145927
21	Burkina Faso	34,80877905	30,95212103	33,8459181
22	Burundi	30,89852643	19,56039496	22,18582033
23	Cambodia	43,59524799	33,94655882	37,46828502
24	Canada	-10,71572083	-12,59677353	-9,263238591
25	China	-0,123943568	0,125033976	-0,077420375
26	Colombia	0,14054966	2,532155901	2,788406793
27	Congo	-13,24946006	-0,622483311	-0,513034575

Рисунок 41 – Фрагмент розробленого прогнозу на основі нейромодельовання
Джерело: побудовано авторами

Отже, розроблена модель, що використовує методи машинного навчання, нейронечіткі технології на прикладі методології «Випадковий Ліс», надає високоякісні прогнозні результати щодо ефективності енергетичних систем країн світу.

ВИСНОВКИ

Для отримання якісних прогнозних результатів електроспоживання в контексті різних країн та років були розглянуті теоретичні основи та термінологію щодо використання моделей «Дерева рішень» та їх ансамблевої архітектури «Випадковий ліс». Така архітектура допомогла знайти оптимальний результат прогнозу без таких неприємних ефектів, як: перетренування, недостатність моделі. Для визначення якості були розглянуті та реалізовані метрики MAE та MSE, такий набір може показати бізнес-цінність, наприклад, MAE лише покаже абсолютну похибку, що може розповісти про якість моделі для людей, які приймають рішення, та MSE-метрика, що може бути корисна для інженерів моделі нейронної мережі для покращення якості за допомогою градієнтного спуску. Для реалізації моделі прогнозу було використано мову програмування Python з використанням бібліотек Numpy, Pandas та Sklearn. Результатом теоретичного вивчення прогнозної моделі є послідовне вивчення деталей та дефініцій стосовно теоретичного базису для розуміння, які проблеми вирішують дерева рішень, та чому їх можна використовувати для створення прогнозу в енергетичній сфері. Результатом практичної реалізації є модель з абсолютною середньою похибкою – 6.90%, що означає що модель є адекватною та працездатною, її можна використовувати, як базис для прогнозування, так і як і самодостатню модель. Дослідження дає алгоритм та демонструє реалізацію послідовності дій для створення прогнозної моделі незалежно від її типу та архітектури, даючи розуміння не тільки в деталях реалізації з допомогою конкретних інструментів, але і на більш абстрактному рівні опису дій. Також продемонстрована робота з обробкою даних під потреби моделей, створення нових змінних, та перетворення даних, що також є обов'язковою практикою для отримання якісних результатів. Абсолютна середня похибка дає загальну інформацію про якість створеної моделі, але і конкретні результати теж можуть дати певну інформацію в розрізі конкретної країни, наприклад результат прогнозу для України на 2021 рік становить -1.90 значення цільової змінної «Net electricity import as share of demand», в той час як реальне значення становить -3.40, різниця між двома показниками навіть менша за очікувану похибку.

SUMMARY

Chygryn O., Koibichuk V., Kocherezhchenko R. Intercountry forecast of energy systems' efficiency

In order to obtain high-quality predictive results of electricity consumption in the context of different countries and years, theoretical foundations and terminology regarding the use of "Decision Tree" models and their ensemble architecture "Random Forest" were considered. This architecture helped to find the optimal forecast result without such unpleasant effects as: overtraining, model insufficiency. MAE and MSE metrics were considered and implemented to determine the quality, such a set can show business value, for example, MAE will only show the absolute error, which can tell the quality of the model for decision makers, and MSE metric, which can be useful for neural network model engineers for quality improvement using gradient descent. To implement the forecast model, the Python programming language was used using Numpy, Pandas and Sklearn libraries. The result of the theoretical study of the predictive model is a consistent study of details and definitions in relation to the theoretical basis for understanding what problems are solved by decision trees and why they can be used to create a forecast in the energy field. The result of practical implementation is a model with an absolute average error of 6.90%, which means that the model is adequate and workable, it can be used both as a basis for forecasting and as a self-sufficient model. The study provides an algorithm and demonstrates the implementation of a sequence of actions for creating a predictive model regardless of its type and architecture, providing insight not only in the details of implementation with the help of specific tools, but also at a more abstract level of description of actions. Also demonstrated is work with data processing to meet the needs of models, creation of new variables, and data transformation, which is also a mandatory practice for obtaining quality results. The absolute average error gives general information about the quality of the created model, but specific results can also give certain information in terms of a specific country, for example, the result of the forecast for Ukraine for 2021 is -1.90 value of the target variable "Net electricity import as share of demand", in while the true value is -3.40, the difference between the two figures is even smaller than the expected error.

Keywords: net import of electricity as a share of demand, neurofuzzy technologies, mean absolute error, ensemble method of machine learning, automation, forecast.

СПИСОК ЛІТЕРАТУРИ

1. Faria, P., Lezama, F., Vale, Z. et al. A methodology for energy key performance indicators analysis. *Energy Inform.* 4, 6 (2021), 73-81. <https://doi.org/10.1186/s42162-021-00140-0>
2. Yerzhigit, B., Mehdi, B., Vasilios, Z. (2019). Optimal allocation of spinning reserves in interconnected energy systems with demand response using a bivariate wind prediction model. *Energies.* 12 (20), 1-21. <https://doi.org/10.3390/en12203816>
3. Data Science Pipeline : Snowflake. URL: <https://www.snowflake.com/guides/data-science-pipeline> (дата звернення: 18.10.2022).
4. Writing production-ready ETL pipelines in Python / Pandas. URL: <https://www.udemy.com/course/writing-production-ready-etl-pipelines-in-python-pandas/> (дата звернення: 18.10.2022).
5. Data on Energy by Our World in Data. URL: <https://github.com/owid/energy-data> (дата звернення: 17.09.2022)
6. Denisko, D., Hoffman, M. (2018). Classification and interaction in random forests. *Proceedings of the National Academy of Sciences of the United States of America*, 115(8), 1690–1692. <http://doi.org/10.1073/pnas.1800256115>
7. Decision Tree : Geeksforgeeks. URL: <https://www.geeksforgeeks.org/decision-tree/> (дата звернення: 18.09.2022).
8. Decision Trees in Machine Learning: Towardsdatascience 2017. URL: <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052> (дата звернення: 18.09.2022).
9. Time Series Analysis and Forecasting. URL: <https://www.statgraphics.com/time-series-analysis-and-forecasting> (дата звернення 27.09.2022).
10. How to Check the Accuracy of Your Machine Learning Model. URL: <https://deepchecks.com/how-to-check-the-accuracy-of-your-machine-learning-model/#:~:text=Accuracy%20is%20a%20metric%20used,the%20total%20number%20of%20predictions> (дата звернення: 23.10.2022).
11. Python. URL: <https://www.python.org/> (дата звернення: 18.10.2022).
12. NumPy : NumPy 1.23.0 released. URL: <https://numpy.org/> (дата звернення: 18.10.2022).
13. Pandas. URL: <https://pandas.pydata.org/> (дата звернення: 18.10.2022).
14. Scikit-learn Machine Learning in Python. URL: <https://scikit-learn.org/stable/> (дата звернення: 18.10.2022).
15. Jupyter. URL: <https://jupyter.org/> (дата звернення: 03.11.2022)
16. Understanding Random Forest // Towardsdatascience: [Веб-сайт]. 2019. URL: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> (дата звернення: 19.09.2022).

REFERENCES

1. Faria, P., Lezama, F., Vale, Z. et al. (2021). A methodology for energy key performance indicators analysis. *Energy Inform.* 4, 6 73-81.
2. Yerzhigit, B., Mehdi, B., Vasilios, Z. (2019). Optimal allocation of spinning reserves in interconnected energy systems with demand response using a bivariate wind prediction model. *Energies.* 12 (20), 1-21
3. Data Science Pipeline : Snowflake (n.d.). Retrieved from <https://www.snowflake.com/guides/data-science-pipeline>
4. Writing production-ready ETL pipelines in Python. Pandas (n.d.). Retrieved from <https://www.udemy.com/course/writing-production-ready-etl-pipelines-in-python-pandas/> (дата звернення: 18.10.2022).
5. Data on Energy by Our World in Data (n.d.). URL: <https://github.com/owid/energy-data>

6. Denisko, D., Hoffman, M. (2018). Classification and interaction in random forests. *Proceedings of the National Academy of Sciences of the United States of America*, 115(8), 1690–1692. <http://doi.org/10.1073/pnas.1800256115>
7. Decision Tree : Geeksforgeeks (n.d.). Retrieved from <https://www.geeksforgeeks.org/decision-tree/>
8. Decision Trees in Machine Learning: Towardsdatascience 2017. Retrieved from <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>
9. Time Series Analysis and Forecasting (n.d.). Retrieved from <https://www.statgraphics.com/time-series-analysis-and-forecasting>
10. How to Check the Accuracy of Your Machine Learning Model (n.d.). Retrieved from <https://deepchecks.com/how-to-check-the-accuracy-of-your-machine-learning-model/#:~:text=Accuracy%20is%20a%20metric%20used,the%20total%20number%20of%20predictions>
11. Python. URL: <https://www.python.org/>
12. NumPy : NumPy 1.23.0 released (n.d.). URL: <https://numpy.org/> (дата звернення: 18.10.2022).
13. Pandas (n.d.). URL: <https://pandas.pydata.org/>
14. Scikit-learn Machine Learning in Python. Retrieved from <https://scikit-learn.org/stable/>
15. Jupyter. Retrieved from <https://jupyter.org/>
16. Understanding Random Forest // Towardsdatascience: [Веб-сайт]. 2019. Retrieved from <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

ДОДАТОК А

Таблиця А.1 – Результат розробленого прогнозу для країн дослідження

Entity	Real 2021	Forecast 2021	Forecast 2022
Afghanistan	101,3677319	78,9018203	78,91022072
Africa	0,615815388	-0,134376342	-0,283254626
Albania	22,1395483	31,48578932	32,54980474
Algeria	-0,262992796	-0,090210201	0,096786349
Argentina	7,050346274	3,511555369	3,715909887
Armenia	-21,26177173	-18,73483104	-17,69221209
Asia	0,109319049	0,09946077	0,072120986
Austria	10,56030401	3,030698997	3,645036194
Azerbaijan	-6,872205482	-4,430006459	-3,910766713
Bahrain	0,393502553	0,676405385	0,427751689
Belarus	1,109258817	-0,790138824	-1,349126579
Belgium	8,667151342	-1,409287944	-1,557186819
Belize	24,34774933	35,4206653	35,90488944
Benin	81,93539767	72,84595375	74,15013077
Bhutan	-202,4144807	-104,8900703	-109,6713392
Bosnia and Herzegovina	-26,59467852	-30,64949692	-28,22031137
Botswana	38,99984215	35,2262738	31,32863322
Brazil	3,948747286	6,665622674	7,153584299
Bulgaria	-18,61601426	-5,259437479	-2,366145927
Burkina Faso	34,80877905	30,95212103	33,8459181
Burundi	30,89852643	19,56039496	22,18582033
Cambodia	43,59524799	33,94655882	37,46828502
Canada	-10,71572083	-12,59677353	-9,263238591
China	-0,123943568	0,125033976	-0,077420375
Colombia	0,14054966	2,532155901	2,788406793
Congo	-13,24946006	-0,622483311	-0,513034575
Costa Rica	0,528527626	-3,913327898	-2,542326302
Cote d'Ivoire	-4,543218844	-12,252241	-11,60994925
Croatia	34,55573905	22,86797535	28,4539959
Czechia	-19,36063004	-15,86729845	-15,76260201
Democratic Republic of Congo	11,10732995	2,301797776	2,147557047
Denmark	18,73677783	17,47888592	18,65005713
Ecuador	-1,98028133	-6,250971231	-4,878539796
Egypt	-0,241476967	0,31289481	0,461454219
El Salvador	14,14529565	13,88896633	13,7530302
Estonia	-7,000183541	43,52968462	43,83119765
Eswatini	60,56428673	53,96403297	54,44800655
Europe	-0,222713797	-0,053013997	0,238788419
European Union (27)	0,121927658	0,776567548	0,936197846
Finland	23,02203467	20,9709661	23,61866919
France	-8,482916196	-4,537808104	-6,313104818
Georgia	2,544949199	10,29123324	8,272481678
Germany	-8,198118351	-2,656415706	-2,606160134
Ghana	-7,255643002	-11,1175402	-10,55283533
Greece	14,91488682	12,29742879	8,598054817
Guatemala	-5,369238623	-5,791545422	-5,89767276
High-income countries	0,214188704	0,099022607	0,165664441
Honduras	0,082650213	2,071566884	2,192280995
Hong Kong	24,1254033	31,01981487	33,78174048
Hungary	31,48406516	27,61242737	32,00138395
India	-0,019817092	-0,075810343	-0,03352099
Iran	-2,143227599	-0,549398353	-1,262930028
Iraq	20,58286915	13,64908561	13,58940269
Ireland	1,065790849	-0,638814368	-0,757007393

Entity	Real 2021	Forecast 2021	Forecast 2022
Israel	-11,54271011	-13,70905899	-11,40513969
Italy	11,40259844	7,95446528	8,436196661
Jordan	0,277743248	0,649722658	0,563520966
Kazakhstan	-2,603575232	-0,4855506	-0,571320521
Kenya	0,155966063	2,750220813	2,059400874
Kosovo	-2,697594623	-0,85315913	0,488477861
Kyrgyzstan	0,777014158	-0,057682374	0,133030228
Laos	-299,5240118	-214,1933924	-176,4591695
Latvia	10,95002008	21,22457459	25,07410958
Lebanon	0,750499946	5,505633473	6,056623725
Lesotho	52,46523624	55,5222953	57,427551
Libya	1,105164291	2,640640736	3,100111216
Lithuania	104,1136382	71,00141796	75,22043476
Lower-middle-income countries	-0,516588303	-0,420317977	-0,439928596
Low-income countries	5,823791759	2,953362828	2,589237098
Luxembourg	81,08633085	81,57749107	79,74323334
Macao	109,8757552	88,26274864	87,39700531
Malaysia	-0,335111759	-0,336712708	-0,297286841
Mexico	0,322244312	3,457548822	3,577231974
Moldova	80,98431687	14,15953146	12,55081144
Mongolia	22,82986263	19,08379826	23,25670345
Montenegro	-2,670538306	8,090950516	5,4342927
Morocco	11,47309179	1,066849763	3,128677011
Mozambique	2,956418969	-15,24896155	-15,77754106
Namibia	78,36235255	73,48429214	77,00612846
Nepal	39,46904521	35,68179445	30,30759924
Netherlands	1,676943042	-2,010949527	-1,180416796
Nicaragua	4,813553124	8,419202413	8,399908082
Niger	67,86179602	73,28239731	72,55299583
North America	-0,012730959	-0,047177904	-0,644672022
North Macedonia	34,31044496	36,36221022	35,9589278
Norway	-10,69960127	-10,97866445	-5,241575702
Pakistan	0,544841695	0,389187844	0,472835347
Palestine	90,61670772	87,04201942	85,12572678
Panama	-2,6003203	-3,780651582	-3,801961833
Paraguay	-132,7355427	-152,6122894	-344,3258565
Poland	4,913642263	7,334998056	6,315849344
Portugal	0,677145171	1,617921768	0,170291103
Romania	-3,561161792	5,136496773	5,994573858
Russia	-1,30267701	-0,32926602	0,153857645
Rwanda	-1,342111138	21,27914286	20,43616241
Serbia	-4,07125471	-2,167702666	-1,813099621
Slovakia	11,900519	0,36491691	0,268657383
Slovenia	-7,720386349	-14,63149129	-9,987167449
South Africa	-2,881833342	-2,792699914	-2,451767047
South America	0,03134805	-0,276378889	-0,08098154
Spain	0,634172095	4,411806059	4,069258502
Sweden	-16,97717421	-15,84739155	-14,79766385
Switzerland	-0,144565265	-7,050038452	-9,113716307
Syria	-2,216699615	-1,564355524	-1,589927284
Tajikistan	-15,82555319	-16,27379066	-14,75718609
Tanzania	0,847933315	2,205000191	2,465504756

Entity	Real 2021	Forecast 2021	Forecast 2022
Thailand	11,59247054	11,98972085	10,74029806
Togo	69,79414671	67,05713276	69,30312759
Tunisia	-0,602076936	-0,714834728	-0,716533902
Turkey	0,143529866	0,1457177	0,080543767
Turkmenistan	-19,51582757	-18,29439562	-17,82024289
Uganda	-0,488669439	-4,322559913	-4,643505458
Ukraine	-3,402549376	-1,903594466	-1,848633251
United Kingdom	5,548428828	5,202708162	3,800140182
United States	1,438539484	3,90644901	2,618632242
Upper-middle-income countries	0,003800742	-0,410217948	-0,272919573
Uruguay	-9,579278872	-0,04580331	-5,815808277
Uzbekistan	-1,065878878	1,90037725	2,259482813
Venezuela	-0,824845867	-0,746258149	-0,755885731
Vietnam	1,631232691	0,650439836	0,800668163
Zambia	-1,029064442	-6,09431906	-5,256528215
Zimbabwe	1,456240486	8,881799539	15,72838176