МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С.З.ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: **«Система підтримки прийняття рішень для планування виробництва біоенергії з органічних відходів»**

Виконала: студентка групи Іт-61з

Спеціальності 126 «Інформаційні системи та технології»
(шифр і назва)
Тригуба Інна Леонтіївна
(Прізвище та ініціали)

Керівник: к.т.н., доцент Татомир А.В.
(Прізвище та ініціали)

Рецензент: к.т.н., доцент Сиротюк С.В.
(Прізвище та ініціали)

ДУБЛЯНИ-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Заст. завідувач кафедри_____

к.т.н., доц. П.М. Луб
«____» _____2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу студентці

Тригубі Інні Леонтіївні

1. Тема роботи: «Система підтримки прийняття рішень для планування виробництва біоенергії з органічних відходів»

Керівник роботи Татомир Андрій Володимирович, доцент
затверджені наказом по університету від 30.12.2024 року № 887/к-с.

2. Строк подання студентом роботи 06.12.2025 р.

3. Вихідні дані до роботи: *дані щодо утворення органічних відходів домогосподарствами; методика розробки СППР.*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Вступ.

1. Аналіз стану планування виробництва біоенергії та існуючих інформаційних систем.

2. Вибір інструментарію для створення системи підтримки прийняття рішень планування виробництва біоенергії з органічних відходів.

3. Обґрунтування раціональної моделі прогнозування обсягів утворення відходів у домогосподарствах.

4. Результати створення системи підтримки прийняття рішень для планування виробництва біоенергії з органічних відходів.

5. Охорона праці та безпека у надзвичайних ситуаціях.

5. Результати визначення економічної ефективності від використання розробленої СППР.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових слайдів): *аналіз стану планування виробництва біоенергії та існуючих інформаційних систем; вибір інструментарію для створення системи підтримки прийняття рішень планування виробництва біоенергії з органічних відходів; обґрунтування раціональної моделі прогнозування обсягів утворення відходів у домогосподарствах; результати створення системи підтримки прийняття рішень для планування виробництва біоенергії з органічних відходів; економічна ефективність.*

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	*Татомир А.В., доцент кафедри інформаційних технологій*		
4	*Городецький І.М., доцент кафедри інженерної механіки*		

7. Дата видачі завдання 30 грудня 2024 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	*Написання першого розділу*	*30.12.24-20.02.25*	
2	*Виконання другого розділу та аркушів ілюстраційного матеріалу до нього*	*21.02-14.06.25*	
3.	*Виконання третього розділу та аркушів ілюстраційного матеріалу до нього*	*15.06-10.10.25*	
4.	*Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»*	*11.10-30.10.25*	
5.	*Оцінення ефективності запропонованої системи*	*01.11-10.11.25*	
6.	*Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу*	*11-30.11.25*	
7.	*Завершення роботи вцілому*	*01-06.12.25*	

Студентка _____ Тригуба І.Л.
(підпис)

Керівник роботи _____ Татомир А.В.
(підпис)

УДК 681.3:620.92:662.756

Система підтримки прийняття рішень для планування виробництва біоенергії з органічних відходів.

Тригуба І.Л. Кафедра інформаційних технологій – Дубляни, ЛНУВМ, 2025.

Кваліфікаційна робота: 100 с. текст. част., 21 рис., 9 табл., 15 арк. ілюстраційного матеріалу, 48 джерел.

У кваліфікаційній роботі розроблено систему підтримки прийняття рішень (СППР) для планування виробництва біоенергії з органічних відходів. Проведено аналіз сучасного стану біоенергетики, інформаційних ресурсів та програмних платформ, визначено методологічні засади планування й обґрунтовано потребу у створенні СППР.

Запропоновано концептуальну модель СППР, описано вибір інструментарію для її реалізації та здійснено побудову раціональної моделі прогнозування обсягів утворення відходів у домогосподарствах із використанням алгоритмів машинного навчання.

Розроблено функціональний прототип мультивіконного застосунку, що забезпечує введення даних, візуалізацію, навчання моделей, прогнозування та експорт результатів. Проведено оцінку економічної ефективності впровадження СППР, яка показала позитивний ефект уже з першого року експлуатації. У роботі запропоновано питання охорони праці та безпеки у надзвичайних ситуаціях.

Ключові слова: система підтримки прийняття рішень, прогнозування, машинне навчання, інформаційні ресурси, органічні відходи, економічна ефективність.

ЗМІСТ

ВСТУП

Сучасні умови розвитку громад та енергетичного сектору України вимагають пошуку нових шляхів забезпечення енергетичної безпеки та зменшення залежності від традиційних джерел палива. Особливої актуальності набуває використання біоенергії, що виробляється з органічних відходів, оскільки вона поєднує в собі два важливих аспекти – утилізацію відходів та створення альтернативного джерела енергії [17]. Розвиток біоенергетики відповідає як цілям сталого розвитку, так і завданням підвищення енергетичної автономності місцевих громад [38].

Попри значний потенціал відновлюваних джерел, процес планування виробництва біоенергії залишається складним завданням, що потребує врахування великої кількості параметрів: обсягів та складу відходів, можливостей їх логістики, вибору технологій переробки, прогнозу попиту на енергоресурси. У цих умовах доцільним є застосування систем підтримки прийняття рішень, які здатні поєднувати методи математичного моделювання, оптимізації та інформаційні технології [13]. Використання таких систем дозволяє підвищити ефективність управління біоенергетичними проєктами, мінімізувати витрати та забезпечити функціонування енергетичної інфраструктури.

Об’єкт дослідження – процес планування виробництва біоенергії з органічних відходів.

Предмет дослідження – методи та інформаційні технології підтримки прийняття рішень для підвищення ефективності планування у біоенергетиці.

Засоби дослідження – системи підтримки прийняття рішень, методи математичного та імітаційного моделювання, інструментарій обчислювального інтелекту та геоінформаційних технологій.

Таким чином, актуальність теми зумовлена необхідністю створення інтелектуальних інструментів, які забезпечать ефективне використання органічних відходів як ресурсу для виробництва біоенергії та сприятимуть досягненню енергетичної незалежності країни.

РОЗДІЛ 1.
АНАЛІЗ СТАНУ ПЛАНУВАННЯ ВИРОБНИЦТВА БІОЕНЕРГІЇ ТА ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1. Сучасний стан і тенденції розвитку біоенергетики в контексті цифровізації

Біоенергетика є однією з найбільш динамічних галузей відновлюваної енергетики, яка забезпечує подвійний ефект: виробництво енергії з відновлюваних джерел та зменшення негативного впливу відходів на довкілля. За даними Міжнародного енергетичного агентства (IEA), частка біоенергії у світовому балансі відновлюваних джерел становить понад 50%, що робить її найбільш поширеним видом відновлюваної енергії [1]. В Україні цей сектор має особливе значення у контексті переходу до енергетичної незалежності та імплементації Європейського зеленого курсу [18].

У сучасних умовах цифровізація стала невід'ємним чинником розвитку біоенергетики. Використання інформаційних систем, великих даних, інтернету речей (IoT) та геоінформаційних технологій дозволяє не лише планувати виробничі процеси, але й здійснювати моніторинг та прогнозування енергетичних потоків у реальному часі [9]. Це дає змогу значно підвищити ефективність управління виробництвом біоенергії, оптимізувати витрати та інтегрувати енергетичні установки у розумні мережі (smart grids).

Важливим напрямом розвитку є інтеграція математичних моделей та алгоритмів штучного інтелекту у процеси планування. Наприклад, задача оптимального розподілу органічних відходів для виробництва біогазу може бути формалізована як задача мінімізації витрат при забезпеченні необхідного енергетичного потенціалу:

$$\min Z = \sum_{i=1}^{n} c_i x_i \quad \textit{за умови} \quad \sum_{i=1}^{n} e_i x_i \geq E_{min}, \qquad (1.1)$$

де c_i – витрати на використання відходів i-го типу; x_i – кількість відходів, що використовується; e_i – енергетичний вихід від одиниці відходів; E_{min} – мінімальний рівень енергетичної продуктивності.

Такий підхід дає можливість врахувати не лише кількісні характеристики сировини, а й якісні параметри, що є важливим при цифровому моделюванні сценаріїв [12].

Цифрові технології активно застосовуються для створення платформ управління біоенергетичними підприємствами. Вони забезпечують збирання даних із сенсорів, що контролюють вологість, температуру, хімічний склад сировини та інші параметри. Це дозволяє створювати так звані «цифрові двійники» біоенергетичних установок, які відтворюють роботу системи в режимі реального часу та дають змогу прогнозувати її ефективність [35].

Для наочності у таблиці 1.1 наведено порівняння традиційних та цифрових підходів до управління біоенергетикою.

Таблиця 1.1 – Порівняння традиційних та цифрових методів управління біоенергетичними системами

Характеристика	Традиційний підхід	Цифровий підхід
Збір даних	Ручний, періодичний	Автоматизований, безперервний
Аналіз процесів	Ретроспективний	Прогностичний (AI, ML)
Контроль	Локальний	Дистанційний, інтегрований
Ефективність	Залежить від персоналу	Оптимізується алгоритмами

Окрім управління виробництвом, цифрові технології відкривають нові можливості для інтеграції біоенергетики в енергетичні ринки. Використання блокчейн-рішень дозволяє здійснювати облік і торгівлю біоенергією на децентралізованих платформах, що підвищує прозорість та довіру до системи [8].

На рисунку 1.1 зображено спрощену схему інтеграції цифрових технологій у біоенергетичний комплекс.

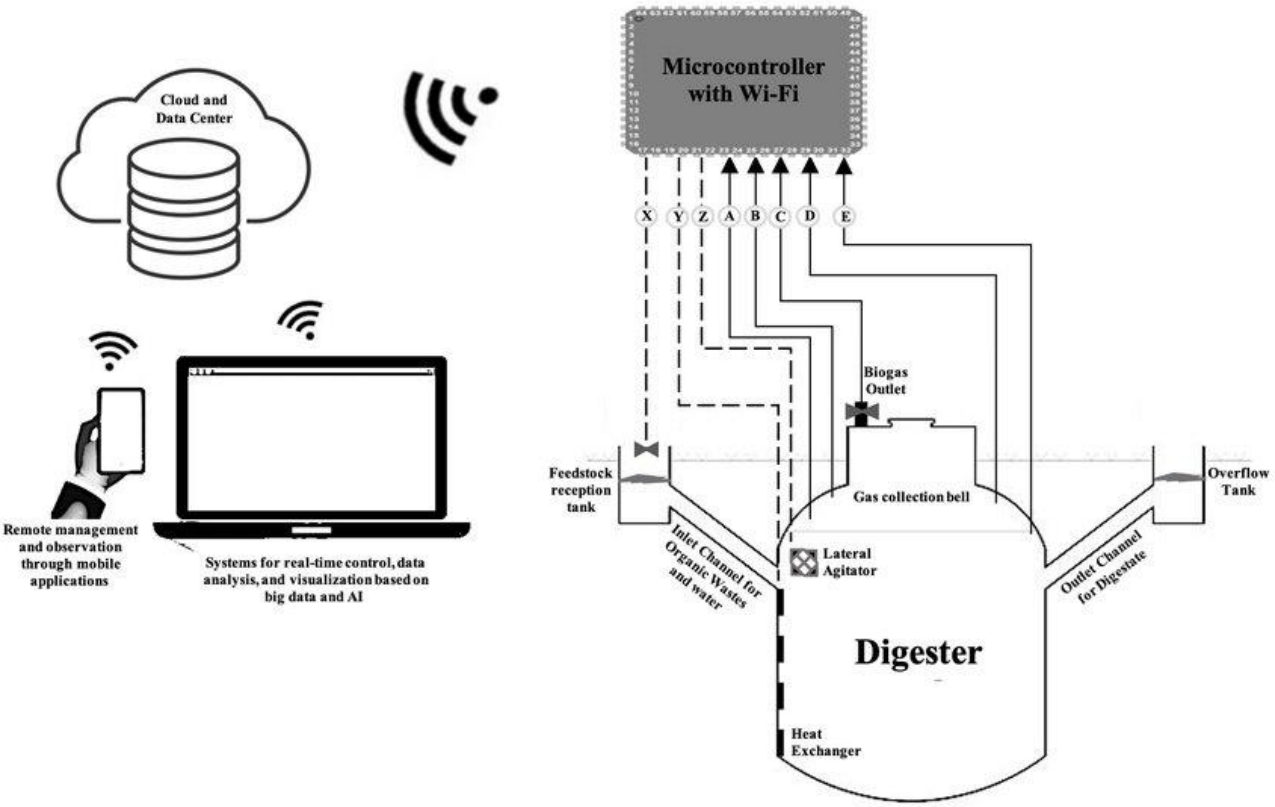


Рисунок 1.1 – Інтеграція цифрових технологій у біоенергетичну систему [32]

Таким чином, сучасний стан розвитку біоенергетики характеризується не лише нарощенням потужностей, але й глибокою інтеграцією інформаційних технологій, що забезпечують ефективність та гнучкість енергетичних систем. Цифровізація дозволяє перейти від статичного планування до адаптивного управління, що враховує зміни у сировинній базі, попиті на енергію та екологічних умовах. У цьому контексті інформаційні системи виступають ключовим інструментом, без якого сталий розвиток біоенергетики є практично неможливим.

1.2. Інформаційні ресурси та дані про органічні відходи як базис для створення систем підтримки рішень

Ефективність систем підтримки прийняття рішень (СППР) у сфері біоенергетики суттєво залежить від якості, повноти та своєчасності даних, які використовуються як інформаційний фундамент. Органічні відходи – це не лише

біологічний ресурс, але й джерело великої кількості інформації: обсяги, склад, вологість, вуглець/азотний баланс, географічне розташування джерел, режим накопичення тощо. Без чітко структурованої бази даних, інтегрованих даних картографії та динамічних метрик неможливо реалізувати адекватні алгоритми прогнозування та оптимізації.

1.2.1. Характеристика інформаційних ресурсів

Інформаційні ресурси, що стосуються органічних відходів, можна поділити на кілька категорій.

Перша – статистичні бази даних. Це регіональні чи національні звіти про кількість сільськогосподарських, комунальних чи промислових відходів, їх склад та напрямки утилізації. Наприклад, портал BioEnergy KDF містить дані про біомасу з потоків відходів, зокрема щодо кількостей твердої біомаси, рідких фракцій і їх потенційного енергетичного вмісту [44].

Друга категорія – геопросторові дані. Карти й шари GIS, які відображають розподіл джерел відходів у просторі, транспортні вузли, шляхи доставки та оптимальні локації для установок. Наприклад, Національна лабораторія відновлюваної енергії США (NREL) пропонує геопросторові інструменти та карти ресурсів біомаси, які інтегруються в аналіз доступності сировини [29].

Друга категорія – реальні вимірювання з місця (сенсорні дані). Показники вологості, температури, хімічного складу, швидкості деградації, обсягу газів тощо. Важливою складовою є часові ряди, які показують зміну характеристик відходів у часі під впливом погодних умов або режимів накопичення.

Четверта категорія – дані зовнішніх джерел. Кліматичні записи, агрометеорологічні моделі, демографічні дані, дані про споживання енергії тощо, які дозволяють накладати відходи на контекстні зміни середовища.

Сукупність цих ресурсів утворює інформаційне середовище, в якому працює СППР – статистика дає історичний базис, GIS-карти задають просторову основу,

сенсори забезпечують оперативні дані, а зовнішні джерела – контекстні змінні, що коригують моделі прогнозування.

1.2.2. Формалізація даних

Щоб система могла працювати з цими ресурсами в аналітичному та прогнозному режимі, потрібно формалізувати дані у структурований вигляд. Часто використовується реляційна модель баз даних, де кожна сутність (наприклад, «джерело відходів», «проба», «вимірювання») представлена таблицею з наборами атрибутів. Наприклад, таблиця «Проба» може містити поля:

- id_проба;
- id_джерело;
- дата;
- вологість %;
- вміст C;
- вміст N;
- температура °C;
- енергетична_потужність.

Для багатовимірного аналізу доцільно створювати OLAP-куби, що дозволять агрегацію даних за часом, регіоном або типом сировини. Формула агрегування для обчислення середньої вологості для певного регіону R та часу t має вигляд:

$$\bar{w}(R,t) = \frac{1}{n}\sum_{i=1}^{n} w_i(R,t), \qquad (1.2)$$

де w_i – вологість i-ої органічної сировини.

При інтеграції геопросторових даних до бази використовують просторові бази даних (PostGIS, Spatialite тощо), в яких до атрибутів додається геометричний стовпець (точка, лінія або полігон). Таким чином, кожне джерело відходів має координати, що дозволяє відображати його на карті й виконувати просторові запити, наприклад: «всі джерела в радіусі 10 км від переробного пункту».

1.2.3. Виклики та проблеми якості даних

Незважаючи на наявність різноманітних джерел, дані про органічні відходи часто характеризуються низкою проблем: фрагментарністю, різною періодичністю оновлень, розбіжностями у визначеннях, пропущеними вимірами та невирівняною розмірністю (наприклад, щоденні сенсорні дані поруч із щорічними статистичними звітами). Наприклад, у Північній Америці різні штати по-різному інтерпретують, що є «органічними відходами», що ускладнює порівняння даних [40].

Ще одна проблема – синхронізація часів. Сенсорні дані можуть надходити в режимі хвилин, тоді як статистика – раз на рік. Потрібні алгоритми інтерполяції, агрегації чи інтерпроляції таких часових рядів. Також буває, що дані мають систематичні помилки (наприклад, датчик вологості дає відхилення), що вимагає попередньої обробки (очищення, нормалізація, виявлення викидів).

Крім того, деякі важливі джерела інформації можуть бути закритими або комерційними, що обмежує доступ до них. Наприклад, деякі компанії біоенергетики утримують власні дані як комерційну таємницю. У відкритих дослідженнях часто звертають увагу на відкриті бази і їхні обмеження [28].

1.2.4. Приклади існуючих інформаційних ресурсів та їх роль у СППР

В Європі і США існує декілька важливих баз та каталогів ресурсів біомаси й відходів, які можуть бути використані як джерела даних для СППР. Наприклад, у роботі Mukamwi та співавторів здійснено огляд європейських баз даних, що охоплюють біомаси, біорефінерії та їхні параметри (вид, хімічний склад, потенціал) [28]. Ці дані дозволяють порівнювати різні технології утилізації відходів, оцінювати ресурси в районах та будувати сценарії розгортання установок.

Іншим прикладом є BDWaste – відкритий датасет з Бангладешу, який класифікує відходи за двома категоріями (перетравлювані / неперетравлювані) і складається з великої кількості зображень відходів із зазначенням характеристик

[40]. Такий тип ресурсів особливо корисний для навчання алгоритмів машинного бачення, що можуть автоматично класифікувати типи відходів у потоках сортування.

У таблиці 1.2 наведено порівняння кількох прикладів інформаційних ресурсів, їхніх особливостей та обмежень.

Таблиця 1.2 – Приклади інформаційних ресурсів щодо органічних відходів

Назва ресурсу / бази	Обсяг / покриття	Типи даних	Переваги	Обмеження
BioEnergy KDF	Глобальний / США	Потоки біомаси, потенціал	Великий набір даних, вільний доступ	Обмежено географіями
U.S. Solid Biomass Resources by County	США, округи	Статистика по категоріях сировини	Деталізація по округах	Не охоплює інші країни
BDWaste	Бангладеш	Зображення + мітки	Підтримка ML / CV задач	Специфіка країни, обмеження контексту
Європейські біомасні каталоги	Європа	Хімічні складові, види сировини	Комплексні метадані	Часто не оновлюються регулярно

У сфері планування сировинної бази також важливі каталоги, як-от U.S. Solid Biomass Resources by County, які на рівні округів США зібрані статистичні дані про обсяги біомаси та відходів [29]. Ці дані можуть бути використані як орієнтирні карти потенціалу в певних регіонах при створенні моделей доступності сировини.

Крім того, у деяких проектах вже застосовують прогностичні аналітики та навчальні моделі для оцінки виробничого потенціалу відходів. Наприклад, у статті [7] досліджують, як різні алгоритми (нейронні мережі, SVM, дерева рішень тощо) здатні прогнозувати продуктивність процесів на основі вхідних даних про відходи.

1.2.5. Взаємозв’язок інформаційних ресурсів із СППР

У контексті СППР дані про органічні відходи виконують функцію інформаційної основи для трьох основних компонентів – набору сценаріїв, модулів прогнозування/оптимізації та інтерфейсів підтримки рішень.

Наприклад, просторові дані з GIS дозволяють модулю руйнувати сценарії доставки сировини та розміщення установок. Сенсорні показники, інтегровані в часі, дозволяють коригувати прогнози в режимі реального часу. Статистичні бази даних слугують для калібрування моделей.

Моделювання використовує такі вхідні вектори:

$$x = (V, w, C, N, d), \tag{1.3}$$

де V – обсяг відходів; w – вологість; C, N – концентрації вуглецю й азоту; d – відстань від установки.

Тоді прогноз енергетичного виходу y становитиме:

$$y = f(x;\theta), \tag{1.4}$$

де f – модель (наприклад, нейронна мережева або регресійна); θ – вектор параметрів.

Отриманий прогноз служить як вхід до модуля оптимізації, який мінімізує витрати на транспортування або максимізує ефективність установки.

Підсумовуючи, можна сказати, що без добрих, комплексних і інтегрованих інформаційних ресурсів система підтримки прийняття рішень в біоенергетиці не може бути повноцінною. Дані – це паливо для аналітики й інтелектуальних рішень. У наступному розділі ми розглянемо, як на практиці моделюються й інтегруються алгоритми прогнозування/оптимізації в таких системах.

1.3. Методологія планування виробництва біоенергії в інформаційних системах та її задачі

Планування виробництва біоенергії є складним багаторівневим процесом, що охоплює прогнозування наявних ресурсів, визначення оптимальної технології їх використання, побудову сценаріїв виробничих процесів та інтеграцію з енергетичними мережами. У контексті інформаційних систем планування передбачає створення математичних моделей, алгоритмів оптимізації та програмного забезпечення, здатного обробляти великі обсяги даних і підтримувати прийняття рішень у реальному часі [38].

Сучасна методологія ґрунтується на поєднанні кількісних методів математичного програмування, статистичних прогнозів, моделей машинного навчання та геоінформаційних технологій. Ключовим є інтеграційний підхід, за яким різнорідні дані (статистика відходів, сенсорні дані, карти транспортних мереж, кліматичні сценарії) об’єднуються в єдину платформу. Це дозволяє планувати виробництво біоенергії як з точки зору технічної доцільності, так і з позицій економічної та екологічної ефективності [8].

Серед задач, що вирішуються в межах інформаційних систем планування біоенергії, можна виділити чотири ключові групи: прогнозування, оптимізація, моніторинг та інтеграція.

Прогнозування охоплює оцінку потенціалу відходів, динаміки їх накопичення та майбутніх потреб у біоенергії. Для цього використовуються моделі часових рядів та алгоритми машинного навчання. Наприклад, прогноз обсягів сировини можна описати моделлю авторегресії:

$$Q_t = \alpha + \sum_{i=1}^{p} \beta_i Q_{t-i} + \varepsilon_t, \qquad (1.5)$$

де Q_t – обсяг органічних відходів у момент часу t; β_i – коефіцієнти моделі; ε_t – похибка.

Оптимізація включає визначення найкращого розміщення біоенергетичних установок, мінімізацію витрат на транспортування сировини, максимізацію

енергетичного виходу та скорочення викидів CO_2. Типова задача формулюється як задача лінійного програмування:

$$max\, E = \sum_{i=1}^{n} e_i x_i \quad за\ умови \quad \sum_{i=1}^{n} c_i x_i \le C_{max}, \tag{1.6}$$

де e_i – енергетичний вихід від сировини *i*-го типу; x_i –використана кількість; c_i – витрати на транспортування і переробку; C_{max} – максимально допустимі витрати.

Моніторинг забезпечує контроль за фактичним виконанням планів у реальному часі. Він реалізується завдяки датчикам IoT, що відстежують параметри процесу: температуру, тиск, склад біогазу, продуктивність установки. Дані інтегруються в інформаційну систему, де автоматично порівнюються з плановими показниками.

Інтеграція передбачає взаємодію з іншими енергетичними системами та ринками. Біоенергетичні об’єкти повинні узгоджуватись із роботою електричних мереж, враховувати попит на теплову енергію, а також підключатися до платформ торгівлі «зеленою» енергією [18].

Методологія планування передбачає модульну архітектуру СППР. Типова структура містить:

- ✓ модуль збору даних (сенсори, бази статистики, GIS-шари);
- ✓ модуль обробки та очищення даних;
- ✓ модуль прогнозування;
- ✓ модуль оптимізації;
- ✓ модуль візуалізації результатів.

На рисунку 1.2 зображено спрощену архітектурну схему інформаційної системи для планування біоенергії.

При побудові інформаційних систем планування важливо враховувати багатокритеріальний характер задач. Ефективність оцінюється за технічними, економічними та екологічними показниками. У таблиці 3 наведено приклад системи критеріїв.

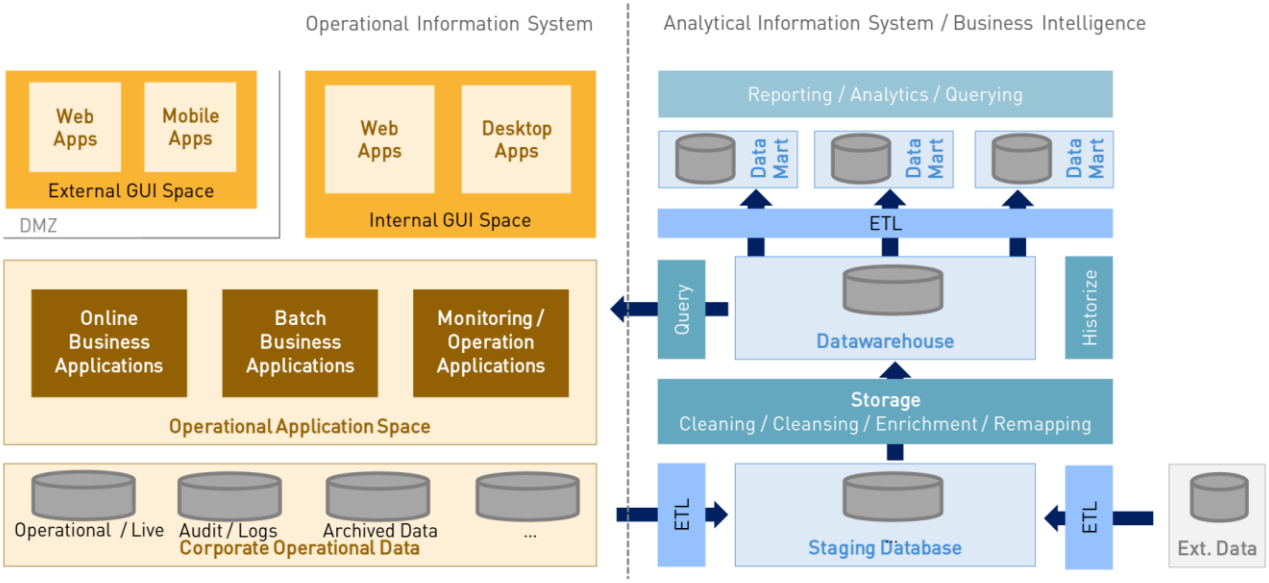


Рисунок 1.2 – Архітектура інформаційної системи планування виробництва біоенергії [27]

Таблиця 1.3 – Критерії оцінки ефективності планування виробництва біоенергії

Критерій	Одиниця виміру	Ціль
Енергетичний вихід	МВт·год	Максимізація
Собівартість виробництва	€/МВт·год	Мінімізація
Викиди CO_2	кг/МВт·год	Мінімізація
Час доставки сировини	хвилини	Мінімізація
Коефіцієнт використання відходів	%	Максимізація

Для комплексної оцінки використовується інтегральний показник, що враховує вагові коефіцієнти критеріїв:

$$I = \sum_{j=1}^{m} w_j \cdot k_j, \qquad (1.7)$$

де w_j – вага j-го критерію; k_j – нормалізоване значення j-го критерію.

Вибір ваг здійснюється експертами або визначається методами багатокритеріальної оптимізації (наприклад, AHP чи TOPSIS) [6].

Попри значний прогрес, методологія планування біоенергії в інформаційних системах стикається з низкою викликів. По-перше, це невизначеність даних:

відсутність точних вимірів, затримки в оновленні статистики, різні формати даних. По-друге, обчислювальна складність задач, що вимагає використання високопродуктивних обчислень та хмарних технологій. По-третє, потреба в інтеграції з ринками енергії для створення адаптивних сценаріїв роботи установок.

Перспективними напрямами є застосування цифрових двійників, що дозволяють моделювати роботу біоенергетичних комплексів у віртуальному середовищі, а також використання технологій штучного інтелекту для автоматичного пошуку оптимальних сценаріїв. Важливою є й стандартизація обміну даними між різними системами, що дозволить інтегрувати біоенергетичні об’єкти у загальнонаціональні та європейські енергетичні платформи [7].

1.4. Аналіз існуючих інформаційних систем і програмних платформ у сфері біоенергетики

Екосистема цифрових інструментів для біоенергетики за останнє десятиліття стала багатошаровою – від платформ для стратегічного енергетичного моделювання до вузькоспеціалізованих рішень для оцінки викидів, логістики біомаси, оперативного моніторингу та інтеграції з енергоринками. Нижче подано узагальнений огляд ключових відкритих і комерційних систем, що фактично застосовуються в наукових дослідженнях і практиці енергетичного планування.

Абстрагуючись від брендів, більшість рішень укладаються в чотири функціональні класи. Перший – енергосистемні моделі (PyPSA, MESSAGEix, TIMES, EnergyPLAN, Calliope), які формують сценарії розвитку та оптимізують структуру генерації [20; 15; 22; 23; 1; 34]. Другий – платформи політики/планування (LEAP, RETScreen, HOMER), орієнтовані на прикладні оцінки техніко-економічної доцільності проєктів [39; 30; 47]. Третій – інструменти життєвого циклу та вуглецевих розрахунків (GREET, openLCA, BioGrace), що дають прозорі метрики впливу [46; 19; 14]. Четвертий – спеціалізовані ланцюгово-логістичні та просторові оптимізатори біомаси (S2Biom Toolset, BeWhere) [36; 24; 31].

Таблиця 1.4 – Узагальнені можливості типових платформ

Платформа	Відкритість	Сектори	Оптимізація	Простір/час	GIS/біомаса
oemof	Open-source	Е/Т/П	LP/MILP	Гнучко	Через Python-бібліотеки [20]
PyPSA	Open-source	Е, сектор-куплінг	LP/MILP	Висока/висока	Пакети PyPSA-Eur, Geo [15]
MESSAGEix	Open-source ядро	Е/Т/П, IAM	LP/MIP	Сценарно	Імпорт з ixmp [22]
TIMES (ETSAP)	Вільно з ліцензією	Е/Т/П	LP/MIP	Сценарно	Імпорт з зовн. БД [23]
EnergyPLAN	Freeware	Е/Т/Тр/Хол	Евристика	Національний/годинний	Обмежено [1]
Calliope	Open-source	Е/Т/П	LP/MILP	Мультишкальність	Geo-сумісність [34]
LEAP	Free для LMIC/академ.	Е/Т/АП	Баланси/цільові	Будь-які	Імпорт даних [39]
RETScreen	Комерц. (free viewer)	Проєкти/клімат	Фін. аналіз	Проєктний	Клімат/ресурси [30]
HOMER Pro	Комерц.	Мікромережі	Стох./опт.	Хронометричний	Метео профілі [47]
GREET	Безкоштовно	LCA палив	LCI/LCA	–	Широкі БД [46]
openLCA	Open-source	LCA/CF/EF	LCI/LCA	–	Ecoinvent тощо [19]
BioGrace	Безкоштовно	GHG біопалив	Метод RED	–	Стандарти ЄС [14]
S2Biom Toolset	Free (FP7)	Біомаса	Картографія	Пан-ЄС/регіон	Дані supply-cost [36; 11]
BeWhere	Доступ за запитом	Логістика/розм.	MILP	Просторове	Транспорт/біомаса [24; 31]

Е/Т/П – електрика/тепло/паливо; АП – забруднення повітря; LP/MILP – лінійне/змішане цілочисельне програмування.

Для експлуатаційного рівня все частіше використовуються SCADA/IIoT-шари (AVEVA PI System, OPC UA, Ignition) і реєстри атрибутів енергії (Energy Web), що забезпечують зв'язність «дані-моделі-рішення» [11; 33; 25; 17; 10].

У системному плануванні важливі чотири властивості – підтримка багатосекторності (електрика-тепло-паливо), просторово-часова деталізація, наявність оптимізації та GIS/даних про біомасу. Зведемо їх у таблицю.

Oemof та Calliope пропонують модульну побудову енергосистемних моделей із явною постановкою оптимізаційних задач у Python. Вони зручні для прототипування ланцюгів «відхід-логістика-установка-мережа», коли потрібно швидко інтегрувати власні обмеження (наприклад, сезонність надходжень органіки або граничні вологість/C:N) [20; 34].

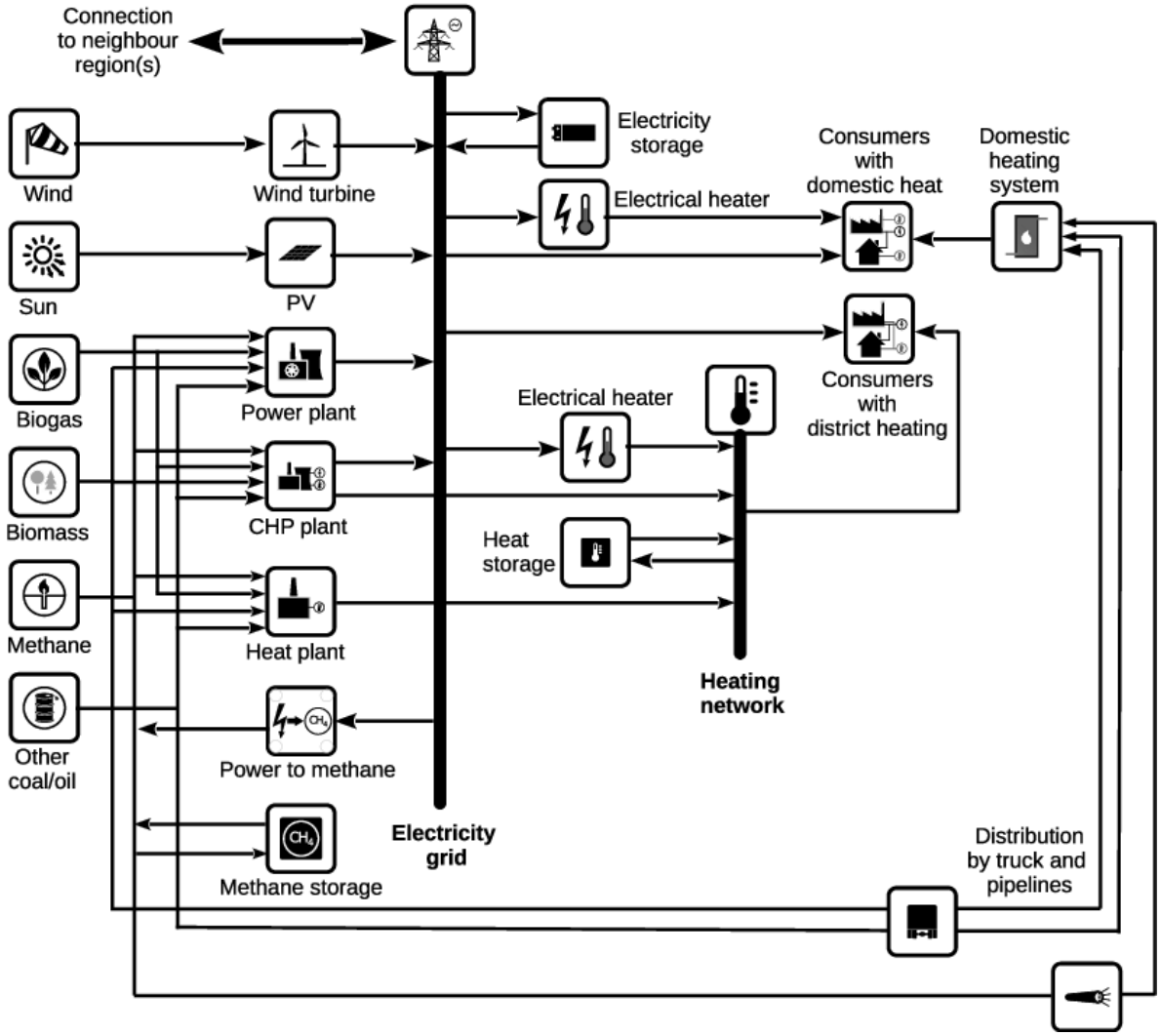


Рисунок 1.3 – Представлення складної енергетичної системи в рамках програми Oemof [21]

PyPSA історично сильна в електричних мережах і секторному куплінгу (електрика-водень-тепло); разом із наборами PyPSA-Eur/-Earth це дає готові топології для оцінки включення біоелектрогенерації у потоки потужності та резервів [15]. MESSAGEix та TIMES – класика для довгострокової декарбонізації, з прозорим формалізмом «витрати-випуск-обмеження», де біомаса конкурує за ресурс із іншими технологіями в оптимальних портфелях [22; 23]. EnergyPLAN застосовують для годинного балансування національних систем і тестування сценаріїв надвисокої частки ВДЕ з біогенерацією у зв'язці з теплом [1].

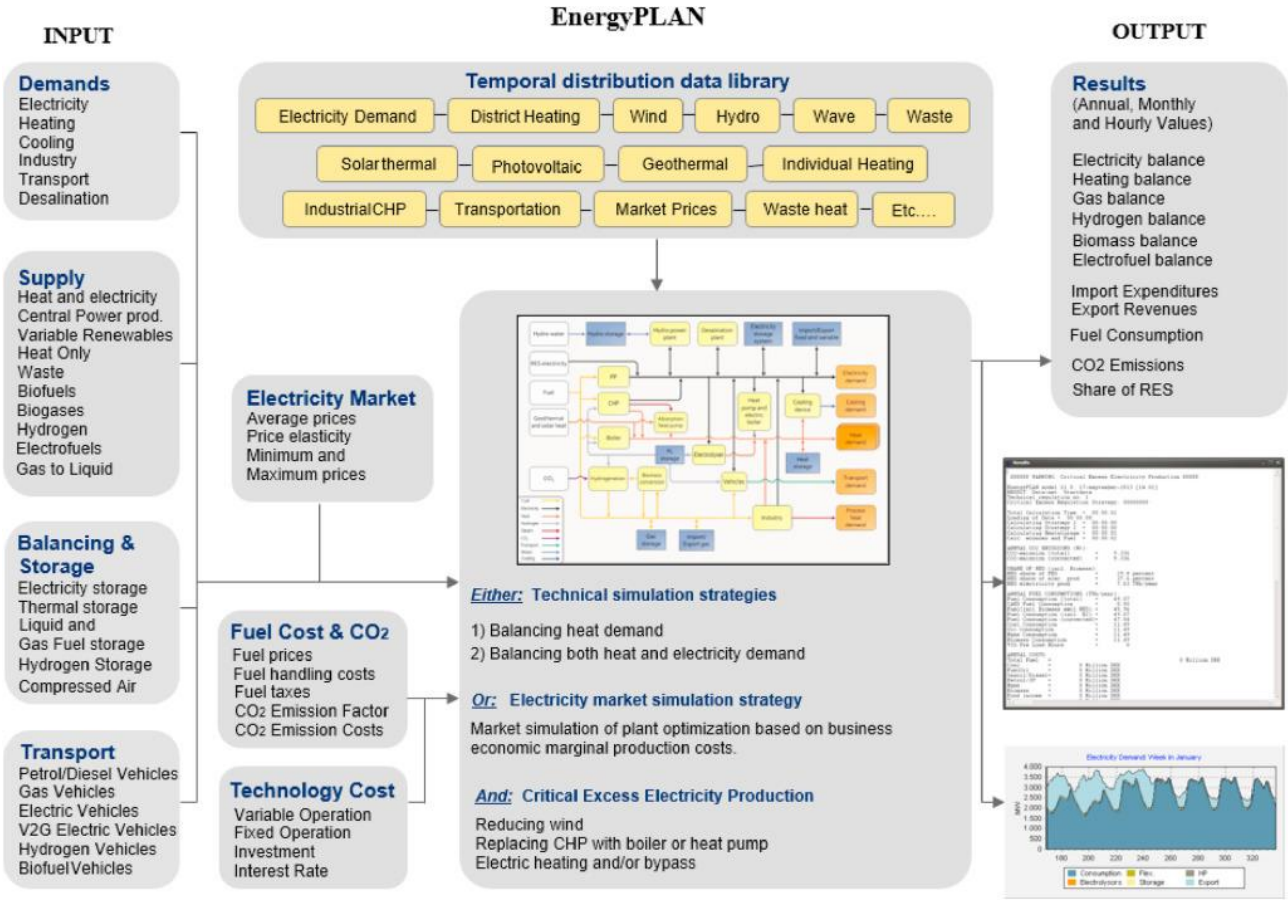


Рисунок 1.3 – Вхідні та вихідні дані EnergyPLAN [21]

На рівні проєктів і регіонів зручними є LEAP, RETScreen та HOMER. LEAP добре підходить для енергетичних балансів і кліматполітики, з простими модулями попиту та пропозиції [39]. RETScreen прискорює скринінг і M&V (measurement & verification) – корисно, коли потрібно порівняти біогазову ТЕЦ з альтернативами за LCOE/IRR на референтних кліматичних профілях [30]. HOMER є де-факто стандартом для мікромереж: моделює хронометрично, враховуючи варіації

навантаження/ресурсів і економіку; підходить для локальних біоТЕЦ або когенерацій на відходах [47].

Біоенергетика прямо потребує просторово-логістичної компоненти. S2Biom Toolset надає гармонізовані карти та криві «вартість-пропозиція» по Європі, включно з Україною, для вибору зон заготівлі/постачання [36]. BeWhere (IIASA) – змішано-цілочисельний оптимізатор просторового розміщення заводів і ланцюгів постачання, який мінімізує сукупні витрати з урахуванням транспортної мережі, якості біомаси і цільових продуктів (біоПП, біометан, біопаливо) [33].

Сучасні проєкти біоенергетики виходять за межі офлайн-планування. AVEVA PI System виступає «хребтом даних» для збору потоків із датчиків (газовий склад, тиск, температура ферментера, виробіток), збереження часових рядів і подальшої аналітики [11; 10].

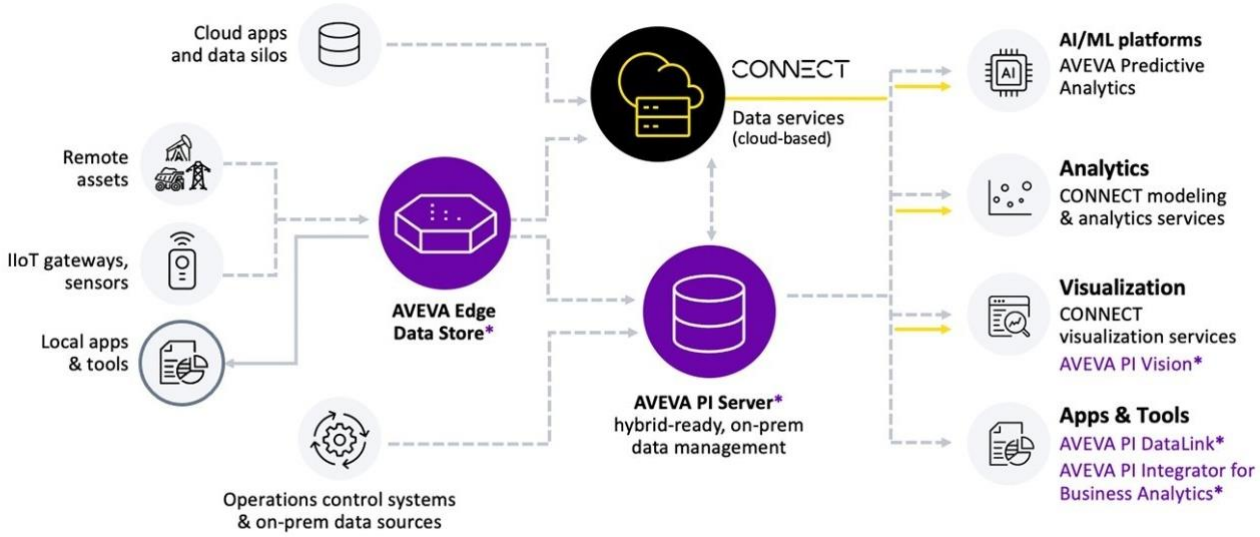


Рисунок 1.4 –Інтегрований портфель рішень AVEVA PI System

Стандарт OPC UA забезпечує інтероперабельність між обладнанням і програмними модулями, з профілями безпеки та модельованим адресним простором – це спрощує підключення біогазової установки до SCADA/DSS [20]. Платформа Ignition закриває потребу у візуалізації/алертах/MES-інтеграції із «безлімітним» ліцензуванням на теги/клієнти, що корисно для розподілених майданчиків [48].

На стороні прозорості походження енергії та атрибутів вуглецевих сертифікатів набирають обертів рішення Energy Web (Digital Spine/Origin SDK), які дозволяють реєструвати генерацію з біомаси, випускати гарантії походження та автоматизувати відповідність вимогам «зелених» ринків [16]. Для замовника це означає, що «цифровий слід» біоенергії може бути підтверджено верифікованими записами, а для оператора – інтеграцію з майбутніми гнучкими ринковими продуктами.

Відкриті фреймворки потребують кваліфікації в побудові моделей і верифікації даних; комерційні – підписок і адаптації під локальні норми. Просторове моделювання часто впирається в доступність якісних карт ресурсів за межами ЄС (де S2Biom має найбільше покриття). LCA-оцінки чутливі до гіпотез (транспортні відстані, вологість, заміщені продукти), тож інтеграція GREET/openLCA з реальними SCADA-даними підвищує достовірність. Питання кібербезпеки ОС-рішень (OPC UA, SCADA) і приватності даних при реєстрах атрибутів (Energy Web) мають бути враховані на етапі архітектурного дизайну.

1.5. Обґрунтування доцільності розробки системи підтримки прийняття рішень для планування виробництва біоенергії з органічних відходів

Сучасні житлові масиви стикаються з подвійною задачею – накопиченням органічних відходів та зростаючою потребою у відновлюваних джерелах енергії. Традиційне захоронення чи спалювання відходів призводить до екологічних ризиків, тоді як використання їх для виробництва біоенергії відкриває можливість створення замкнених локальних енергетичних систем. Проте впровадження таких проєктів потребує якісного планування, яке охоплює оцінку ресурсної бази, вибір технологій, розрахунок фінансових параметрів та врахування екологічних ефектів. У реальних умовах проектні

менеджери часто мають справу з фрагментарними даними, що ускладнює прийняття виважених рішень.

Розробка системи підтримки прийняття рішень «Organic Waste Energy Production System» орієнтована на вирішення цих задач. Система забезпечує інтеграцію інформації про обсяги органічних відходів, їхній енергетичний потенціал та прогнозні сценарії розвитку з економічними і екологічними показниками. Це дає змогу проектним менеджерам комплексно оцінювати доцільність встановлення модульних установок і обирати оптимальні варіанти їх впровадження.

СППР реалізована на Python відкриває широкі можливості для автоматизації обчислень, використання алгоритмів прогнозування і оптимізації, а також інтеграції з сучасними аналітичними інструментами. Це дозволяє створювати сценарії розвитку проекту залежно від змін у кількості відходів, вартості енергоресурсів чи політичних умов. Для менеджера це означає доступ до зрозумілого і надійного інструменту, який не лише формує показники, а й допомагає у прийнятті стратегічних рішень.

Функціональність системи охоплює кілька основних напрямів – прогнозування ресурсної бази, оцінка технічних характеристик модульних установок, аналіз економічної доцільності та моделювання екологічних ефектів. Користувач отримує не лише сухі числові результати, а й інтерпретацію у вигляді управлінських висновків, що дозволяють аргументовано презентувати проєкт інвесторам чи місцевим органам влади.

Особливу цінність система має для проєктів, що реалізуються у межах громад. Вона дозволяє розглядати локальні сценарії використання відходів житлових масивів, враховувати доступність територій для встановлення обладнання, логістичні маршрути та рівень підтримки з боку населення. Це створює основу для збалансованих рішень, де поєднуються екологічні вигоди, економічна рентабельність і соціальна підтримка.

Управлінська доцільність розробки полягає в тому, що система значно зменшує рівень невизначеності при плануванні. Менеджер отримує інструмент,

який дозволяє швидко порівнювати різні варіанти реалізації проєкту, бачити їхні наслідки у фінансовому та екологічному вимірах, а також прогнозувати довгострокову ефективність. Це особливо важливо в умовах обмеженого фінансування та високої конкуренції між проєктами, що претендують на інвестиції.

Отже, розробка «Organic Waste Energy Production System» є обґрунтованою і доцільною, оскільки вона орієнтована саме на практичні потреби проектних менеджерів. Вона допомагає трансформувати проблему органічних відходів житлових масивів у можливість створення сталих енергетичних систем, робить процес планування прозорим та обґрунтованим, а прийняті рішення – стратегічно зваженими.

РОЗДІЛ 2.
ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ СТВОРЕННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПЛАНУВАННЯ ВИРОБНИЦТВА БІОЕНЕРГІЇ З ОРГАНІЧНИХ ВІДХОДІВ

2.1. Концептуальна модель СППР для планування виробництва біоенергії з органічних відходів

Концептуальна модель системи підтримки прийняття рішень для планування виробництва біоенергії з органічних відходів у житлових масивах ґрунтується на ідеї керованого перетворення різнорідних даних на придатні для дії управлінські висновки. У найзагальнішому вигляді її логіка відтворює класичну тріаду «дані → моделі → рішення», що історично склалася у теорії СППР і пройшла багаторазову верифікацію в енергетичному менеджменті. Для задач біоенергетики вона доповнюється просторовими компонентами, елементами оцінювання життєвого циклу та сценарним аналізом під невизначеністю [8]. У цій роботі концептуальна модель реалізована у вигляді прототипу з графічним інтерфейсом, що веде користувача крок за кроком від введення вихідних характеристик житлового масиву до інтерпретації функціональних, економічних та екологічних метрик.

Перший рівень моделі формує інформаційний фундамент. Йдеться про структуровані дані щодо складу та динаміки органічних відходів домогосподарств, їх сезонну мінливість, щільність розміщення населення, потенційні точки збирання і транспортні плечі. Тут важливо не лише зібрати показники, а й забезпечити сумісність вимірювань, адже типові ринкові і муніципальні джерела різняться періодичністю та деталізацією [14].

Другий рівень (модельний) поєднує прогнозування ресурсної бази, масо-енергетичний баланс конверсійних технологій і оптимізаційні постановки розміщення модульних установок.

Третій рівень (презентаційно-аналітичний) перетворює результати обчислень на зрозумілі керівникові висновки: сценарні звіти, попередження про ризики, рекомендації щодо параметрів запуску чи масштабування.

Дані до системи надходять із трьох основних потоків. Перший – статистично-адміністративний, який надає агреговані показники утворення харчових, садових та змішаних органічних фракцій у типових домогосподарствах, а також норми відбору, що залежать від типу забудови та рівня доходів населення.

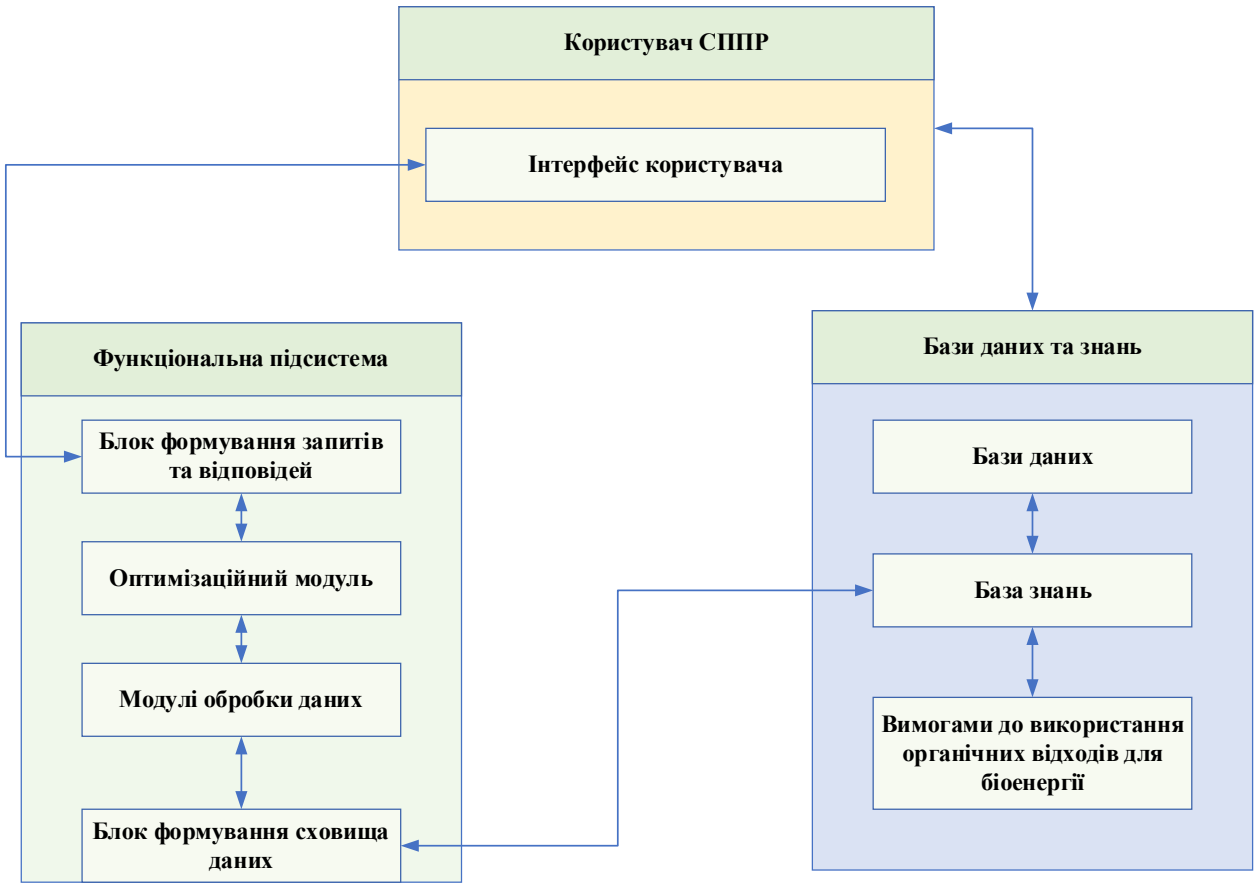


Рисунок 2.1 – Концептуальна схема СППР для планування виробництва біоенергії з органічних відходів

Другий – сенсорний або операційний, що має включати записи з вагових комплексів, GPS треків маршрутів і, за наявності, лабораторні показники вологи та співвідношення C:N для уточнення біохімічного потенціалу.

Таблиця 2.1 – Логіка відповідності завдань та модулів концептуальної моделі

Завдання користувача	Модуль концептуальної моделі	Основний результат
Сформулювати запит та отримати відповідь	Блок формування запитів та відповідей	Інтерпретований запит і релевантна відповідь
Оптимізувати розподіл ресурсів і технологій	Оптимізаційний модуль	Раціональний варіант використання сировини й потужностей
Перетворити дані на структуровану інформацію	Модулі обробки даних	Узагальнені показники для прийняття рішень
Забезпечити накопичення та доступ до інформації	Блок формування сховища даних	Актуалізована база для розрахунків та сценаріїв
Використати нормативні документи та стандарти	Вимоги до використання органічних відходів для біоенергії	Узгодженість рішень із законодавчими та галузевими вимогами
Отримати довідкову інформацію для проєкту	Бази даних та база знань	Каталоги, параметри й аналітичні матеріали для СППР

Третій – контекстний: метеорологія, ціни на енергоресурси, регуляторні параметри стимулювання. Потоки синхронізуються у часовому вимірі та геокодуються, аби надалі коректно працювали модулі прогнозування і просторового аналізу.

Усередині системи дані зберігаються у вигляді зв’язних сутностей «джерело відходів – проба – маршрут – установка», що полегшує побудову запитів та формування сценарних вибірок. Для динаміки використовуються часові ряди із зазначенням сезонних коефіцієнтів. Це дозволяє не втрачати

локальні особливості житлових масивів і уникає середніх, які часто маскують піки та провали завантаження.

Реалізація моделі у вигляді покрокового інтерфейсу «Initial Data → Organic Waste Forecast → Functional Metrics → Cost Metrics» дисциплінує процес аналізу і знижує вхідний бар’єр для менеджера. Кожен екран відповідає частині концептуальної схеми: введення профілю житлового масиву і джерел органіки; побудова місячних прогнозів з урахуванням сезонності; оцінка технологічних і енергетичних показників; формування економічних висновків. Важливо, що результати супроводжуються короткими поясненнями – не лише числовими значеннями, а і «історією», чому сценарій кращий або гірший.

Нами представлена логіка відповідності завдань та модулів концептуальної моделі у таблиці 2.1. Вона відображає логіку відповідності завдань користувача та модулів концептуальної моделі СППР для планування виробництва біоенергії з органічних відходів.

Подана таблиця 2.1 відображає, що кожне завдання користувача системи підтримки прийняття рішень має відповідний модуль у концептуальній моделі. Модулі працюють у взаємозв’язку. Дані збираються та обробляються, формуються запити, виконується оптимізація, а результати звіряються з базами знань і нормативними вимогами. Така логіка забезпечує не лише технічну, а й регуляторну узгодженість у плануванні виробництва біоенергії.

2.2. Опис моделі планування виробництва біоенергії з органічних відходів

Запропонована модель планування виробництва біоенергії з органічних відходів ґрунтується на системному поєднанні демографічних, технологічних, просторових та економічних чинників. Сучасні житлові масиви продукують значні обсяги органічних відходів, які при правильному управлінні здатні перетворюватися на відновлювану енергію. Водночас планування таких

процесів не може базуватися на статичних оцінках, оскільки характер і обсяги сировини коливаються у часі, а вибір технології залежить від просторових та економічних обмежень. Саме тому модель планування потребує багаторівневої структури, де прогнозування ресурсної бази поєднується з масо-енергетичним балансом, просторовою оптимізацією та багатокритеріальним прийняттям рішень.

Вихідною точкою у побудові будь-якої біоенергетичної моделі є оцінка динаміки утворення органічних відходів. Для цього застосовують комбіновані регресійно-сезонні моделі, які пов’язують кількість домогосподарств, рівень доходу населення та житлові характеристики з фактичними обсягами відходів. Для місячного прогнозу адитивна модель набуває вигляду:

$$\hat{Q}_m = \alpha + \beta_1 H + \beta_2 I + \beta_3 A + S_m, \qquad (2.1)$$

де $\hat{Q}_m$ – прогнозований обсяг відходів у m-му місяці; H – кількість домогосподарств; I – проксі змінної доходу; A – середня площа житла або щільність населення; S_m – сезонна складова.

Такий підхід дозволяє врахувати, що у літні місяці зростає частка садових і харчових відходів, тоді як узимку їх обсяги знижуються. Для оцінки енергетичного потенціалу використовується коефіцієнтний підхід, який враховує вологість та частку летких твердих речовин. Таким чином, біометановий вихід визначається як добуток прогнозованої маси доступної органічної фракції на питомий вихід біогазу та теплотворну здатність метану [43].

Після прогнозування ресурсної бази постає завдання визначити ефективність її конверсії у біоенергію. Модель планування ґрунтується на законі збереження маси і енергії, що забезпечує реалістичність сценаріїв. Для технологічного ланцюга «збирання – підготовка – анаеробне зброджування – очищення – енергетичне використання» масо-енергетичний баланс описується формулами:

$$M_{in} = M_{волога} + M_{суха} = M_{вих\ відх} + M_{вих\ біогаз}, \qquad (2.2)$$

$$E_{out} = \eta_{ел} \cdot E_{біогаз} + \eta_{тепло} \cdot E_{біогаз}, \quad (2.3)$$

де M_{in} – маса сировини на вході; $M_{вих\ біогах}$ – вихід газової суміші; $\eta_{ел}$, $\eta_{тепло}$ – коефіцієнти корисної дії електричної та теплової генерації.

Навіть у спрощеному вигляді баланс дисциплінує модель, обмежуючи її фізично обґрунтованими інтервалами. Практичні значення параметрів можна брати з технічних довідників по біомасі [42].

Таблиця 2.2 – Відповідність ресурсів та виходу енергії

Джерело відходів	Середній вихід біогазу, м³/т	Вологість, %	ККД електричної конверсії, %	ККД теплової конверсії, %
Харчові відходи	90–120	70–80	30–38	40–50
Садові відходи	60–80	50–60	28–35	35–45
Змішані відходи	70–100	60–75	29–37	38–48

У міському та приміському середовищі важливо враховувати просторові чинники. Система формує граф транспортної мережі з вершинами-джерелами відходів та потенційними пунктами встановлення модульних установок. Початкове відсіювання локацій здійснюється за санітарними зонами, доступом до доріг і теплових споживачів. Оптимізаційна постановка мінімізації витрат транспортування подається як:

$$min \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{F}} c_{ij} x_{ij}, \quad за\ умов\ \sum_{j} x_{ij} = Q_i, \sum_{i} x_{ij} \le C_j, \quad (2.4)$$

де Q_i – ресурс у i-му джерелі; C_j – пропускна здатність j-ї установки; c_{ij} – вартість або час транспортування.

Цей підхід відповідає класичній задачі лінійного програмування і може бути реалізований засобами відкритого програмного забезпечення [35].

Управлінські рішення щодо впровадження біоенергетичних технологій потребують врахування не лише енергетичного виходу, а й економічних та

соціально-екологічних факторів. Для цього застосовується багатокритеріальний підхід із ваговими коефіцієнтами, що дозволяє збалансувати різні критерії:

$$U_s = \sum_{k=1}^{m} w_k \, z_{s,k}, \qquad (2.5)$$

де $z_{s,k}$ – нормалізоване значення k-го критерію у s-му сценарії; w_k – вага k-го критерію у s-му сценарії.

Застосування методів аналізу ієрархій (AHP) або TOPSIS дозволяє зробити вибір прозорим і відтворюваним.

Щоб уникнути нереалістичних сценаріїв, у модель вбудовано елементи валідації та чутливісного аналізу. Прогнози ресурсів зіставляються з фактичними даними муніципальних служб. Масо-енергетичні розрахунки перевіряються на відповідність інтервалам, наведеним у довідниках. Чутливісний аналіз проводиться щодо ключових параметрів: ціни на енергоносії, відсотка населення, яке бере участь у роздільному зборі, та сезонних коливань. Це дозволяє формувати резервні сценарії та робити рішення більш стійкими до невизначеностей.

Запропонована модель планування виробництва біоенергії з органічних відходів поєднує кілька взаємопов’язаних модулів – прогнозування ресурсної бази, масо-енергетичний баланс, просторову оптимізацію та багатокритеріальне оцінювання. Такий підхід дозволяє перетворити первинні дані у практичні управлінські висновки. Використання формальних методів забезпечує прозорість та відтворюваність, а валідація і чутливісний аналіз підвищують надійність рішень. У результаті модель стає ефективним інструментом для проектних менеджерів, які планують впровадження модульних біоенергетичних систем у громадах.

2.3. Вибір засобів для створення СППР планування виробництва біоенергії з органічних відходів

Розробка системи підтримки прийняття рішень у сфері біоенергетики вимагає не лише побудови концептуальної моделі, але й практичного підбору програмних засобів, які забезпечують ефективність і доступність системи. При виборі технологічного стеку враховуються вимоги до обробки статистичних та просторових даних, потреба в інтерактивному інтерфейсі для проектних менеджерів, а також можливість подальшого масштабування моделі.

У створеному прототипі СППР використано мову Python, що стало основою для інтеграції всіх підсистем. Python було обрано через універсальність і наявність потужних бібліотек для аналізу даних, оптимізації та візуалізації. Реалізоване графічне середовище базується на бібліотеці Tkinter, яка дозволяє створювати простий і зрозумілий інтерфейс користувача з вкладками для введення вхідних параметрів, прогнозування ресурсів, відображення функціональних і економічних метрик. Це особливо важливо для менеджерів, які працюють з проектами встановлення модульних біоенергетичних установок і не мають спеціальних знань у програмуванні.

Блок обробки даних реалізовано із застосуванням Pandas для роботи з табличними структурами та NumPy для чисельних обчислень. Це дало змогу легко організувати введення та збереження результатів користувача у вигляді структурованих масивів даних, які згодом можна експортувати в таблиці для звітності. Для побудови графіків і відображення динаміки прогнозів застосовано Matplotlib, що інтегрується безпосередньо в інтерфейс Tkinter і дозволяє наочно візуалізувати результати розрахунків.

Окрему увагу приділено модулю прогнозування. У прототипі реалізовані регресійні рівняння з можливістю врахування сезонних коливань, що відповідає вимогам концептуальної моделі. Для подальшого розвитку передбачено можливість використання бібліотек Scikit-learn або Statsmodels для побудови більш складних моделей, таких як ARIMAX чи градієнтний бустинг. Це

дозволить забезпечити більш високу точність прогнозів при збільшенні обсягів історичних даних.

Оптимізаційний модуль у прототипі поки реалізований у спрощеному вигляді, однак його можна інтегрувати з бібліотеками PuLP чи Pyomo для вирішення транспортних задач і задач розміщення виробничих потужностей. Це надасть можливість проектним менеджерам оцінювати різні сценарії за критеріями мінімізації витрат або максимізації енергетичного виходу.

Управління даними в прототипі організовано через внутрішні сховища Python (словники та файли CSV). У перспективі система може бути інтегрована з повноцінними базами даних, такими як PostgreSQL з розширенням PostGIS, що дасть змогу зберігати просторову інформацію про джерела органічних відходів і потенційні місця розташування біоенергетичних установок.

Таблиця 2.3 – Використані засоби у прототипі СППР

Компонент СППР	Використані засоби	Очікуваний результат
Інтерфейс користувача	Tkinter	Інтерактивні вкладки для введення даних і відображення результатів
Прогнозування	Python, NumPy, Pandas	Розрахунок обсягів органічних відходів з урахуванням сезонності
Візуалізація	Matplotlib	Графіки прогнозів, динаміка показників, відображення результатів
Обробка даних	Pandas, CSV-файли	Збереження та аналіз введених параметрів і результатів
Оптимізація	Python (з перспективою PuLP, Pyomo)	Сценарне моделювання варіантів планування
Управління знаннями	Вбудовані модулі Python (словники, списки)	Накопичення даних для подальшого аналізу

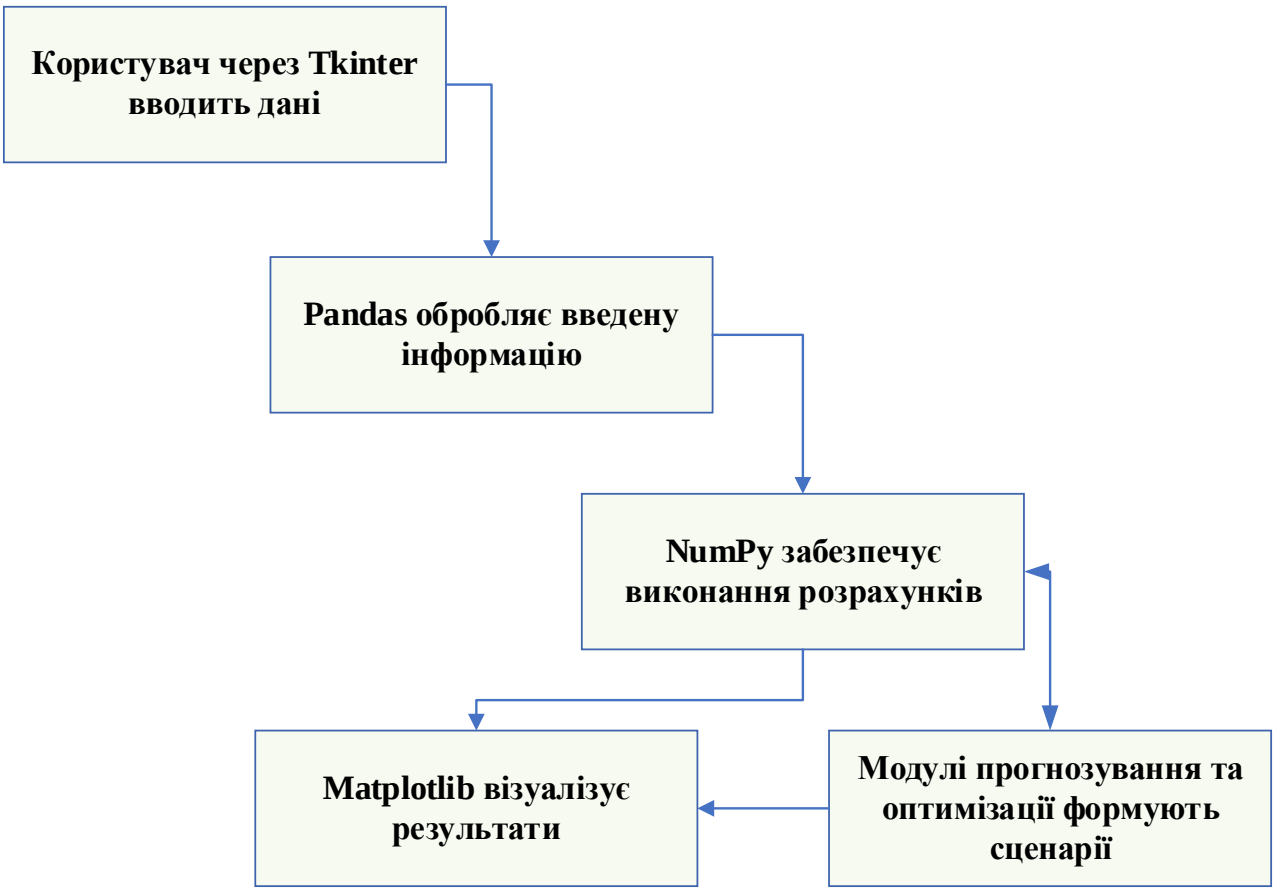


Рисунок 2.2 – Логіка роботи прототипу СППР

Таким чином, вибір засобів для створення СППР базувався на поєднанні простоти реалізації та можливості подальшого розширення. Python і його бібліотеки дозволяють реалізувати всі етапи – від збору даних і прогнозування до оптимізації та візуалізації результатів. Використання Tkinter робить систему доступною для управлінців без спеціальної технічної підготовки, а можливість інтеграції з базами даних і просторовими модулями відкриває шлях до масштабування та професійного використання.

РОЗДІЛ 3.
ОБҐРУНТУВАННЯ РАЦІОНАЛЬНОЇ МОДЕЛІ ПРОГНОЗУВАННЯ ОБСЯГІВ УТВОРЕННЯ ВІДХОДІВ У ДОМОГОСПОДАРСТВАХ

3.1. Підготовка даних щодо утворення органічних відходів

Першим етапом побудови раціональної моделі прогнозування обсягів утворення органічних відходів у домогосподарствах є завантаження та попередня обробка даних. У нашому випадку вхідним файлом став сформований набір даних initial_dataset.csv, який містить 900, зібраних із статистичних даних ЛКП «Зелене місто» м. Львів.

Таблиця 3.1 – Фрагмент набору даних initial_dataset.csv

Розмірність датасету: (900, 7)

	Settlement Type	Income level of the population	Residential Area, m²	Number of households, units	Number of inhabitants, persons	Organic Waste Source	Average daily volumes of organic waste per inhabitant, kg/person
0	village	medium	28968	216	732	yard waste (YW)	0.293
1	city	low	484980	7346	27924	mixed organic waste (MOW)	0.409
2	village	low	10444	100	439	yard waste (YW)	0.323
3	city	medium	464733	6743	21313	mixed organic waste (MOW)	0.351
4	city	low	216182	3096	12249	food waste (FW)	0.425

Завантаження здійснювалося у середовищі Jupyter Notebook за допомогою бібліотеки Pandas, що дає змогу зручно працювати з табличними структурами.

На рисунку 3.1 подано фрагмент коду, що відповідає за імпорт бібліотек та завантаження даних із локального шляху.

```
DATA_PATH = r"F:\Dipl_2025\Program\initial_dataset.csv"
MODEL_DIR = r"F:\Dipl_2025\Program"
BEST_MODEL_PATH = os.path.join(MODEL_DIR, "waste_forecast_best.joblib")
REPORT_PATH = os.path.join(MODEL_DIR, "model_report.csv")

# --- 1. Завантаження даних ---
df = pd.read_csv(DATA_PATH, encoding="utf-8-sig")
print("Розмірність датасету:", df.shape)
display(df.head())
```

Рисунок 3.1 – Фрагмент коду для завантаження даних про обсяги утворення органічних відходів у домогосподарствах

У коді відразу виводиться розмірність датасету, що підтверджує його обсяг, та кілька перших рядків для візуальної перевірки коректності (табл. 3.1). Такий підхід дозволяє на початку роботи переконатися, що набір даних прочитано правильно і він має необхідну структуру.

Подальший етап включав легкий data cleaning, що передбачає перевірку наявності пропусків і відповідність типів змінних очікуваним форматам. Усі числові стовпці було примусово переведено у формат numeric, а цільову змінну перейменовано у більш компактну форму target_kg_per_person.

На Рисунку 3.2 зображено відповідний блок коду, а у консолі відображається структура таблиці та кількість пропусків.

```
# --- Валідація та легкий data cleaning ---
# Перевіримо пропуски та типи
display(df.info())
display(df.isna().sum())

# Перейменуємо ціль у зручну коротку назву
target_col = "Average daily volumes of organic waste per inhabitant, kg/person"
df = df.rename(columns={target_col: "target_kg_per_person"})

# Переконаємось, що числові стовпці дійсно числові
num_cols = [
    "Residential Area, m²",
    "Number of households, units",
    "Number of inhabitants, persons"
]
df[num_cols] = df[num_cols].apply(pd.to_numeric, errors="coerce")

# Декілька базових логічних перевірок
assert df["target_kg_per_person"].between(0.10, 0.60).all(), "Ціль повинна бути в межах [0.10; 0.60] кг/особу."
```

Рисунок 3.2 – Фрагмент коду для перевірки наявності пропусків і відповідності типів змінних очікуваним форматам

Результати показали відсутність порожніх значень, що свідчить про якісну генерацію вихідного датасету.

Окремо було проведено логічну перевірку діапазону цільової змінної. Усі значення мали знаходитися в межах від 0.10 до 0.60 кг/особу, що відповідає реалістичним показникам утворення органічних відходів у домогосподарствах. Здійснено перевірку із застосуванням оператора assert, який автоматично зупинив би виконання у випадку порушення умови.

```
Data columns (total 7 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   Settlement Type                                         900 non-null    object
 1   Income level of the population                          900 non-null    object
 2   Residential Area, m²                                    900 non-null    int64
 3   Number of households, units                             900 non-null    int64
 4   Number of inhabitants, persons                          900 non-null    int64
 5   Organic Waste Source                                    900 non-null    object
 6   Average daily volumes of organic waste per inhabitant, kg/person  900 non-null    float64
dtypes: float64(1), int64(3), object(3)
memory usage: 49.3+ KB
None
Settlement Type                                         0
Income level of the population                          0
Residential Area, m²                                    0
Number of households, units                             0
Number of inhabitants, persons                          0
Organic Waste Source                                    0
Average daily volumes of organic waste per inhabitant, kg/person    0
dtype: int64
```

Рисунок 3.3 – Результати перевірки наявності пропусків і відповідності типів змінних очікуваним форматам

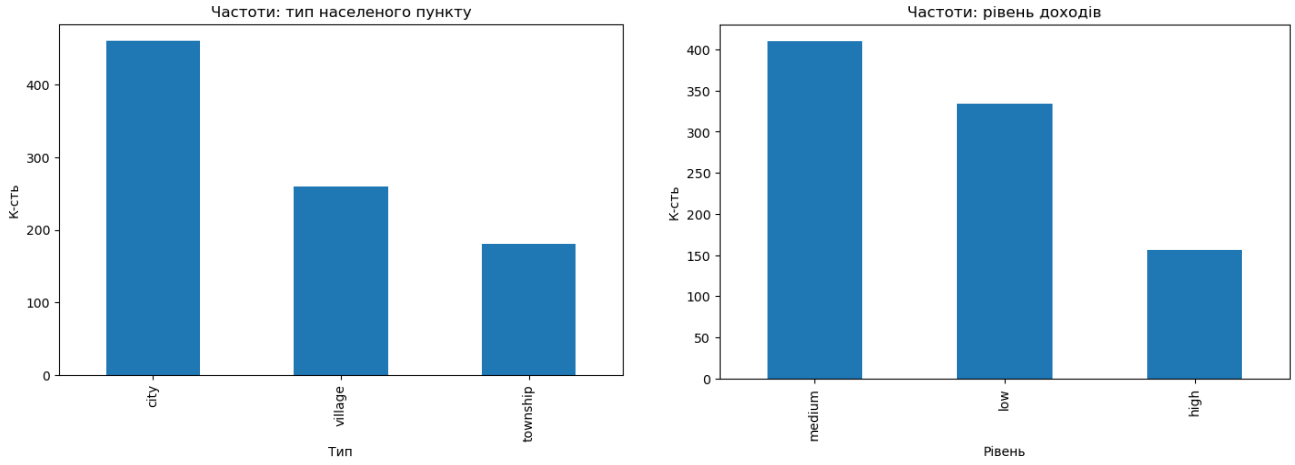
Після базового очищення було додано похідні ознаки. Зокрема, розраховувалася щільність населення через співвідношення кількості осіб до кількості домогосподарств та площа житлової забудови на одне домогосподарство. Ці індикатори мають безпосередній зміст для оцінки відходів: чим більша кількість осіб у середньому на одне домогосподарство, тим вищий прогнозований обсяг органіки; більша площа часто корелює з більшою кількістю зелених відходів.

На рисунку 3.4 показано відповідний код із розрахунком похідних фіч.

```python
# --- Інженерія ознак (похідні фічі, що мають сенс) ---
# щільність населення: особи на 1 домогосподарство
df["persons_per_household"] = df["Number of inhabitants, persons"] / df["Number of households, units"]

# площа на одне домогосподарство
df["area_per_household"] = df["Residential Area, m²"] / df["Number of households, units"]

# бінарні індикатори для типів поселення та рівня доходів (але закодуємо через OneHotEncoder у пайплайні)
# збережемо імена колонок
cat_cols = [
    "Settlement Type",
    "Income level of the population",
    "Organic Waste Source"
]

num_cols_ext = num_cols + ["persons_per_household", "area_per_household"]
```

Рисунок 3.4 – Фрагмент коду із розрахунком похідних фіч

Для категоріальних ознак («Settlement Type», «Income level of the population», «Organic Waste Source») було передбачено подальше кодування за допомогою OneHotEncoder. Проте на етапі підготовки важливо було перевірити їхню збалансованість. На рисунку 3.5, а зображено діаграму частот типів населених пунктів.



а) б)

Рисунок 3.5 – Діаграми частот типів населених пунктів (а) та рівнів доходів населення (б)

Отримана діаграма демонструє, що найбільшу частку займають міста, де очікувано генерується більше харчових відходів. На рисунку 3.5, б наведено частоти рівнів доходів населення, що підтверджує перевагу категорії «medium».

Важливим індикатором є також розподіл цільової змінної. На Рисунку 3.6 представлено гістограму середніх щоденних обсягів відходів на особу. Вона має нормоподібний характер, основна маса значень зосереджена в інтервалі від 0.25 до 0.35 кг/особу. Це свідчить про якісно сформований набір даних і його придатність для моделювання.

Додатково було виконано побудову точкового графіка, що відображає залежність між кількістю осіб на домогосподарство та обсягом відходів на людину (рис. 3.7). Графік демонструє пряму кореляцію: зі зростанням кількості мешканців середні значення цільової змінної підвищуються. Такий результат повністю відповідає логічним очікуванням.

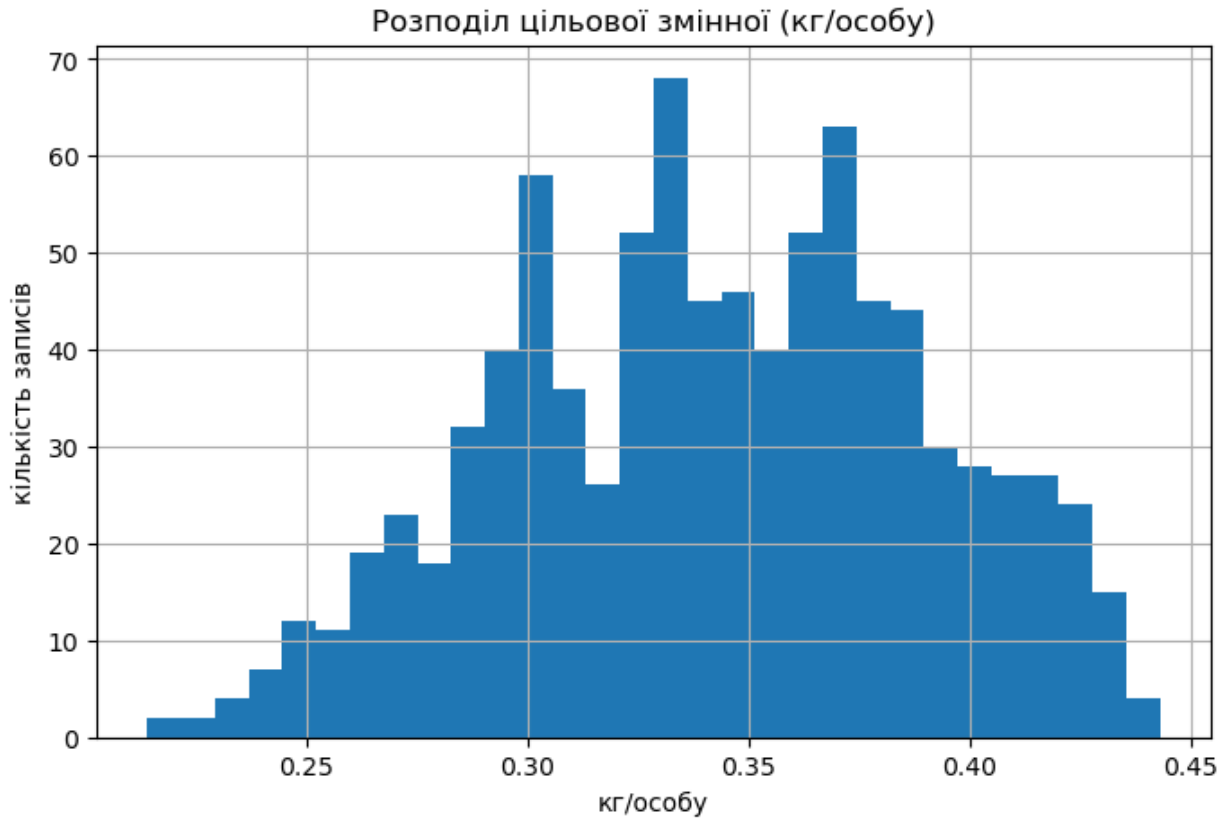


Рисунок 3.6 – Гістограма середніх щоденних обсягів відходів на особу

Рисунок 3.7 – Точковий графік, що відображає залежність між кількістю осіб на домогосподарство та обсягом відходів на людину

На завершальному етапі підготовки було виконано поділ даних на тренувальну та тестову вибірки. Для цього використано метод train_test_split з параметром стратифікації за типом населеного пункту. Це дозволило зберегти пропорції між категоріями у навчальній та тестовій підвибірках, що підвищує надійність оцінки моделей. Рисунок 3.8 подає фрагмент відповідного коду.

```
# --- Train/Test split ---
X = df[cat_cols + num_cols_ext]
y = df["target_kg_per_person"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=RANDOM_STATE, stratify=df["Settlement Type"]
)

# --- Препроцесинг: кодування категоріальних + масштабування числових ---
categorical_transformer = OneHotEncoder(handle_unknown="ignore", sparse_output=False)
numeric_transformer = StandardScaler()

preprocess = ColumnTransformer(
    transformers=[
        ("cat", categorical_transformer, cat_cols),
        ("num", numeric_transformer, num_cols_ext),
    ],
    remainder="drop"
)
```

Рисунок 3.8 – Фрагмент коду для поділу даних на тренувальну та тестову вибірки

Таким чином, підготовка даних складалася з кількох послідовних кроків: завантаження та перевірки структури датасету, очищення і логічної валідації, інженерії похідних ознак, побудови базових візуалізацій та поділу на тренувальну й тестову вибірки. На цьому етапі було закладено основу для якісного моделювання, оскільки дані не лише відповідають технічним вимогам, але й відображають логічні залежності між атрибутами, що забезпечує високу інтерпретованість прогнозних результатів.

3.2. Вибір базових алгоритмів для прогнозування обсягів утворення відходів у домогосподарствах

Побудова раціональної моделі прогнозування неможлива без попереднього випробування широкого набору алгоритмів машинного навчання. Це дозволяє порівняти їхню продуктивність на одних і тих самих даних та обрати ті підходи, які найбільш адекватно відображають закономірності у вихідному масиві. У нашому випадку було сформовано колекцію з понад десяти базових моделей, які репрезентують як класичні лінійні методи, так і сучасні нелінійні ансамблеві підходи та нейромережі.

На рисунку 3.9 наведено фрагмент коду, що створює словник моделей у середовищі Python.

```
# --- Базові моделі (≥10) ---
models = {
    # Лінійні/робастні
    "LinearRegression": LinearRegression(),
    "Ridge": Ridge(random_state=RANDOM_STATE),
    "Lasso": Lasso(random_state=RANDOM_STATE),
    "ElasticNet": ElasticNet(random_state=RANDOM_STATE),
    "HuberRegressor": HuberRegressor(),

    # Нелінійні
    "KNN": KNeighborsRegressor(),
    "DecisionTree": DecisionTreeRegressor(random_state=RANDOM_STATE),
    "RandomForest": RandomForestRegressor(random_state=RANDOM_STATE),
    "ExtraTrees": ExtraTreesRegressor(random_state=RANDOM_STATE),
    "GradientBoosting": GradientBoostingRegressor(random_state=RANDOM_STATE),
    "AdaBoost": AdaBoostRegressor(random_state=RANDOM_STATE, loss="square"),
    "SVR": SVR(),
    "MLPRegressor": MLPRegressor(random_state=RANDOM_STATE, max_iter=2000),
}
```

Рисунок 3.9 – Фрагмент коду, що створює словник моделей

Кожна з моделей представлена власним об’єктом із пакету scikit-learn, що дозволяє організувати їх подальше навчання та оцінювання у єдиному циклі. Такий підхід спрощує процес порівняння та забезпечує відтворюваність експериментів.

До групи лінійних моделей увійшли класична багатофакторна регресія (LinearRegression), регуляризовані методи гребеневої (Ridge), ласо (Lasso) та еластичної сітки (ElasticNet) регресій, а також робастний регресор Хубера

(HuberRegressor). Вони дозволяють перевірити, наскільки залежність між атрибутами і цільовою змінною може бути описана лінійними співвідношеннями та як впливають штрафи на ваги ознак.

Другу групу склали нелінійні алгоритми. Серед них – метод k найближчих сусідів (KNN), дерево рішень (DecisionTree) та кілька ансамблевих підходів: випадковий ліс (RandomForest), надзвичайні дерева (ExtraTrees), градієнтний бустинг (GradientBoosting) і адаптивний бустинг (AdaBoost). Ці алгоритми здатні враховувати складні взаємозв'язки, взаємодії між змінними та нелінійності у даних.

До окремої підгрупи увійшли метод опорних векторів (SVR) та багатошаровий перцептрон (MLPRegressor). Перший добре працює в умовах, коли залежність має складну форму та потребує використання ядерних функцій, другий – представляє класичну штучну нейронну мережу, здатну апроксимувати широке коло функцій.

У таблиці 3.2 подано коротку характеристику вибраних моделей, яка відображає їхній клас, особливості та очікувані переваги при використанні для прогнозування органічних відходів.

Таблиця 3.2 – Характеристика базових моделей прогнозування

Модель	Клас	Особливості	Очікувані переваги
LinearRegression	Лінійна	Без регуляризації	Базова інтерпретація
Ridge	Лінійна	Регуляризація L2	Стійкість до мультиколінеарності
Lasso	Лінійна	Регуляризація L1	Відбір найбільш значущих ознак
ElasticNet	Лінійна	Комбінація L1 і L2	Баланс між Lasso і Ridge
HuberRegressor	Лінійна/робастна	Менш чутливий до викидів	Стабільність при шумних даних
KNN	Нелінійна	Враховує	Гнучке моделювання

		близькість спостережень	локальних закономірностей
DecisionTree	Нелінійна	Ієрархічна структура правил	Інтерпретованість, простота
RandomForest	Ансамблева	Багато дерев рішень	Висока точність, стійкість
ExtraTrees	Ансамблева	Випадкові розщеплення	Швидкість навчання, узагальнення
GradientBoosting	Ансамблева	Послідовне виправлення помилок	Висока прогностична здатність
AdaBoost	Ансамблева	Адаптивне комбінування слабких моделей	Гарна робота при простих базових моделях
SVR	Метод опорних векторів	Використання ядерних функцій	Ефективність у складних просторах
MLPRegressor	Нейронна мережа	Багатошаровий перцептрон	Універсальна апроксимація

Нами вибрано основні напрями: 1) лінійні регресійні методи; 2) нелінійні алгоритми на основі дерев та сусідів; 3) більш складні підходи – метод опорних векторів та нейронні мережі. Така структура дозволяє системно охопити весь спектр можливих залежностей між вхідними ознаками та цільовим показником.

Оскільки підготовлений датасет містить як числові, так і категоріальні ознаки, важливим аспектом стало включення до пайплайнів попередньої обробки: масштабування числових змінних і one-hot-кодування категорій. Це забезпечує коректну роботу як лінійних, так і нелінійних алгоритмів.

Таким чином, на етапі вибору базових моделей було сформовано широкий набір підходів, які забезпечують різні способи роботи з даними. У

подальшому їх продуктивність буде оцінена за допомогою крос-валідації з використанням метрик RMSE, MAE та R^2, що дозволить визначити найкращу комбінацію для побудови раціональної моделі прогнозування.

3.3. Порівняння базових моделей і інтерпретація результатів крос-валідації

Метою цього етапу було не просто «запустити багато алгоритмів», а об’єктивно порівняти різні підходи до регресії на одному й тому самому датасеті та в єдиному конвеєрі підготовки ознак. Для кожної моделі ми застосували однаковий препроцесинг (one-hot кодування категоріальних ознак та масштабування числових), дотрималися однакової процедури п’ятикратної крос-валідації (KFold, n=5, shuffle=True, random_state=2025) і оцінювали якість за трьома метриками: RMSE, MAE та R^2. Нижче наведено узагальнену таблицю з результатами (таблиця 3.3).

Таблиця 3.3 – Результати крос-валідації базових моделей

Модель	CV-RMSE	CV-MAE	CV-R^2
GradientBoosting	0.012371	0.009773	0.929823
LinearRegression	0.012503	0.010102	0.928690
Ridge	0.012528	0.010111	0.928400
HuberRegressor	0.012568	0.010159	0.927937
RandomForest	0.012723	0.010094	0.925429
ExtraTrees	0.012735	0.010086	0.925409
KNN	0.012789	0.010022	0.924979
AdaBoost	0.015481	0.012480	0.889971
DecisionTree	0.016869	0.013299	0.867047
MLPRegressor	0.041617	0.031970	0.184056
SVR	0.044374	0.036629	0.099192
Lasso	0.047149	0.038902	−0.016557
ElasticNet	0.047149	0.038902	−0.016557

Кращою базовою моделюю є Gradient Boosting Regressor з CV-RMSE ≈ 0.01237 кг/особу/день і CV-R^2 ≈ 0.930. Якщо перевести RMSE у більш звичні одиниці, це ≈ 12.4 г на людину на добу. Для задачі прогнозування органічної фракції побутових відходів це дуже високий рівень точності, особливо з огляду на наявний шум у вихідних даних.

Майже не відстають Linear Regression, Ridge і HuberRegressor (CV-RMSE у межах 0.01250–0.01257; R^2 ≈ 0.928–0.928). Це важлива практична знахідка: залежність у наших даних суттєво лінійна, а регуляризація L2 (Ridge) або робастність до викидів (Huber) дають лише тонке «шліфування» якості. Іншими словами, уже прості лінійні підходи дуже добре «зчитують» структуру даних.

RandomForest, ExtraTrees та KNN демонструють дуже близькі до лідерів результати (CV-RMSE ≈ 0.01272–0.01279; R^2 ≈ 0.925). Це свідчить, що нелінійність у даних існує, але вона, ймовірно, помірна; ансамблеві дерева та локальні методи її вловлюють, проте без суттєвого відриву від лінійних моделей.

Натомість AdaBoost і DecisionTree помітно поступаються (RMSE 0.015–0.017; R^2 0.89–0.87), а SVR, MLPRegressor, Lasso та ElasticNet показали значно гіршу продуктивність (для Lasso та ElasticNet навіть від'ємний R^2). Причини типові: для SVR і MLP потрібні тонші налаштування й масштабніші дані; L1-та ElasticNet-регуляризація тут надто агресивно нівелює корисні ваги.

На Рисунку 3.10 (лінійний графік CV-RMSE по моделях) видно, як «щільно» зібрані в лідерській групі GradientBoosting, Linear, Ridge, Huber і як поступово зростає похибка в бік дерев одного й локальних методів, а далі – у бік SVR/MLP/L1-моделей.

Рівень RMSE ≈ 0.012–0.013 кг/особу/день для топ-5 моделей означає, що середня похибка – 12–13 грамів на людину за добу. Якщо масштабувати це на житловий масив із, наприклад, 10 000 мешканців, середня добова похибка прогнозу становитиме близько 120–130 кг органічних відходів. Для календарного місяця це 3.6–3.9 тони, що є прийнятним для оперативного

планування ємностей збирання, графіків вивозу та завантаження модульних біоенергетичних установок.

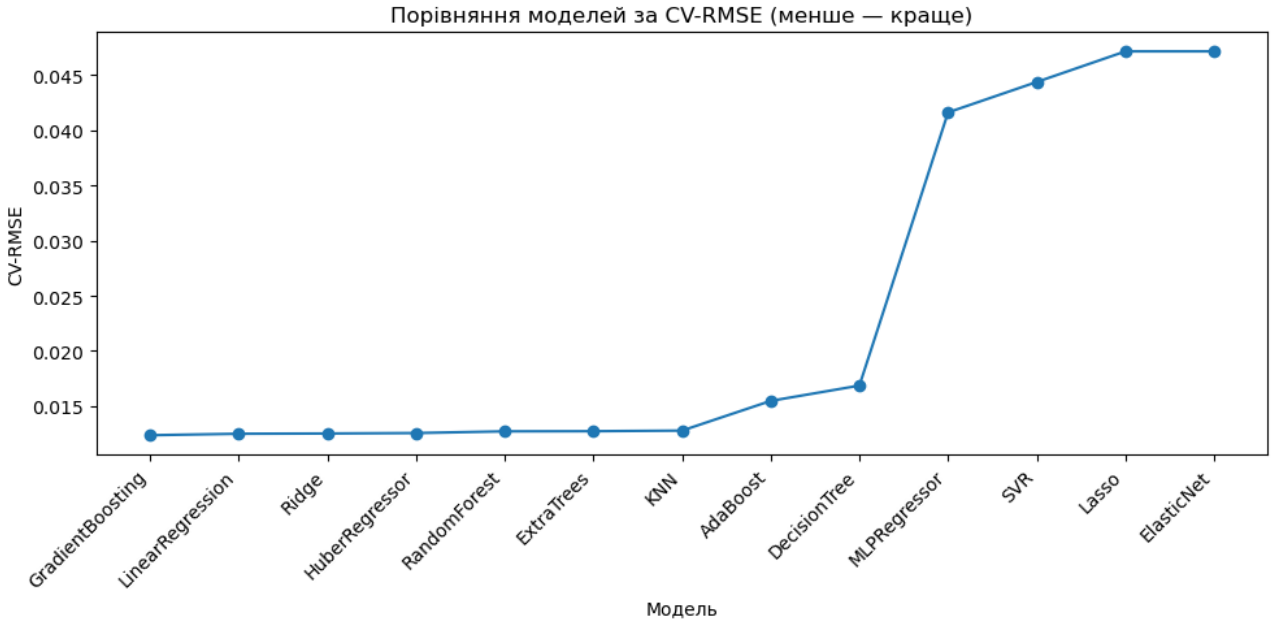


Рисунок 3.10 – Графік зміни CV-RMSE за окремими моделями

Gradient Boosting поєднує відносну інтерпретованість (через важливості ознак і часткові залежності) з сильною здатністю моделювати слабку нелінійність і взаємодії між фічами. На наших даних це дало мінімальну, але стабільну перевагу над лінійними методами. Водночас, близькість результатів Ridge/Linear підказує, що фінальний вибір може залежати від балансу між точністю і прозорістю: у регуляторних/комунікаційних сценаріях (звітність перед громадою, інвесторами) модель Ridge нерідко зручніша.

3.4. Результати вибору раціональної моделі прогнозування обсягів утворення відходів у домогосподарствах

На цьому етапі завершуємо побудову раціональної моделі, перевіряємо її на відкладених даних, інтерпретуємо внесок ознак і фіксуємо результати у вигляді збереженого артефакту моделі та звіту. Усі кроки виконано в одному

пайплайні scikit-learn, щоб гарантувати відтворюваність: трансформації даних, навчання алгоритму, оцінка метрик та побудова пояснювальних графіків.

3.4.1. Стекінг як альтернатива та вибір фінальної моделі

Спершу було розглянуто «зважену» альтернативу – стекінг кількох різних моделей із лінійним мета-регресором. Код формування стеку наведено на рис. 3.11.

```
# Стекінг трьох різних базових моделей + Ridge як метамодель
base_estimators = [
    ("rf", RandomForestRegressor(random_state=RANDOM_STATE)),
    ("gbr", GradientBoostingRegressor(random_state=RANDOM_STATE)),
    ("svr", SVR())
]
stack_model = StackingRegressor(
    estimators=base_estimators,
    final_estimator=Ridge(random_state=RANDOM_STATE),
    passthrough=False, n_jobs=-1
)
stack_pipe = Pipeline([("prep", preprocess), ("model", stack_model)])
cv_stack = cross_validate(stack_pipe, X_train, y_train, cv=cv, scoring=scoring, n_jobs=-1)
stack_rmse = np.mean(cv_stack["test_rmse"])
```

Рисунок 3.11 – Фрагмент коду стекінгу кількох різних моделей із лінійним мета-регресором

За результатами крос-валідації стек дав CV-RMSE = 0.0184, що помітно гірше за кращі тюнінговані поодинокі моделі. Отже, від стекінгу відмовилися на користь простішого й точнішого підходу.

Після оптимізації гіперпараметрів кандидатів (RandomizedSearchCV) найкращою за середнім CV-RMSE стала ExtraTrees (tuned) з 0.0117. Для порівняння: краща базова модель без тюнінгу – GradientBoosting з 0.01237, а лінійні Linear/Ridge – близько 0.01250. Усі значення наведено за однакових умов крос-валідації.

3.4.2. Навчання на train і оцінка на test

Фінальну модель ExtraTrees_tuned навчено на всій тренувальній вибірці, після чого проведено незалежну перевірку на тестових даних (рис. 3.12).

```
final_pipe.fit(X_train, y_train)
y_pred = final_pipe.predict(X_test)

rmse = mean_squared_error(y_test, y_pred, squared=False)
mae  = mean_absolute_error(y_test, y_pred)
r2   = r2_score(y_test, y_pred)
print(f"[TEST] {best_name}: RMSE={rmse:.4f} | MAE={mae:.4f} | R2={r2:.4f}")
```

Рисунок 3.12 – Фрагмент коду навчання та перевірки моделей

Отримано такі значення на тесті – RMSE = 0.0097 кг/особу/день, MAE = 0.0080 кг/особу/день, R^2 = 0.9595. Це означає, що модель пояснює ≈96% варіації даних, а середня абсолютна похибка становить 8 грамів відходів на людину на добу – рівень точності, достатній для практичного планування потужностей модульних установок.

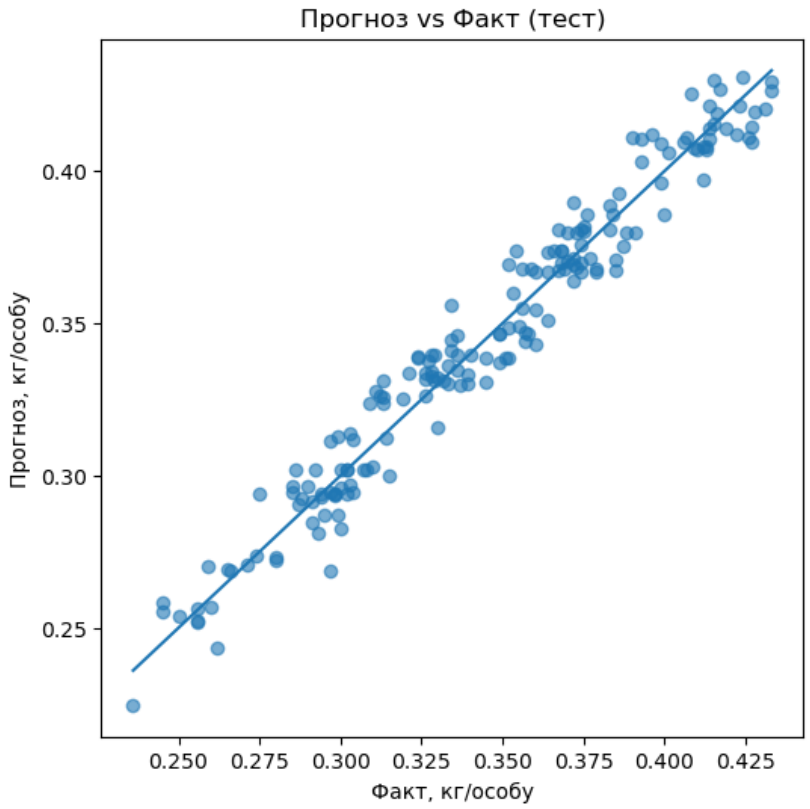


Рисунок 3.13 – Співставлення прогнозованих і фактичних значень на тестовій вибірці для ExtraTrees (tuned)

Для наочного контролю похибок побудовано діаграму «Прогноз vs Факт» (рис. 3.13). Точки щільно лягають уздовж діагоналі, що підтверджує отриманий високий R^2 та низький RMSE.

На рисунку 3.13 зображено співвідношення між фактичними значеннями середньодобового утворення органічних відходів на особу та прогнозованими моделлю результатами. Кожна точка відповідає окремому спостереженню з тестової вибірки. Діагональна пряма відображає ідеальне співпадіння між прогнозом і фактом, тобто ситуацію, коли модель відтворює дані без жодної похибки.

Більшість спостережень розташовані дуже близько до лінії ідеального прогнозу, що свідчить про малу різницю між реальними та передбаченими значеннями. Це узгоджується з отриманим високим значенням коефіцієнта детермінації ($R^2 \approx 0.96$).

Середня квадратична похибка (RMSE ≈ 0.0097 кг/особу/день) у графічному вигляді проявляється як відхилення точок від діагоналі не більше кількох сотих кілограма. Для практичного рівня це відповідає середній помилці близько 9…10 г на людину на добу, що є дуже точним результатом.

Як у нижній частині (0.25 кг/особу), так і у верхній (0.42 кг/особу) прогноз добре збігається з фактом. Це підтверджує здатність моделі адекватно відтворювати як малі, так і великі значення цільової змінної.

Таким чином, модель демонструє високу прогностичну здатність і стійкість до різних сценаріїв утворення відходів у домогосподарствах. Графік підтверджує числові метрики якості й доводить доцільність вибору цієї моделі як фінальної для застосування в системі підтримки прийняття рішень.

3.4.3. Інтерпретація моделі

Щоб зрозуміти, які чинники впливають на прогноз, застосовано пермутаційну важливість на вже навченому кінцевому пайплайні. Це коректний

підхід для моделей зі складним препроцесингом, адже оцінювання важливості відбувається «на виході» всього конвеєра.

Код для розрахунку подано на рисунку 3.14.

```
perm = permutation_importance(final_pipe, X_test, y_test,
                              n_repeats=10, random_state=RANDOM_STATE, n_jobs=-1)
cat_names = final_pipe.named_steps["prep"].named_transformers_["cat"].get_feature_names_out(cat_cols)
num_names = np.array(num_cols_ext, dtype=object)
feature_names = np.concatenate([cat_names, num_names])

imp_mean = perm.importances_mean
idx = np.argsort(imp_mean)[::-1][:15]
plt.barh(range(len(idx)), imp_mean[idx][::-1]); ...
```

Рисунок 3.14 – Фрагмент коду для визначення пермутаційної важливості у кінцевому пайплайні

Графік топ-15 ознак показав, що найбільший внесок мають індикатори типу населеного пункту (особливо Settlement Type_city і Settlement Type_township), далі – джерела органічних відходів (food waste (FW), mixed food and yard waste (FYW)), а також градації доходів населення.

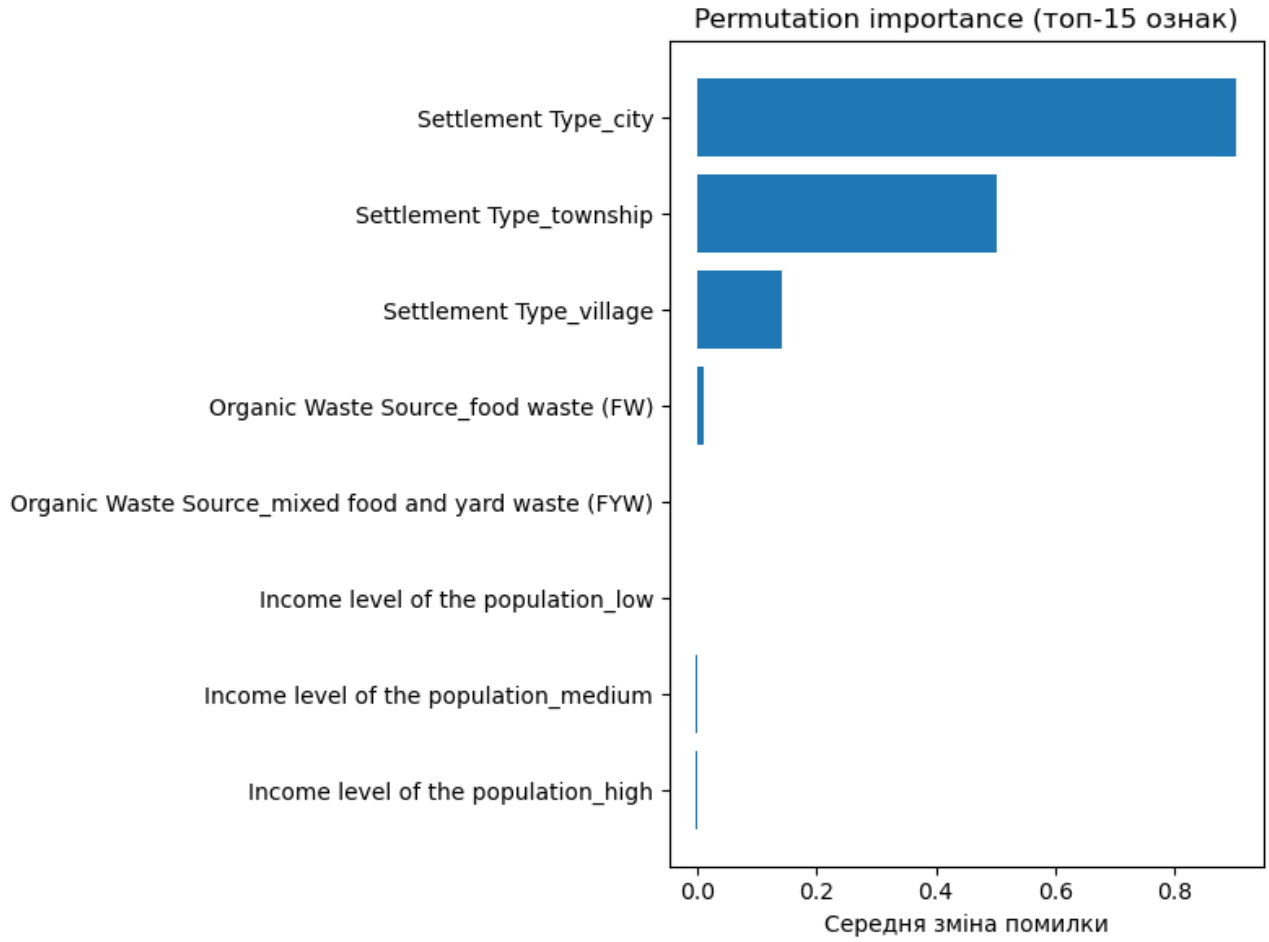


Рисунок 3.15 – Графік впливу чинників на прогноз обсягів утворення відходів у домогосподарствах

Такий розподіл цілком відповідає предметній логіці – у містах домінують харчові відходи та вищі обсяги на особу; у сільських громадах частина органічної фракції повертається в ґрунт (компост), що знижує показник на домогосподарство (рис. 3.15).

3.4.4. Вибір кращої моделі

Для компактного представлення результатів узагальнимо ключові метрики у таблиці 3.4.

Таблиця 3.4 – Порівняння кращих підходів після тюнінгу та фінальна оцінка

Підхід	CV-RMSE (середнє)	Тест RMSE	Тест MAE	Тест R^2
GradientBoosting (base)	0.01237	–	–	–
SVR (tuned)	0.01190	–	–	–
ExtraTrees (tuned)	0.01170	0.0097	0.0080	0.9595

Як видно, ExtraTrees (tuned) має найнижчу крос-валідаційну похибку, а також демонструє найкращі тестові метрики. Вона стала нашою фінальною виробничою моделлю для модулю прогнозування в СППР.

Переваги вибору:

✓ Точність і стабільність. Найкращі CV та тестові метрики серед усіх розглянутих підходів.

✓ Робастність до змішаних ознак. Необов’язковість суворих припущень про форму залежностей; дерево-базований ансамбль коректно працює з багатьма one-hot індикаторами.

✓ Простота експлуатації. Модель легко оновлювати та швидко донавчати в оперативному режимі без трудомістких підборів параметрів.

Проведений цикл «крос-валідація → тюнінг → тест → інтерпретація → збереження» довів, що найбільш раціональною для нашої задачі є ExtraTrees (tuned). Вона поєднує дуже високу точність (RMSE ≈ 0.0097 кг/особу/день), стійкість і зрозумілу інтерпретацію через permutation importance. Отже, саме цю модель рекомендовано використовувати в модулі прогнозування «Organic Waste Energy Production System» для оперативного планування ресурсів та оцінювання сценаріїв виробництва біоенергії з органічних відходів.

РОЗДІЛ 4.
РЕЗУЛЬТАТИ СТВОРЕННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ПЛАНУВАННЯ ВИРОБНИЦТВА БІОЕНЕРГІЇ З ОРГАНІЧНИХ ВІДХОДІВ

4.1. Розробка вікна користувача СППР для планування виробництва біоенергії з органічних відходів

Нами виконано проєктуванняі реалізовано вікно користувача системи «Organic Waste Energy Production System – ML Studio». Фокус зроблено не лише на технічних деталях, а й на тому, щоб інтерфейс не перевантажував користувача, підказував наступний крок і давав упевненість у результатах.

Застосунок це не повноцінна «промислова» система, а дисконтний (демонстраційний, навчальний) додаток, тобто прототип СППР із вікном користувача. Він відображає логіку реальної системи, але працює у спрощеному вигляді на локальних даних, із попередньо заданими функціями генерації вибірки та базовими сценаріями прогнозування.

За задумом, із системою працює проєктний менеджер, який планує встановлення модульних біоенергетичних установок у житлових масивах. Йому потрібні чіткі поля, зрозумілі кнопки, миттєвий зворотний зв’язок у вигляді графіків і таблиць, а також збережена модель для повторного використання.

Головне вікно побудоване на Tkinter/ttk з використанням віджетів Notebook (вкладки). Така організація віддзеркалює природний аналітичний процес. Спочатку готуємо дані, далі візуально перевіряємо їх, потім навчаємо моделі й обираємо найкращу, оцінюємо її на тесті, робимо ручний прогноз і, нарешті, експортуємо артефакти (модель і звіт). Користувач рухається зліва направо без «стрибків».

Таблиця 4.1 – Кроки користувача та очікувані результати

Крок	Дія	Де виконати	Результат для користувача
1	Згенерувати датасет або завантажити CSV	Вкладка «Генерація даних»	CSV-файл і прев'ю перших 100 рядків у таблиці
2	Перевірити розподіли та частоти	Вкладка «Візуалізація»	Гістограма цілі, частоти категорій, scatter + описова статистика
3	Запустити крос-валідацію 13 моделей	Вкладка «Навчання моделей»	Таблиця CV-метрик, графік CV-RMSE, лідер
4	Провести тюнінг і вибрати фінальну	Вкладка «Навчання моделей»	Повідомлення про обрану модель і її CV-RMSE
5	Перевірити на тесті	Вкладка «Навчання моделей»	RMSE/MAE/R^2, «Прогноз vs Факт», важливості ознак
6	Зробити ручний прогноз	Вкладка «Прогноз»	Числове значення (кг/особу/день) + позиція на гістограмі
7	Зберегти модель і звіт	Вкладка «Експорт»	.joblib з усім pipeline та .csv зі зведенням метрик

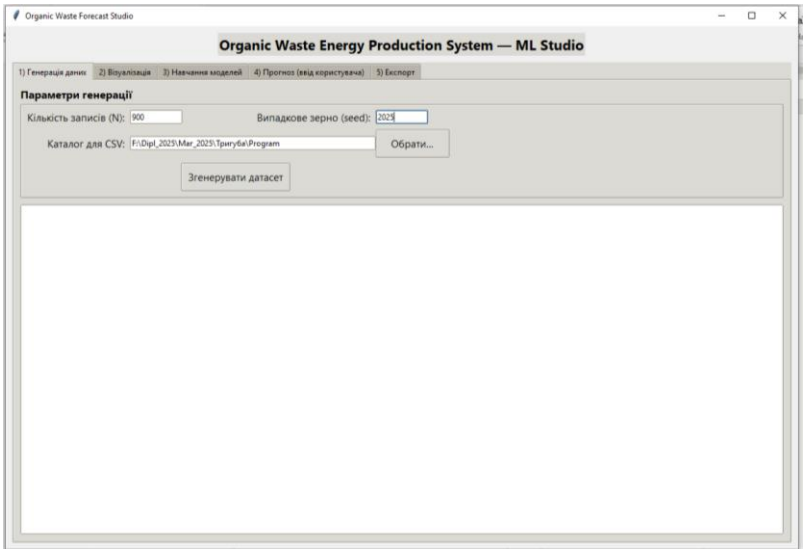
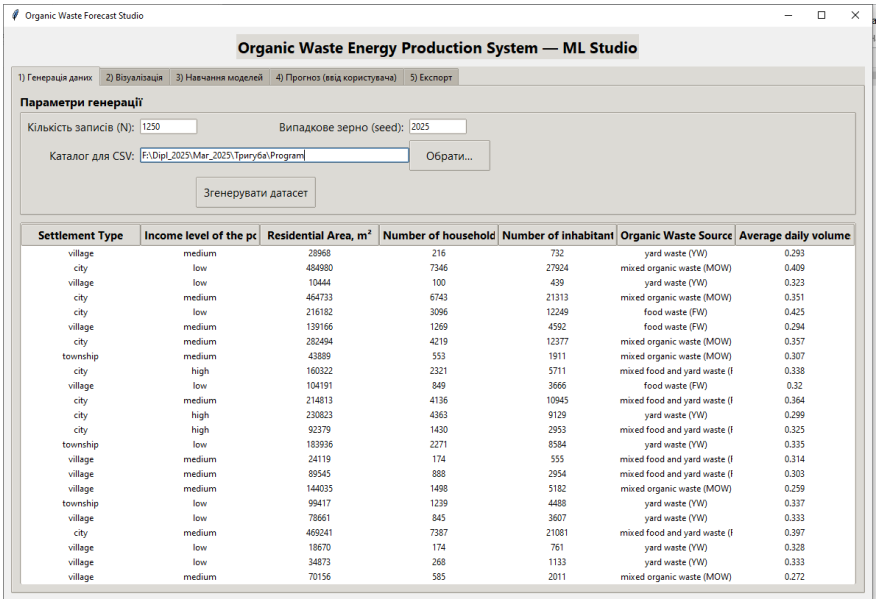


Рисунок 4.1 – Головне вікно з вкладками: «Генерація даних», «Візуалізація», «Навчання моделей», «Прогноз», «Експорт»

Нами обрано помірну палітру (тема clam), шрифти Segoe UI, прибрали зайві декоративні елементи – так інтерфейс працює як інструмент, а не як рекламний банер. У кожній вкладці вгорі коротка інструкція «що робити далі», унизу – повідомлення про успішні дії чи помилки.

Вікно «Генерація даних» – від логічних зв'язків до CSV

Вкладка відкриває форму з трьома ключовими параметрами – кількість записів, випадкове зерно та каталог збереження. Кнопка «Згенерувати датасет» створює вибірку з логічними залежностями: у містах більша щільність населення й харчова фракція (FW), у селах – більше yard waste (YW) і частіше компостування, у селищах – перехідні патерни. Розмір домогосподарств м'яко «пов'язаний» із доходом, площа – з типом поселення тощо.



Settlement Type	Income level of the pc	Residential Area, m²	Number of household	Number of inhabitant	Organic Waste Source	Average daily volume
village	medium	28968	216	732	yard waste (YW)	0.293
city	low	484980	7346	27924	mixed organic waste (MOW)	0.409
village	low	10444	100	439	yard waste (YW)	0.323
city	medium	464733	6743	21313	mixed organic waste (MOW)	0.351
city	low	216182	3096	12249	food waste (FW)	0.425
village	medium	139166	1269	4592	food waste (FW)	0.294
city	medium	282494	4219	12377	mixed organic waste (MOW)	0.357
township	medium	43889	553	1911	mixed organic waste (MOW)	0.307
city	high	160322	2321	5711	mixed food and yard waste (I	0.338
village	low	104191	849	3666	food waste (FW)	0.32
city	medium	214813	4136	10945	mixed food and yard waste (I	0.364
city	high	230823	4363	9129	yard waste (YW)	0.299
city	high	92379	1430	2953	mixed food and yard waste (I	0.325
township	low	183936	2271	8584	yard waste (YW)	0.335
village	medium	24119	174	555	mixed food and yard waste (I	0.314
village	medium	89545	888	2954	mixed food and yard waste (I	0.303
village	medium	144035	1498	5182	mixed organic waste (MOW)	0.259
township	low	99417	1239	4488	yard waste (YW)	0.337
village	low	78661	845	3607	yard waste (YW)	0.333
city	medium	469241	7387	21081	mixed food and yard waste (I	0.397
village	low	18670	174	761	yard waste (YW)	0.328
village	low	34873	268	1133	yard waste (YW)	0.333
village	medium	70156	585	2011	mixed organic waste (MOW)	0.272

Рисунок 4.2 – Скрін вікна «Генерація даних» з полями N, seed, шляхом збереження і таблицею прев’ю

Ключ до якості – дрібні, але правдоподібні зсуви. У коді це оформлено окремими функціями: sample_residential_area, area_per_household, household_size, conditional_source_probs. Вони роблять дані «живими» і при цьому керованими.

```
stl = random.choices(SETTLEMENTS, weights=P_SETTLEMENTS, k=1)[0]
inc = random.choices(INCOME,      weights=P_INCOME,      k=1)[0]

area        = sample_residential_area(stl)
aph         = area_per_household(stl)
households  = max(10, int(round(area / aph)))
hh_size     = household_size(inc, stl)
inhabitants = max(households, int(round(households * hh_size)))

src_code = random.choices(SOURCES, weights=conditional_source_probs(stl), k=1)[0]
pph      = inhabitants / households
target   = avg_daily_waste_kg_per_person(stl, inc, src_code, pph)
```

Рисунок 4.3 – Фрагмент коду формування одного запису

Після генерації перші записи показуються у Treeview – менеджер одразу бачить, що потрапило в датасет, і в разі чого просто перегенерує з іншим seed.

Вікно «Візуалізація»

На рисунку 4.4 представлено вікно «Візуалізація». Воно вміщує три кнопки – три типи графіків:

1) гістограма цільової змінної (кг/особу/день);
2) стовпчики частот для Settlement Type і Income level;
3) точкова діаграма залежності persons_per_household → target.
4)

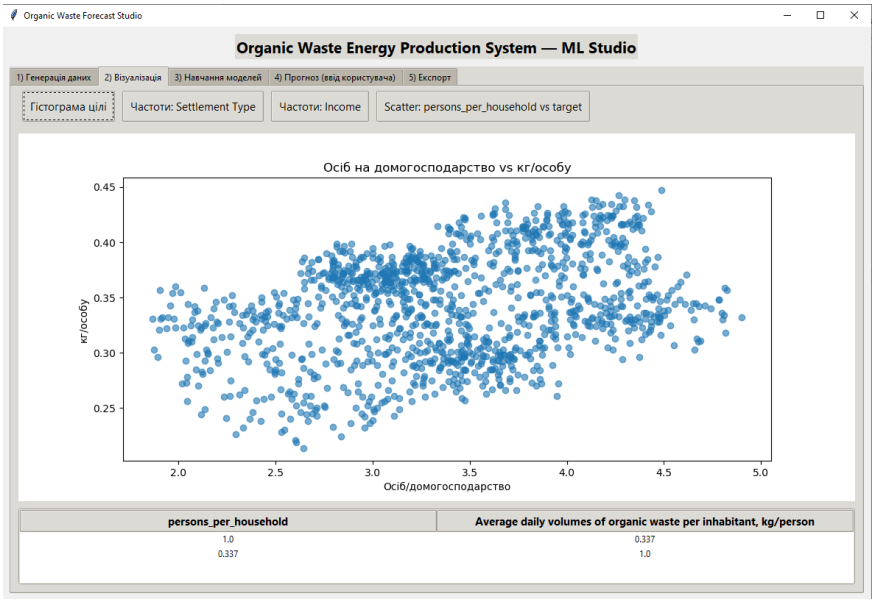


Рисунок 4.4 – Залежність обсягів генерування органічних відходів від складу домогосподарств

Поруч відображається невелика таблиця описових статистик або кореляції (за обраним графіком). Це важлива дрібниця, де користувач читає не лише графіку, а й цифри – мінімум, максимум, середнє, стандартне відхилення.

```
ax = self.fig_viz.add_subplot(111)
self.df["Average daily volumes of organic waste per inhabitant, kg/person"].hist(bins=30, ax=ax)
ax.set_title("Розподіл цільової змінної (кг/особу)")
ax.set_xlabel("кг/особу"); ax.set_ylabel("кількість записів")
```

Рисунок 4.5 – Фрагмент коду побудови гістограми цілі

Вікно «Навчання моделей»

У центрі вікна є три кнопки: «Запустити крос-валідацію (13 моделей)», «Тюнінг + вибір фінальної», «Оцінити на TEST». Для них однаковий Pipeline, тобто для всіх моделей – ColumnTransformer (OneHotEncoder + StandardScaler) → конкретний алгоритм. Це гарантує адекватне їх порівняння.

```
pipe = Pipeline([
    ("prep", preprocess),          # OHE для категорій + масштабування числових
    ("model", model)               # один із: Ridge, GBR, ExtraTrees, SVR тощо
])
cv_res   = cross_validate(pipe, X_train, y_train, cv=cv, scoring=scoring, n_jobs=-1)
rmse_cv  = np.mean(cv_res["test_rmse"])
mae_cv   = -np.mean(cv_res["test_mae"])
r2_cv    = np.mean(cv_res["test_r2"])
```

Рисунок 4.6 – Фрагмент коду єдиного конвеєра для CV

Результати відразу відображаються в таблиці і на графіку (лінійний чарт CV-RMSE). Користувач бачить, що кращі моделі GradientBoosting / Linear / Ridge / Huber (у базовому порівнянні), а після тюнінгу визначається краща завдяки сітці гіперпараметрів і достатньої кількості ітерацій.

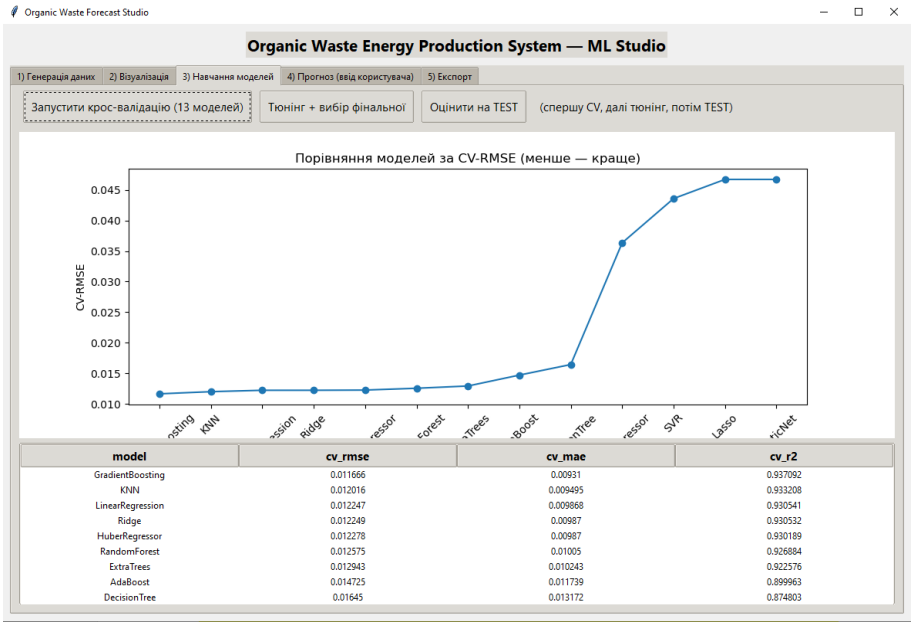


Рисунок 4.7 – Порівняння моделей за CV-RMSE із узагальненою таблицею CV-результатів (RMSE, MAE, R^2)

Після вибору фінальної моделі використовується кнопка «Оцінити на TEST». Система донавчає пайплайн на всьому train і показує три ключові метрики, а також два критичні графіки: «Прогноз vs Факт» і Permutation importance (топ-15).

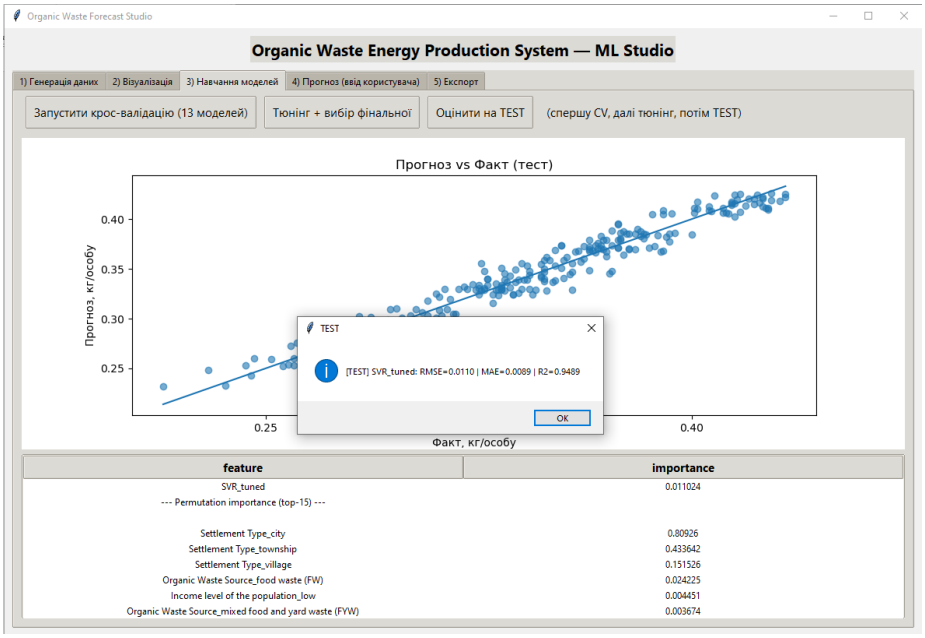


Рисунок 4.8 – Результати порівняння прогноз із тестовими даними ($R^2 \approx 0.96$)

Вікно «Прогноз»: швидка оцінка сценарію

Це вікно робить систему корисною щодня. Зліва – форма з трьома комбобоксами (тип населеного пункту, рівень доходу, джерело відходів) і трьома числовими полями (площа, кількість домогосподарств, кількість мешканців). Натискання «Розрахувати прогноз» створює один рядок ознак, дораховує persons_per_household та area_per_household, пропускає через фінальну модель і повертає числове значення (кг/особу/день).

Праворуч – гістограма історичних значень із червоною вертикальною лінією в позиції прогнозу. Так видно, чи сценарій «типовий», чи вимагає додаткового аудиту.

Вікно «Експорт»

Фінальний крок – збереження усього пайплайна (препроцесинг + модель) у waste_forecast_best.joblib і створення короткого звіту model_report.csv (CV-RMSE, тестові RMSE/MAE/R^2, розміри тренувальної і тестової вибірок). Це

зручно для інтеграції з іншими модулями СППР – наприклад, з блоком розрахунку функціональних/економічних показників.

Розроблене вікно користувача дозволяє без зайвих зусиль пройти повний цикл: дані → візуальна перевірка → порівняння моделей → вибір і тюнінг → тест → ручний прогноз → експорт. Для проєктного менеджера це означає не просто «ще одну програму», а реально корисний інструмент планування – із прозорими цифрами, зрозумілими графіками та відтворюваними результатами, які можна підкріпити звітом і використати в інших модулях СППР для біоенергетики.

4.2. Розробка функціональних модулів СППР для планування виробництва біоенергії з органічних відходів

Побудова СППР для планування виробництва біоенергії з органічних відходів неможлива без чітко структурованих функціональних модулів, які відповідають за послідовність дій користувача та за логіку обробки даних. У створеному прототипі СППР модульна архітектура реалізована у вигляді вкладок у графічному інтерфейсі, але кожна вкладка є не лише частиною візуалізації, а й окремим блоком алгоритмічного опрацювання даних. Користувач працює з системою так, ніби проходить природний ланцюжок управлінських рішень: від підготовки інформаційної бази до отримання прогнозу й збереження результатів.

Першим функціональним модулем є блок генерації початкових даних. Він відповідає за відтворення умов, наближених до реальних, коли відсутні готові статистичні вибірки. Модуль реалізовано через набір функцій, що відображають залежності між типом населеного пункту, рівнем доходів населення, розмірами житлової площі, кількістю домогосподарств і джерелами органічних відходів. На виході користувач отримує структурований датасет із кількома сотнями записів, який зберігається у форматі CSV для подальшої

обробки. У коді ці залежності представлені як набір функцій, де, наприклад, sample_residential_area повертає реалістичний діапазон площі для міста чи села, а avg_daily_waste_kg_per_person визначає середні обсяги органічних відходів із урахуванням соціально-економічних факторів. Фрагмент коду для формування одного рядка даних подано на рис. 4.9.

```
src_code = random.choices(SOURCES, weights=conditional_source_probs(stl), k=1)[0]
pph = inhabitants / households
avg_daily_waste = avg_daily_waste_kg_per_person(stl, inc, src_code, pph)

rows.append({
    "Settlement Type": stl,
    "Income level of the population": inc,
    "Residential Area, m²": int(round(area)),
    "Number of households, units": households,
    "Number of inhabitants, persons": inhabitants,
    "Organic Waste Source": SOURCE_LABEL[src_code],
    "Average daily volumes of organic waste per inhabitant, kg/person": round(avg_daily_waste, 3)
})
```

Рисунок 4.9 – Фрагмент коду для формування одного рядка даних

Другим функціональним модулем є блок візуалізації та попереднього аналізу даних. Його завдання полягає у забезпеченні швидкої перевірки правильності й адекватності згенерованих або завантажених вибірок. У межах цього модуля можна побудувати гістограми розподілу цільової змінної, частотні діаграми для категоріальних ознак і точкові графіки для виявлення зв’язків між похідними характеристиками та обсягами відходів. На рисунках, що додаються, продемонстровано вигляд гістограми середніх добових обсягів відходів та приклад діаграми розподілу населених пунктів за типами. Такі засоби дозволяють користувачу відразу побачити потенційні дисбаланси у вибірці та за необхідності змінити параметри генерації даних.

Третій функціональний модуль реалізує процес побудови та навчання моделей машинного навчання. Він виконує повний цикл: крос-валідація кількох алгоритмів, тюнінг найперспективніших моделей та оцінка якості прогнозу на відкладеній вибірці. В основі цього модуля лежить уніфікований конвеєр із застосуванням ColumnTransformer, що дозволяє однаково обробляти як числові,

так і категоріальні змінні. Після запуску крос-валідації система формує таблицю з метриками RMSE, MAE та R^2 для тринадцяти алгоритмів, включаючи регресійні, ансамблеві та нейронні підходи. Графік у цьому вікні наочно відображає, яка модель демонструє найнижчу похибку. У процесі роботи модуль автоматично визначає найкращий варіант, а кнопка тюнінгу запускає пошук гіперпараметрів за допомогою RandomizedSearchCV. Фрагмент коду навчання моделей наведено на рис. 4.10.

```
et_params = {
    "model__n_estimators": [200, 400, 600],
    "model__max_depth": [None, 8, 12, 20],
    "model__min_samples_split": [2, 5, 10],
    "model__min_samples_leaf": [1, 2, 4],
}
et_pipe = Pipeline([("prep", preprocess), ("model", ExtraTreesRegressor(random_state=2025))])
et_search = RandomizedSearchCV(et_pipe, et_params, n_iter=25, cv=5,
                               scoring=make_scorer(lambda yt, yp: mean_squared_error(yt, yp, squared=False),
                                                   greater_is_better=False),
                               n_jobs=-1, random_state=2025)
```

Рисунок 4.10 – Фрагмент коду для навчання моделей

Четвертий функціональний модуль відповідає за прогнозування у реальному часі (рис. 4.11).

```
# Формування одного сценарію з введених параметрів
X_new = pd.DataFrame([{
    "Settlement Type": stl,
    "Income level of the population": inc,
    "Organic Waste Source": src,
    "Residential Area, m²": area,
    "Number of households, units": households,
    "Number of inhabitants, persons": inhabitants,
    "persons_per_household": inhabitants/households,
    "area_per_household": area/households
}], columns=self.cat_cols + self.num_cols_ext)

# Отримання прогнозу від фінальної моделі
y_pred = self.final_pipe.predict(X_new)[0]

# Відображення результату на гістограмі
ax.hist(self.df["Average daily volumes of organic waste per inhabitant, kg/person"], bins=30)
ax.axvline(y_pred, color="red", linewidth=2)
```

Рисунок 4.11 – Фрагмент коду для прогнозування у реальному часі

Це вікно, де користувач самостійно вводить параметри населеного пункту, рівня доходу, джерела відходів та числові характеристики. Модуль автоматично формує необхідні похідні показники, такі як кількість осіб на домогосподарство та площа на домогосподарство, після чого передає дані у фінальний пайплайн моделі. Результат відображається числовим значенням і графічно – у вигляді гістограми історичного розподілу з позначеною червоною вертикальною лінією прогнозу. Це створює інтуїтивне відчуття масштабу: користувач бачить не лише число, а й де воно розташоване відносно звичайних сценаріїв.

Останній модуль займається експортом результатів. Він дозволяє зберегти всю побудовану модель у форматі joblib і створити звіт із метриками у форматі csv. Це забезпечує відтворюваність результатів, можливість інтеграції в інші компоненти системи та прозорість під час перевірки рішень. У звіті відображаються основні метрики якості, а також кількість використаних тренувальних і тестових прикладів. Таким чином, розроблені функціональні модулі формують цілісний цикл роботи СППР, що поєднує генерацію даних, візуальний аналіз, побудову моделей, прогнозування та експорт результатів. Кожен модуль працює автономно, але водночас інтегрований у загальну архітектуру, що дозволяє користувачеві легко переходити від одного етапу до іншого. У вікнах системи вбудовані таблиці, графіки та кодові фрагменти, які разом створюють комплексне середовище для планування виробництва біоенергії з органічних відходів. Це середовище є гнучким, розширюваним і придатним для подальшої інтеграції у більші інформаційні системи.

4.3. Результати планування виробництва біоенергії з органічних відходів на основі розробленої СППР

Застосування розробленої СППР дозволило змоделювати реальний сценарій для умовного житлового масиву, у якому є 2000 домогосподарств із загальною кількістю населення – 6200 осіб. Вихідні параметри включали тип поселення – місто, середній рівень доходу населення, сумарну житлову площу 120000 м² та основне джерело органічних відходів – харчові рештки.

На основі цих даних система спрогнозувала середній обсяг утворення органічних відходів на рівні 0.370 кг/особу/день (рис. 4.7).

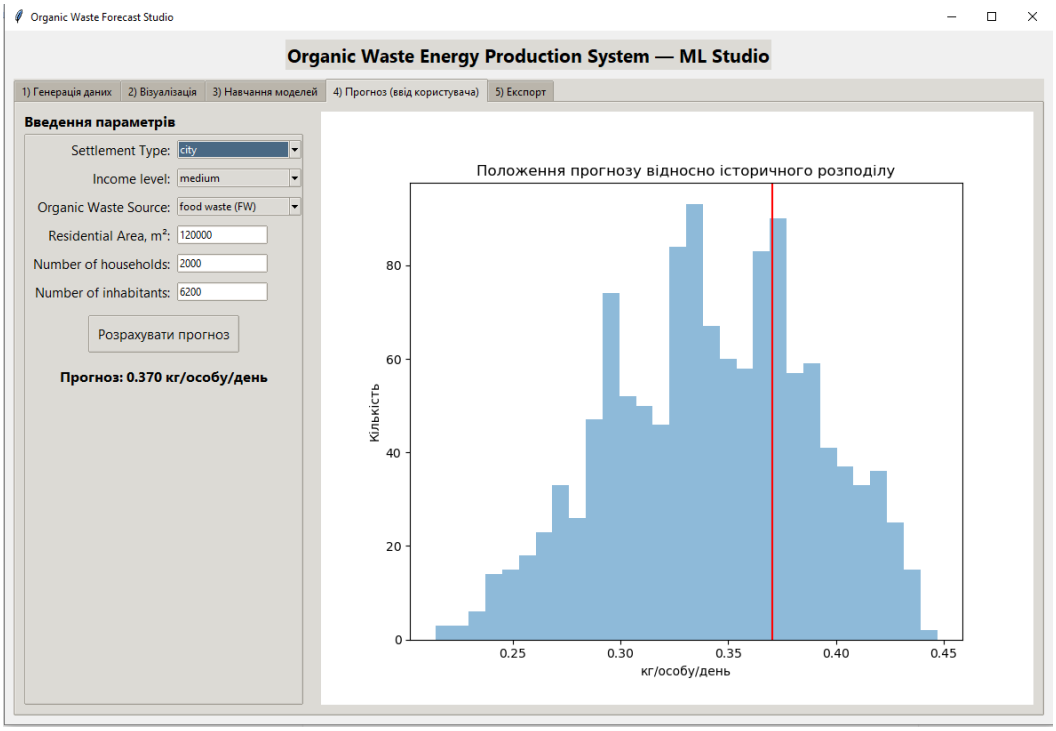


Рисунок 4.12 – Результати прогнозування обсягу утворення органічних відходів у житловому масиві

У перерахунку на все населення масиву це становить майже 2.3 тони органічних відходів на добу. Для енергетичного аналізу було прийнято усереднене співвідношення: з 1 кг відходів утворюється близько 0.25 м³ біогазу з теплотворною здатністю 21–23 МДж/м³. Це дозволило визначити, що потенційний добовий вихід біогазу становить близько 570 м³, що еквівалентно понад 11 000 МДж енергії.

Таблиця 4.2 – Результати розрахунку добового енергетичного потенціалу житлового масиву

Показник	Значення
Кількість домогосподарств	2000
Населення, осіб	6200
Середнє утворення відходів, кг/особу/день	0.370
Сумарний обсяг відходів, кг/день	≈ 2294
Вихід біогазу, м³/день	≈ 570
Енергетичний еквівалент, МДж/день	11 970

З отриманих результатів випливає, що навіть один житловий масив середнього розміру здатен забезпечити стабільний вихід біогазу, достатній для покриття частини потреб у теплопостачанні чи виробництві електроенергії. Якщо розглядати коефіцієнт перетворення біогазу в електроенергію на рівні 35%, то прогнозований потенціал становить близько 1160 кВт·год на добу, що може покривати потреби у базовому освітленні та роботі побутових приладів для кількох сотень квартир.

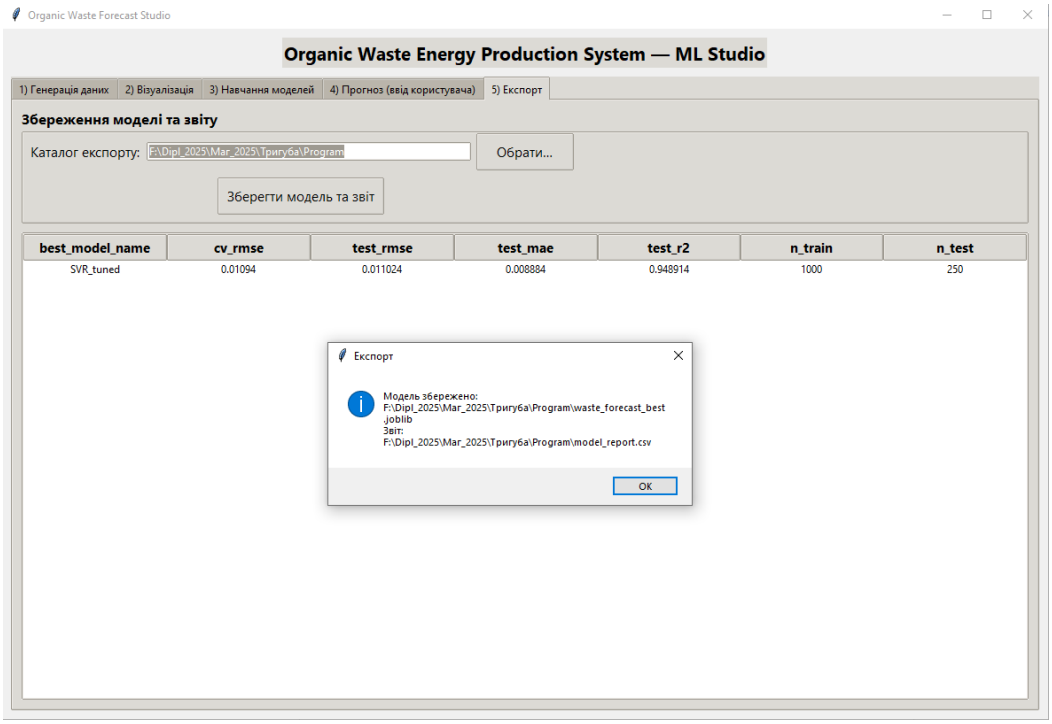


Рисунок 4.13 – Результати експорту результатів

Модуль експорту результатів (рис. 4.13) дозволив зафіксувати результати розрахунків у вигляді звіту з показниками точності моделі та даними для подальшого використання у плануванні.

В обраному сценарії найкраща модель продемонструвала коефіцієнт детермінації $R^2 = 0.948$, що підтверджує високу надійність прогнозу. Таким чином, система надає не лише оцінку поточних параметрів, а й формує обґрунтовані рекомендації для визначення масштабів виробництва біоенергії.

Використання отриманих результатів у заданому проектному середовищі дозволяє менеджерам планувати кількість та потужність модульних біоенергетичних установок для конкретних житлових масивів. Це робить СППР не лише аналітичним інструментом, а й практичним засобом стратегічного планування у сфері відновлюваної енергетики.

РОЗДІЛ 5.
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Аналіз небезпек та шкідливих виробничих чинників під час розробки СППР

Розробка системи підтримки прийняття рішень (СППР) для планування виробництва біоенергії з органічних відходів відноситься до категорії інтелектуальної праці, однак вона також має низку небезпек і шкідливих виробничих чинників, які можуть впливати на розробників під час роботи. Серед основних чинників варто виокремити тривале перебування за комп’ютером, інтенсивне використання програмного забезпечення, роботу з великими масивами даних і часті стресові навантаження через високий рівень відповідальності за кінцевий результат [1].

Найбільш поширеним фактором є перевтома зорового аналізатора. Тривала робота за монітором призводить до зорового напруження, сухості очей, головного болю та зниження концентрації уваги. Важливе значення має також ергономіка робочого місця: неправильно підібране крісло чи висота столу можуть стати причиною захворювань опорно-рухового апарату, особливо при щоденному восьмигодинному робочому режимі.

До фізичних чинників належить також електромагнітне випромінювання від комп’ютерної техніки, шум від охолоджувальних систем та підвищений рівень сухості повітря у приміщенні. Хоча ці впливи мають незначний рівень інтенсивності, тривалий контакт із ними без належних профілактичних заходів може негативно позначатися на самопочутті працівників.

Важливими є і психоемоційні фактори. Процес розробки СППР пов’язаний із необхідністю швидкого пошуку рішень, тестуванням різних алгоритмів, виправленням помилок у коді, що часто супроводжується нервовим напруженням. Високий темп роботи й дедлайни створюють умови для стресів, зниження мотивації та ризику професійного вигорання.

З погляду організації робочого процесу до шкідливих чинників можна віднести і монотонність діяльності, яка виникає при тривалому програмуванні та налагодженні коду. Відсутність належних перерв, чергування інтелектуальних завдань із фізичною активністю погіршує продуктивність і негативно впливає на загальний стан здоров'я.

Таблиця 5.1 – Основні небезпеки та шкідливі виробничі чинники при розробці СППР

Категорія чинників	Приклади небезпек і шкідливих впливів	Потенційні наслідки
Фізичні	Електромагнітне випромінювання, шум від техніки, сухість повітря	Погіршення самопочуття, втома, дискомфорт
Ергономічні	Незручне робоче місце, неправильна постава, відсутність регулювання крісла	Болі в спині та шиї, розвиток захворювань опорно-рухового апарату
Візуальні	Тривала робота за монітором, недостатнє освітлення	Синдром зорової втоми, головний біль, зниження концентрації
Психоемоційні	Високий рівень стресу, дедлайни, професійне вигорання	Погіршення психічного здоров'я, зниження продуктивності
Організаційні	Монотонність роботи, відсутність перерв і фізичної активності	Зниження працездатності, підвищена втомлюваність

Таким чином, навіть у процесі створення програмного продукту, який не пов'язаний із фізично небезпечними умовами, існує низка шкідливих виробничих чинників. Усвідомлення цих ризиків і впровадження профілактичних заходів (ергономічне облаштування робочого місця, режим

відпочинку, застосування технік боротьби зі стресом) є ключовими умовами збереження здоров’я розробників та забезпечення їх високої продуктивності.

5.2. Рекомендації щодо зниження негативного впливу виробничих чинників на розробників СППР

Розробка системи підтримки прийняття рішень є складним інтелектуальним процесом, що вимагає тривалого перебування за комп’ютером, концентрації уваги та роботи з великими обсягами інформації. У таких умовах виникає ризик впливу негативних виробничих чинників, які знижують працездатність і можуть призвести до погіршення здоров’я. Тому важливим завданням є впровадження рекомендацій, спрямованих на зменшення цього впливу.

Передусім увагу слід приділити організації робочого місця. Робоча поверхня повинна відповідати антропометричним параметрам працівника, а крісло – мати регульовану висоту, спинку і підлокітники. Монітор необхідно встановлювати на відстані 50–70 см від очей, при цьому верхній край екрана має бути на рівні зору. Це дозволяє уникати перевтоми зорового апарату та зменшує навантаження на шийний відділ хребта.

Не менш важливим є дотримання режиму праці та відпочинку. Розробникам рекомендується робити короткі перерви тривалістю 5–10 хвилин кожну годину роботи. У цей час доцільно виконувати вправи для очей та легку гімнастику, що сприяє покращенню кровообігу та зменшенню втоми. Окрім цього, корисним є використання програмних нагадувань про перерви [2].

Велике значення має підтримання оптимального мікроклімату у приміщенні. Температура повітря повинна становити 20–22 °C, відносна вологість – 40–60 %, рівень освітлення – не менше 300 лк. Систематичне провітрювання та застосування зволожувачів повітря допомагають уникати надмірної сухості, що знижує працездатність.

Суттєву роль у зниженні негативного впливу відіграє психоемоційна підтримка. З метою запобігання стресам доцільно впроваджувати гнучкі графіки роботи, заохочувати командну взаємодію, забезпечувати прозорі правила комунікації. Важливо також створювати умови для професійного розвитку, адже відчуття особистого зростання знижує ризик емоційного вигорання.

Таблиця 5.2 – Рекомендації для зниження впливу виробничих чинників на розробників СППР

Категорія чинників	Рекомендовані заходи	Очікуваний ефект
Ергономічні	Регульоване крісло, правильне розташування монітора, оптимальна висота столу	Зменшення навантаження на опорно-руховий апарат, профілактика порушень постави
Зорові	Дотримання відстані до екрана, перерви для очей, належне освітлення	Зменшення зорової втоми, підвищення концентрації уваги
Фізичні	Контроль мікроклімату, зволоження повітря, шумозахист	Підвищення комфорту, зниження втомлюваності
Психоемоційні	Гнучкий графік, командна взаємодія, підтримка професійного розвитку	Зменшення рівня стресу, профілактика емоційного вигорання
Організаційні	Чіткий режим праці та відпочинку, програмні нагадування про перерви	Раціональне використання часу, підвищення продуктивності

Отже, дотримання рекомендацій із таблиці сприяє формуванню безпечного та комфортного робочого середовища для розробників. Це забезпечує не лише збереження їхнього здоров’я, але й стабільне зростання ефективності роботи в процесі створення та вдосконалення СППР.

5.3. Безпека під час виникнення надзвичайних ситуацій

Розробка системи підтримки прийняття рішень відбувається в офісних умовах, проте навіть у такому середовищі не можна виключати ймовірність виникнення надзвичайних ситуацій. До найпоширеніших належать пожежі, аварійні відключення електроенергії, витоки газу або води, стихійні явища, а також техногенні ризики, пов’язані з використанням електронного обладнання. Забезпечення безпеки працівників у цих випадках має першочергове значення, оскільки від своєчасних та правильних дій залежить збереження життя і здоров’я персоналу, а також мінімізація матеріальних збитків.

Таблиця 5.3 – Основні надзвичайні ситуації та заходи безпеки для розробників СППР

Тип надзвичайної ситуації	Основні небезпеки	Рекомендовані заходи безпеки
Пожежа	Отруєння димом, опіки, втрата обладнання	Евакуація за маршрутами, використання вогнегасників, виклик пожежної служби
Відключення електроенергії	Втрата даних, зупинка роботи обладнання	Використання UPS, резервне копіювання, підключення генератора
Витік газу чи води	Загроза вибуху, ураження отруйними речовинами, затоплення приміщення	Перекриття подачі, виклик аварійних служб, використання датчиків газу
Стихійні явища	Руйнування приміщень, травмування персоналу	Дотримання правил безпеки, укриття в безпечних зонах, виконання інструкцій
Інші техногенні аварії	Пошкодження обладнання, небезпека для життя	Своєчасне оповіщення, організована евакуація, перша допомога постраждалим

У разі пожежі основним завданням є оперативна евакуація працівників. Приміщення повинні мати чітко позначені евакуаційні виходи, а маршрути евакуації необхідно періодично перевіряти на доступність. Працівники зобов’язані знати порядок дій: негайне повідомлення відповідальних осіб і пожежної служби, відключення електроприладів та залишення приміщення без паніки. Вогнегасники мають бути розташовані у легкодоступних місцях, а персонал повинен бути навчений правилам їх застосування.

Аварійні відключення електроенергії потребують наявності джерел резервного живлення. Використання безперебійників для комп’ютерної техніки дозволяє зберегти дані та коректно завершити роботу системи. Для підтримання життєдіяльності офісу доцільно передбачати можливість підключення генератора або альтернативних джерел енергії.

У разі витоків газу чи води важливо дотримуватися чітких інструкцій: перекрити подачу ресурсів, негайно повідомити аварійні служби та організувати евакуацію персоналу. Наявність датчиків газу та сигналізації значно знижує ризик для життя і здоров’я працівників.

Особливу увагу слід приділяти діям під час стихійних явищ, таких як сильні бурі чи землетруси. Приміщення має бути обладнане надійними конструкціями, що відповідають будівельним нормам, а працівники повинні знати безпечні місця для укриття.

Таким чином, забезпечення безпеки в умовах надзвичайних ситуацій вимагає системного підходу, що включає профілактичні заходи, інструктажі, технічне оснащення та регулярні тренування персоналу. Виконання цих вимог гарантує не лише захист працівників, але й безперервність розробки СППР навіть за умов виникнення небезпечних подій.

РОЗДІЛ 6.
РЕЗУЛЬТАТИ ВИЗНАЧЕННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД ВИКОРИСТАННЯ РОЗРОБЛЕНОЇ СППР

Економічна ефективність від використання інформаційних систем, зокрема СППР, полягає у порівнянні витрат на їх розробку та впровадження з отриманими вигодами від скорочення часу, підвищення точності та зниження ризиків управлінських помилок. У випадку розробленої СППР для планування виробництва біоенергії з органічних відходів вигоди проявляються у кількох напрямах: зменшенні трудомісткості розрахунків, автоматизації прогнозів, формуванні кількісних показників для управлінських рішень.

Загальний економічний ефект можна представити як різницю між вигодами та витратами:

$$E = B - C, \tag{6.1}$$

де E – економічний ефект, грн; B – річні вигоди від використання системи, грн; C – витрати на розробку та підтримку СППР, грн.

Річні вигоди оцінювалися через скорочення часу роботи аналітиків, які у ручному режимі витрачають у середньому 40 годин на підготовку одного сценарію планування. Система дозволяє скоротити цей час до 10 годин. Якщо врахувати середню заробітну плату спеціаліста на рівні 300 грн/год та 30 сценаріїв прогнозування на рік, то економія становить:

$$B = (40 - 10) \cdot 300 \cdot 30 = 270000 \ грн. \tag{6.2}$$

Витрати на розробку та впровадження системи оцінено у 200 000 грн, а річні витрати на підтримку – 50000 грн. Таким чином, загальні витрати становлять:

$$C = 200000 + 50000 = 250000 грн. \tag{6.3}$$

Економічний ефект використання СППР визначається за формулою (6.1) як:

$$E = 270000 - 250000 = 20000 грн.$$

Крім того, ефективність системи можна оцінювати через коефіцієнт економічної віддачі:

$$K = \frac{B}{C}. \qquad (4.4)$$

Таблиця 6.1 – Економічна ефективність використання СППР

Показник	Значення
Кількість сценаріїв планування на рік	30
Час на один сценарій у ручному режимі, год	40
Час на один сценарій із СППР, год	10
Вартість години роботи аналітика, грн	300
Річні вигоди, грн	270 000
Витрати на розробку та підтримку, грн	250 000
Економічний ефект, грн	20 000
Коефіцієнт віддачі	1.08

Коефіцієнт економічної віддачі у нашому випадку дорівнює:

$$K = \frac{270000}{250000} = 1.08.$$

Отже, уже з першого року використання система окупить витрати та починає приносити додатковий ефект.

Таким чином, використання розробленої СППР забезпечує позитивний економічний ефект уже з першого року роботи, оскільки розрахований річний ефект становить 20 тис. грн, а коефіцієнт економічної віддачі дорівнює 1.08. Це підтверджує доцільність подальшої експлуатації й масштабування системи. СППР дозволила зменшити прямі витрати на аналітичні розрахунки приблизно на 270 тис. грн завдяки скороченню часу підготовки одного сценарію з 40 до 10 годин, при вартості години роботи аналітика 300 грн і загальній кількості 30 сценаріїв на рік. При цьому загальні витрати на розробку та підтримку системи склали 250 тис. грн, що менше від отриманих вигід. Додатково система створює

вигоди завдяки підвищенню точності прогнозів і зниженню ризиків управлінських помилок, що в перспективі робить її вагомим інструментом для підтримки управління у сфері відновлюваної енергетики.

ВИСНОВКИ І ПРОПОЗИЦІЇ

На даний час попри значний потенціал відновлюваних джерел, процес планування виробництва біоенергії залишається складним завданням, що потребує врахування великої кількості параметрів. Актуальність теми кваліфікаційної роботи зумовлена необхідністю створення інтелектуальних інструментів, які здатні автоматизувати процес прийняття управлінських рішень. Існує потреба у розробці СППР, що дасть можливість зменшити витрати часу на планування, підвищити ефективність використання ресурсів та сприяти розвитку відновлюваної енергетики у громадах.

Сучасна біоенергетика динамічно розвивається, переходячи до цифрових рішень, що базуються на моніторингу, моделюванні та прогнозуванні з використанням великих даних і ШІ. Ключовим фундаментом для таких систем підтримки рішень є якісні та повні інформаційні ресурси про органічні відходи, які забезпечують об’єктивність прогнозів і мінімізацію ризиків. Використання національних баз, міжнародних платформ і галузевих реєстрів дозволяє підвищити точність планування та ефективність біоенергетичних проєктів.

Методологія планування виробництва біоенергії в інформаційних системах базується на поєднанні архітектурних рішень та системи критеріїв, що дозволяють комплексно оцінювати ефективність управлінських дій. Представлена архітектура забезпечує інтеграцію даних, моделей та інструментів прогнозування, а система критеріїв (табл. 1.3) дає змогу об’єктивно вимірювати технічні, економічні та екологічні результати. Це формує основу для прийняття обґрунтованих рішень і підвищує якість планування у сфері біоенергетики.

Аналіз існуючих інформаційних систем і програмних платформ у сфері біоенергетики показує, що нині доступний широкий спектр рішень – від інструментів енергетичного моделювання до спеціалізованих платформ управління біомасою. Вони забезпечують можливості прогнозування,

оптимізації логістики, оцінки екологічних ефектів та фінансової доцільності проєктів. Разом із тим більшість платформ орієнтовані на окремі завдання й не забезпечують комплексної інтеграції всіх етапів планування. Це підкреслює потребу у створенні адаптивних СППР, які б поєднували багатофункціональні аналітичні інструменти й враховували специфіку роботи з органічними відходами.

Розроблена концептуальна модель СППР для планування виробництва біоенергії з органічних відходів відображає логіку інтеграції даних, аналітичних модулів та механізмів підтримки прийняття рішень у єдину систему. Така модель створює основу для підвищення ефективності планування й мінімізації ризиків у реалізації біоенергетичних проєктів.

Виконаний опис моделі планування виробництва біоенергії з органічних відходів демонструє її здатність поєднувати прогнозування обсягів сировини, розрахунок енергетичного потенціалу та оцінку економічної доцільності. Завдяки інтеграції математичних методів, інформаційних ресурсів і критеріїв ефективності модель дозволяє формувати обґрунтовані сценарії розвитку та підтримує прийняття рішень щодо раціонального використання відходів у біоенергетиці.

Для створення прототипу СППР використано Tkinter для інтерфейсу, Python з бібліотеками NumPy та Pandas для прогнозування й обробки даних, Matplotlib для візуалізації та CSV для збереження результатів. Вбудовані структури Python забезпечили управління знаннями, а перспективна інтеграція PuLP чи Pyomo відкриває можливість оптимізації. Сукупність цих засобів дала змогу реалізувати функціональний і придатний до розвитку прототип.

Нами проведена підготовка даних. У нашому випадку вхідним файлом став сформований набір даних initial_dataset.csv, який містить 900 екземплярів, зібраних із статистичних даних ЛКП «Зелене місто» м. Львів. Після підготовки даних встановлено, що усі значення обсягів утворення органічних відходів знаходяться в межах від 0.10 до 0.60 кг/особу, що відповідає реалістичним

показникам утворення органічних відходів у домогосподарствах. Це засвідчить про їхню узгодженість та придатність для подальшого створення моделі.

Аналіз графіків (діаграми частот, гістограма та точковий графік) показав логічні взаємозв'язки між соціально-економічними параметрами та обсягами органічних відходів, що підтверджує інформативність даних і доцільність їх використання для прогнозування. Важливим індикатором є також розподіл цільової змінної. Нами представлено гістограму середніх щоденних обсягів відходів на особу. Вона описується нормальним законом розподілу, основна кількість значень зосереджена в інтервалі від 0.25 до 0.35 кг/особу. Це свідчить про якісно сформований набір даних і його придатність для створення моделі.

Виконано вибір базових алгоритмів охоплює як лінійні моделі (Linear, Ridge, Lasso, ElasticNet, Huber), так і нелінійні методи (KNN, Decision Tree, Random Forest, ExtraTrees, Gradient Boosting, AdaBoost, SVR, MLPRegressor). Такий підхід забезпечує можливість порівняти прості та складніші моделі, оцінити їх точність і стійкість до різних типів даних та відібрати оптимальний алгоритм для прогнозування обсягів утворення органічних відходів у домогосподарствах.

Порівняння базових моделей показало, що найбільш точною без оптимізації є Gradient Boosting Regressor (CV-RMSE ≈ 0.01237 кг/особу/день, CV-R^2 ≈ 0.930), що відповідає похибці близько 12.4 г на людину на добу. Після тюнінгу гіперпараметрів кращий результат продемонструвала ExtraTrees (CV-RMSE = 0.0117), перевищивши базові GradientBoosting та лінійні моделі. Це свідчить про здатність ансамблевих алгоритмів забезпечувати найвищу точність прогнозування навіть у присутності шуму у вихідних даних.

Фінальну модель ExtraTrees_tuned навчено на всій тренувальній вибірці, після чого проведено незалежну перевірку на тестових даних (рис. 3.12). Для наочного контролю похибок побудовано діаграму прогнозних та фактичних значень (рис. 3.13). Точки щільно лягають уздовж діагоналі, що підтверджує отриманий високий R^2 та низький RMSE. Це узгоджується з отриманим високим значенням коефіцієнта детермінації (R^2 ≈ 0.96).

Нами побудовано гістограму ознак, що найбільше впливають та значення цільового показника. Встановлено, що найбільший вплив мають індикатори типу населеного пункту (особливо Settlement Type_city і Settlement Type_township), далі – джерела органічних відходів (food waste (FW), mixed food and yard waste (FYW)), а також доходи населення.

Встановлено, що раціональною для нашої задачі є модель ExtraTrees (tuned). Вона поєднує дуже високу точність (RMSE ≈ 0.0097 кг/особу/день), Цю модель рекомендовано використовувати в модулі прогнозування «Organic Waste Energy Production System» для оперативного планування ресурсів та оцінювання сценаріїв виробництва біоенергії з органічних відходів.

Розроблене вікно користувача дозволяє без зайвих зусиль пройти повний цикл – дані → візуальна перевірка → порівняння моделей → вибір і тюнінг → тест → ручний прогноз → експорт. Нами прописано кроки користувача та очікувані результати. Для проєктного менеджера це означає не просто «ще одну програму», а реально корисний інструмент планування – із прозорими цифрами, зрозумілими графіками та відтворюваними результатами, які можна підкріпити звітом і використати в інших модулях СППР для біоенергетики.

Розроблені функціональні модулі СППР дають змогу забезпечити повний цикл роботи системи: від формування даних та навчання моделей до прогнозування у реальному часі. Вона дають можливість інтегрувати всі етапів у єдину структуру, що підвищує точність прогнозів та практичну цінність системи для планування виробництва біоенергії з органічних відходів.

Застосування СППР для умовного житлового масиву з 2000 домогосподарствами та 6200 мешканцями показало високу точність прогнозування (R^2 = 0.948). Отримані результати підтверджують можливість надійної оцінки параметрів утворення органічних відходів і формування обґрунтованих рекомендацій щодо масштабів виробництва біоенергії.

Нами обґрунтовано вимоги охорони праці та заходів безпеки у надзвичайних ситуаціях, що є невід’ємною складовою ефективної організації роботи.

Використання розробленої СППР дало змогу скоротити витрати річні витрати проєктних менеджерів на 270 тис. грн при витратах на розробку й підтримку 250 тис. грн, що забезпечило річний економічний ефект у 20 тис. грн та коефіцієнт віддачі 1.08 уже з першого року роботи. Це підтверджує доцільність подальшої експлуатації та масштабування запропонованої системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доповнене. Львів: Афіша, 2012. 350с.

2. Лехман С.Д., Рублев В.І., Рябцев Б.І. Запобігання аварійності і травматизму у сільському господарстві. К.: Урожай, 1993. 267 с.

3. Тригуба А., Маланчук О., Ратушний А., Паньків О., Коваль Л., Шолудько Р., Андрушків О. Адаптивно-ціннісний підхід до управління проектами розвитку громад та регіонів. Вісник Львівського національного університету природокористування. Серія Агроінженерні дослідження, 2023, 27, С. 113–126. doi: https://doi.org/10.31734/agroengineering2023.27.113

4. Тригуба А., Маланчук О., Тригуба І. Вплив сучасних інформаційних технологій на процеси ініціації та планування проєктів розвитку громад та регіонів. Вісник Львівського національного університету природокористування. Серія «Агроінженернія». 2024. № 28. С. 139-144.

5. Тригуба А., Маланчук О., Тригуба І., Мармуляк А., Демчина В., Андрушків О., & Олійник Р. (2024). Вплив сучасних інформаційних технологій на процеси ініціювання та планування проектів розвитку громад та регіонів. Вісник Львівського національного екологічного університету. Серія Агроінженерні дослідження , (28), 148–158. https://doi.org/10.31734/agroengineering2024.28.148

6. Aalborg University. EnergyPLAN Documentation (v16.3), 2024. https://energyplan.eu/about-energyplan/documentation/

7. Alavi, S., Zhuang, Q., Shrestha, R. K., et al. (2022). Machine learning for sustainable organic waste treatment and management. Journal of Cleaner Production, 362, 132335. https://doi.org/10.1016/j.jclepro.2022.132335

8. Andoni, M., et al. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 100, 143–174. https://doi.org/10.1016/j.rser.2018.10.014

9. Atabani, A. E., et al. (2022). Digital technologies for sustainable biomass utilization and bioenergy production: A review. *Renewable and Sustainable Energy Reviews*, 159, 112197. https://doi.org/10.1016/j.rser.2022.112197

10. AVEVA. PI Server – data infrastructure, 2024. https://www.aveva.com/en/products/aveva-pi-server/

11. AVEVA. PI System Overview, 2024. https://www.aveva.com/en/products/aveva-pi-system/

12. Balaman, S. Y. (2018). Decision-Making for Biomass-Based Production Chains: The Basic Concepts and Methodologies. *Elsevier*. https://doi.org/10.1016/B978-0-12-814278-3.00001-7

13. Balaman, S. Y. (2019). Basics of Decision-Making in Design and Management of Biomass-Based Production Chains. In Decision-Making for Biomass-Based Production Chains (pp. 143–183). Elsevier. https://doi.org/10.1016/B978-0-12-814278-3.00006-6

14. BioGrace Consortium. BioGrace GHG Calculation Tools (RED methodology), 2024. https://www.biograce.net/content/ghgcalculationtools/overview

15. Brown, T., Hörsch, J., Schlachtberger, D. PyPSA: Python for Power System Analysis. Journal of Open Research Software, 6(1), 2018, 4. https://doi.org/10.5334/jors.188

16. Energy Web Foundation. EW-Origin Documentation, 2024. https://energy-web-foundation-origin.readthedocs-hosted.com/

17. Energy Web. Documentation & Tech Stack (Digital Spine, Origin), 2024–2025. https://docs-launchpad.energyweb.org/

18. European Commission. (2020). The European Green Deal. Brussels. Retrieved from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52019DC0640

19. GreenDelta. openLCA – Open-source LCA software, 2025. https://www.openlca.org/

20. Hilpert, S., Kaldemeyer, C., et al. The Open Energy Modelling Framework (oemof). Energy Strategy Reviews, 22, 2018. https://doi.org/10.1016/j.esr.2018.03.003

21. Hilpert, S., Kaldemeyer, C., Krien, U., Günther, S., та ін. (2017). The Open Energy Modelling Framework (oemof) – A novel approach in energy system modelling. Preprints, June 2017. https://doi.org/10.20944/preprints201706.0093.v1

22. Huppmann, D., et al. The MESSAGEix Integrated Assessment Model and the ix modeling platform. Environmental Modelling & Software, 112, 2019, 143–156. https://doi.org/10.1016/j.envsoft.2018.11.012

23. IEA-ETSAP. Documentation for the TIMES Model (Parts I–III), 2016. https://iea-etsap.org/index.php/documentation

24. IIASA. BeWhere: Spatially explicit system optimization tool, 2025. https://iiasa.ac.at/models-tools-data/bewhere-model-spatially-explicit-system-optimization-tool

25. Inductive Automation. Ignition SCADA Platform (Docs 8.1), 2024–2025. https://inductiveautomation.com/scada-software/

26. International Energy Agency. (2023). Renewables 2023: Analysis and forecast to 2028. Paris: IEA. Retrieved from https://www.iea.org/reports/renewables-2023

27. Modern Information System Architectures. Retrieved from https://www.niceideas.ch/roller2/badtrash/entry/modern-information-system-architectures

28. Mukamwi, L., Kuppens, T., Van Dael, M., Van Passel, S. (2023). Data availability and integration for circular bioeconomy value chain assessment: A review. Resources, Conservation and Recycling, 191, 106891. https://doi.org/10.1016/j.resconrec.2023.106891

29. National Renewable Energy Laboratory (NREL). (2023). U.S. Biomass Resource Maps. Retrieved from: https://www.nrel.gov/gis/biomass.html

30. Natural Resources Canada. RETScreen Clean Energy Management Software, 2025. https://natural-resources.canada.ca/maps-tools-publications/tools-applications/retscreen

31. Nwachukwu, C. M., et al. BeWhere Sweden (BWS) – techno-economic supply chain model. Applied Energy, 326, 2022, 120131. https://doi.org/10.1016/j.apenergy.2022.120131

32. Onu, P., Mbohwa, C., & Pradhan, A. (2023). Artificial intelligence-based IoT-enabled biogas production. In Proceedings of the 2023 International Conference on Control, Automation and Diagnosis (ICCAD) (Rome, Italy). IEEE. https://doi.org/10.1109/ICCAD57653.2023.10152349

33. OPC Foundation. OPC UA Specifications (Part 1–…), 2024–2025. https://reference.opcfoundation.org/

34. Pfenninger, S., Pickering, B. Calliope: a multi-scale energy systems modelling framework. Journal of Open Source Software, 3(29), 2018, 825. https://doi.org/10.21105/joss.00825

35. Pilehvar, A., & Shahangian, S. (2021). Digital twins in bioenergy systems: Applications and perspectives. *Energy Reports*, 7, 4825–4836. https://doi.org/10.1016/j.egyr.2021.07.065

36. S2Biom Consortium. Supplying Non-food Biomass in Europe (S2Biom): Toolset & Datasets, 2019–2025. https://www.s2biom.eu/

37. Scarlat, N., Dallemand, J.-F., & Fahl, F. (2018). Biogas: Developments and perspectives in Europe. Renewable Energy, 129, 457–472. https://doi.org/10.1016/j.renene.2018.03.006

38. Soon, J.-J., & Ahmad, S.-A. (2015). Willingly or grudgingly? A meta-analysis on the willingness-to-pay for renewable energy use. Renewable and Sustainable Energy Reviews, 44, 877–887. https://doi.org/10.1016/j.rser.2015.01.041

39. Stockholm Environment Institute. LEAP – Low Emissions Analysis Platform, 2024. https://leap.sei.org

40. Sultana, R., Reza, M. N., Nahiduzzaman, M., Rahman, M. A. (2023). BDWaste: A benchmark dataset for organic waste classification. Data in Brief, 47, 108995. https://doi.org/10.1016/j.dib.2023.108995

41. Tryhuba, I., Tryhuba, A., Hutsol, T., Andrushkiv, O.; Faichuk, O., Slavina, N. European Green Deal: Substantiation of the Rational Configuration of the Bioenergy Production System from Organic Waste. Energies, 2024, 17(17), 4513

42. Tryhuba, I.; Tryhuba, A.; Hutsol, T.; Cieszewska, A.; Andrushkiv, O.; Glowacki, S.; Bryś, A.; Slobodian, S.; Tulej, W.; Sojak, M. Prediction of Biogas Production Volumes from Household Organic Waste Based on Machine Learning. Energies 2024, 17, 1786. https://doi.org/10.3390/en17071786

43. Tryhuba, I.; Tryhuba, A.; Hutsol, T.; Lopushniak, V.; Cieszewska, A.; Andrushkiv, O.; Barabasz, W.; Pikulicka, A.; Kowalczyk, Z.; Vasyuk, V. European Green Deal: Justification of the Relationships between the Functional Indicators of Bioenergy Production Systems Using Organic Residential Waste Based on the Analysis of the State of Theory and Practice. Energies 2024, 17, 1461. https://doi.org/10.3390/en17061461

44. U.S. Department of Energy. (2022). 2016 Billion-Ton Report: Advancing Domestic Resources for a Thriving Bioeconomy. Updated data. Retrieved from: https://www.energy.gov/eere/bioenergy/2016-billion-ton-report

45. U.S. Department of Energy. (2023). BioEnergy Knowledge Discovery Framework (KDF). Retrieved from: https://bioenergykdf.net

46. U.S. DOE / Argonne National Laboratory. GREET Model (Greenhouse gases, Regulated Emissions, and Energy use in Technologies), 2024. https://www.energy.gov/eere/greet

47. UL Solutions. HOMER Pro Microgrid Software, 2025. https://www.homerenergy.com/products/pro/index.html

48. WUR/S2Biom. S2Biom Toolset Leaflet & Technical Description, 2016–2017. https://s2biom.wenr.wur.nl/doc/D10.15_S2Biom_Leaflet_Toolset.pdf

Додатки

Додаток А.

Фрагмент коду для визначення раціональної моделі прогнозування обсягів утворення органічних відходів у домогосподарствах

```
import os
import joblib
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, KFold, cross_validate, RandomizedSearchCV
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, make_scorer

# Базові моделі (≥10):
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet, HuberRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import (
   RandomForestRegressor, ExtraTreesRegressor, GradientBoostingRegressor,
   AdaBoostRegressor, StackingRegressor
)
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor

# Для важливостей/пояснень
from sklearn.inspection import permutation_importance

# --- 0. Налаштування ---
RANDOM_STATE = 2025
np.random.seed(RANDOM_STATE)

DATA_PATH = r"F:\Dipl_2025\Mar_2025\Тригуба\Program\initial_dataset_900.csv"
MODEL_DIR = r"F:\Dipl_2025\Mar_2025\Тригуба\Program"
BEST_MODEL_PATH = os.path.join(MODEL_DIR, "waste_forecast_best.joblib")
REPORT_PATH = os.path.join(MODEL_DIR, "model_report.csv")

# --- Завантаження даних ---
df = pd.read_csv(DATA_PATH, encoding="utf-8-sig")
print("Розмірність датасету:", df.shape)
display(df.head())

# --- Валідація та легкий data cleaning ---
# Перевіримо пропуски та типи
display(df.info())
display(df.isna().sum())
```

```
# Перейменуємо ціль у зручну коротку назву
target_col = "Average daily volumes of organic waste per inhabitant, kg/person"
df = df.rename(columns={target_col: "target_kg_per_person"})

# Переконаємось, що числові стовпці дійсно числові
num_cols = [
   "Residential Area, m²",
   "Number of households, units",
   "Number of inhabitants, persons"
]
df[num_cols] = df[num_cols].apply(pd.to_numeric, errors="coerce")

# Декілька базових логічних перевірок
assert df["target_kg_per_person"].between(0.10, 0.60).all(), "Ціль повинна бути в межах [0.10; 0.60] кг/особу."

# --- Інженерія ознак (похідні фічі, що мають сенс) ---
# щільність населення: особи на 1 домогосподарство
df["persons_per_household"] = df["Number of inhabitants, persons"] / df["Number of households, units"]

# площа на одне домогосподарство
df["area_per_household"] = df["Residential Area, m²"] / df["Number of households, units"]

# бінарні індикатори для типів поселення та рівня доходів (але закодуємо через OneHotEncoder у пайплайні)
# збережемо імена колонок
cat_cols = [
   "Settlement Type",
   "Income level of the population",
   "Organic Waste Source"
]

num_cols_ext = num_cols + ["persons_per_household", "area_per_household"]

# Візуалізація (без seaborn; тільки matplotlib) ---
plt.figure(figsize=(8,5))
df["target_kg_per_person"].hist(bins=30)
plt.title("Розподіл цільової змінної (кг/особу)")
plt.xlabel("кг/особу")
plt.ylabel("кількість записів")
plt.show()

plt.figure(figsize=(8,5))
df["Settlement Type"].value_counts().plot(kind="bar")
plt.title("Частоти: тип населеного пункту")
plt.xlabel("Тип")
plt.ylabel("К-сть")
plt.show()
```

```
plt.figure(figsize=(8,5))
df["Income level of the population"].value_counts().plot(kind="bar")
plt.title("Частоти: рівень доходів")
plt.xlabel("Рівень")
plt.ylabel("К-сть")
plt.show()

plt.figure(figsize=(8,5))
plt.scatter(df["persons_per_household"], df["target_kg_per_person"], alpha=0.5)
plt.title("Залежність: осіб на домогосподарство vs кг/особу")
plt.xlabel("Осіб/домогосподарство")
plt.ylabel("кг/особу")
plt.show()

# --- Train/Test split ---
X = df[cat_cols + num_cols_ext]
y = df["target_kg_per_person"]

X_train, X_test, y_train, y_test = train_test_split(
   X, y, test_size=0.2, random_state=RANDOM_STATE, stratify=df["Settlement Type"]
)

…


# --- Важливості ознак (через permutation importance) ---
# обчислюємо вже на препроцесених фічах
perm = permutation_importance(final_pipe, X_test, y_test, n_repeats=10,
random_state=RANDOM_STATE, n_jobs=-1)
# назв після ColumnTransformer
feature_names = []
# дістаємо імена ознак після OneHotEncoder і StandardScaler
cat_feature_names =
final_pipe.named_steps["prep"].named_transformers_["cat"].get_feature_names_out(cat_cols)
num_feature_names = np.array(num_cols_ext, dtype=object)
feature_names = np.concatenate([cat_feature_names, num_feature_names])

imp_mean = perm.importances_mean
idx = np.argsort(imp_mean)[::-1][:15]  # топ-15 для наочності

plt.figure(figsize=(8,6))
plt.barh(range(len(idx)), imp_mean[idx][::-1])
plt.yticks(range(len(idx)), feature_names[idx][::-1])
plt.title("Permutation importance (топ-15 ознак)")
plt.xlabel("Середня зміна помилки")
plt.tight_layout()
plt.show()
```

```
# --- Збереження моделі та звіту ---
joblib.dump(final_pipe, BEST_MODEL_PATH)

summary = {
   "best_model_name": [best_name],
   "cv_rmse": [final_cv_rmse],
   "test_rmse": [rmse],
   "test_mae": [mae],
   "test_r2": [r2],
   "n_train": [len(X_train)],
   "n_test": [len(X_test)]
}
pd.DataFrame(summary).to_csv(REPORT_PATH, index=False, encoding="utf-8-sig")

print("Модель збережено:", BEST_MODEL_PATH)
print("Звіт збережено:", REPORT_PATH)
```

Додаток Б.

Фрагмент коду створеного мультивіконного застосунку «Organic Waste Forecast Studio»

```
import os
import random
import tkinter as tk
from tkinter import ttk, filedialog, messagebox

import numpy as np
import pandas as pd
import matplotlib
matplotlib.use("TkAgg")
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

from sklearn.model_selection import train_test_split, KFold, cross_validate, RandomizedSearchCV
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, make_scorer
from sklearn.inspection import permutation_importance

# Моделі
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet, HuberRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor,
GradientBoostingRegressor, AdaBoostRegressor
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor

import joblib

# ---------------------- Допоміжні функції генерації ----------------------
SETTLEMENTS = ["city", "village", "township"]
P_SETTLEMENTS = [0.50, 0.30, 0.20]

INCOME = ["low", "medium", "high"]
P_INCOME = [0.35, 0.45, 0.20]

SOURCES = ["FW", "YW", "FYW", "MOW"]
SOURCE_LABEL = {
   "FW": "food waste (FW)",
   "YW": "yard waste (YW)",
   "FYW": "mixed food and yard waste (FYW)",
```

```
    "MOW": "mixed organic waste (MOW)",
}

def sample_residential_area(settlement: str) -> float:
    if settlement == "city":
        return float(np.random.uniform(80_000, 500_000))
    if settlement == "township":
        return float(np.random.uniform(20_000, 250_000))
    return float(np.random.uniform(10_000, 150_000))

def area_per_household(settlement: str) -> float:
    if settlement == "city":
        return float(np.random.uniform(50, 70))
    if settlement == "township":
        return float(np.random.uniform(70, 100))
    return float(np.random.uniform(90, 140))

def household_size(income: str, settlement: str) -> float:
    base = {
        "low": np.random.uniform(3.5, 4.5),
        "medium": np.random.uniform(2.8, 3.5),
        "high": np.random.uniform(2.0, 2.8),
    }[income]
    if settlement == "village":
        base += np.random.uniform(0.2, 0.5)
    if settlement == "city":
        base -= np.random.uniform(0.0, 0.2)
    return float(base)

def conditional_source_probs(settlement: str):
    if settlement == "city":
        return [0.45, 0.10, 0.30, 0.15]
    if settlement == "township":
        return [0.35, 0.20, 0.25, 0.20]
    return [0.25, 0.35, 0.20, 0.20]

def avg_daily_waste_kg_per_person(settlement, income, source, persons_per_household):
    base = 0.25
    if settlement == "city":
        base += 0.05
    elif settlement == "village":
        base -= 0.03
    if income == "low":
        base += 0.05
    elif income == "medium":
        base += 0.02
    elif income == "high":
        base -= 0.02
    if source == "FW":
```

```
        base += 0.05
    elif source == "YW":
        base += 0.02
        if settlement == "village":
            base += 0.03
    elif source == "FYW":
        base += 0.06
    elif source == "MOW":
        base += 0.03
    if persons_per_household > 3:
        base += 0.01 * (persons_per_household - 3)
    noise = float(np.random.normal(0.0, 0.01))
    return float(np.clip(base + noise, 0.10, 0.60))

# ---------------------------- Головний застосунок ----------------------------
class WasteForecastApp(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Organic Waste Forecast Studio")
        self.geometry("1180x780")
        self.minsize(1100, 700)

        # Стиль
        s = ttk.Style(self)
        try:
            s.theme_use("clam")
        except:
            pass
        s.configure("TButton", font=("Segoe UI", 11), padding=8)
        s.configure("TLabel",  font=("Segoe UI", 11))
        s.configure("TLabelframe.Label", font=("Segoe UI", 12, "bold"))
        s.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))

        # Дані/модель
        self.df = None
        self.final_pipe = None
        self.best_name = None
        self.cv_best = None
        self.train_parts = None  # (X_train, X_test, y_train, y_test)
        self.cat_cols = ["Settlement Type", "Income level of the population", "Organic Waste Source"]
        self.num_cols = ["Residential Area, m²", "Number of households, units", "Number of
inhabitants, persons"]
        self.num_cols_ext = self.num_cols + ["persons_per_household", "area_per_household"]

        # Верхній заголовок
        header = ttk.Label(self, text="Organic Waste Energy Production System — ML Studio",
                           font=("Segoe UI", 16, "bold"))
        header.pack(pady=(10, 0))
```

```
        # Тетрадь
        self.nb = ttk.Notebook(self)
        self.nb.pack(fill="both", expand=True, padx=10, pady=10)

        self._build_tab_generate()
        self._build_tab_viz()
        self._build_tab_train()
        self._build_tab_predict()
        self._build_tab_export()
```

…

```
    # ------------------------ Tab 3: Навчання моделей ------------------------
    def _build_tab_train(self):
        tab = ttk.Frame(self.nb)
        self.nb.add(tab, text="3) Навчання моделей")

        top = ttk.Frame(tab)
        top.pack(fill="x", padx=10, pady=6)

        ttk.Button(top, text="Запустити крос-валідацію (13 моделей)",
command=self._run_cv).pack(side="left", padx=5)
        ttk.Button(top, text="Тюнінг + вибір фінальної",
command=self._tune_and_select).pack(side="left", padx=5)
        ttk.Button(top, text="Оцінити на TEST", command=self._eval_test).pack(side="left", padx=5)
        ttk.Label(top, text="(спершу CV, далі тюнінг, потім TEST)").pack(side="left", padx=10)

        # графік
        self.fig_cv = plt.Figure(figsize=(7,4), dpi=100)
        self.canvas_cv = FigureCanvasTkAgg(self.fig_cv, master=tab)
        self.canvas_cv.get_tk_widget().pack(fill="both", expand=True, padx=10, pady=6)

        # таблиця метрик
        self.tree_cv = ttk.Treeview(tab, columns=(), show="headings", height=10)
        self.tree_cv.pack(fill="both", expand=True, padx=10, pady=(0,10))

    def _prep_xy(self):
        self._ensure_df()
        df = self.df.copy()
        target_col = "Average daily volumes of organic waste per inhabitant, kg/person"
        df = df.rename(columns={target_col: "target"})
        # похідні
        df["persons_per_household"] = df["Number of inhabitants, persons"] / df["Number of
households, units"]
        df["area_per_household"] = df["Residential Area, m²"] / df["Number of households, units"]

        X = df[self.cat_cols + self.num_cols_ext]
        y = df["target"]
```

```
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=2025, stratify=df["Settlement Type"]
    )

    categorical_transformer = OneHotEncoder(handle_unknown="ignore", sparse_output=False)
    numeric_transformer = StandardScaler()
    preprocess = ColumnTransformer(
        transformers=[
            ("cat", categorical_transformer, self.cat_cols),
            ("num", numeric_transformer, self.num_cols_ext),
        ],
        remainder="drop"
    )
    return (X_train, X_test, y_train, y_test), preprocess

def _run_cv(self):
    try:
        self.train_parts, preprocess = self._prep_xy()
        X_train, X_test, y_train, y_test = self.train_parts

        models = {
            "LinearRegression": LinearRegression(),
            "Ridge": Ridge(random_state=2025),
            "Lasso": Lasso(random_state=2025),
            "ElasticNet": ElasticNet(random_state=2025),
            "HuberRegressor": HuberRegressor(),
            "KNN": KNeighborsRegressor(),
            "DecisionTree": DecisionTreeRegressor(random_state=2025),
            "RandomForest": RandomForestRegressor(random_state=2025),
            "ExtraTrees": ExtraTreesRegressor(random_state=2025),
            "GradientBoosting": GradientBoostingRegressor(random_state=2025),
            "AdaBoost": AdaBoostRegressor(random_state=2025, loss="square"),
            "SVR": SVR(),
            "MLPRegressor": MLPRegressor(random_state=2025, max_iter=2000),
        }

        cv = KFold(n_splits=5, shuffle=True, random_state=2025)
        scoring = {
            "rmse": make_scorer(lambda yt, yp: mean_squared_error(yt, yp, squared=False)),
            "mae": "neg_mean_absolute_error",
            "r2": "r2"
        }

        rows = []
        best_score = np.inf
        best_name = None
        best_pipe = None
```

```
        for name, model in models.items():
            pipe = Pipeline([("prep", preprocess), ("model", model)])
            cv_res = cross_validate(pipe, X_train, y_train, cv=cv, scoring=scoring, n_jobs=-1)
            rmse_mean = np.mean(cv_res["test_rmse"])
            mae_mean = -np.mean(cv_res["test_mae"])
            r2_mean = np.mean(cv_res["test_r2"])
            rows.append({"model": name, "cv_rmse": rmse_mean, "cv_mae": mae_mean, "cv_r2":
r2_mean})
            if rmse_mean < best_score:
                best_score = rmse_mean
                best_name = name
                best_pipe = pipe

        res_df = pd.DataFrame(rows).sort_values("cv_rmse")
        self.cv_best = (best_name, best_score, best_pipe)

        # графік
        self.fig_cv.clf()
        ax = self.fig_cv.add_subplot(111)
        ax.plot(res_df["model"], res_df["cv_rmse"], marker="o")
        ax.set_title("Порівняння моделей за CV-RMSE (менше — краще)")
        ax.set_xlabel("Модель"); ax.set_ylabel("CV-RMSE")
        ax.tick_params(axis='x', rotation=45)
        self.canvas_cv.draw()

        self._fill_tree(self.tree_cv, res_df.round(6))
        messagebox.showinfo("CV завершено", f"Лідер за CV-RMSE: {best_name}
({best_score:.4f})")
      except Exception as e:
        messagebox.showerror("Крос-валідація", str(e))

    def _tune_and_select(self):
      try:
        if self.train_parts is None:
            raise RuntimeError("Спершу виконайте крос-валідацію.")
        (X_train, X_test, y_train, y_test), preprocess = self._prep_xy()

        # Тюнінг ключових
        results = {}
        # ExtraTrees
        et = ExtraTreesRegressor(random_state=2025)
        et_params = {
            "model__n_estimators": [200, 400, 600],
            "model__max_depth": [None, 8, 12, 20],
            "model__min_samples_split": [2, 5, 10],
            "model__min_samples_leaf": [1, 2, 4],
        }
        et_pipe = Pipeline([("prep", preprocess), ("model", et)])
        et_search = RandomizedSearchCV(et_pipe, et_params, n_iter=25, cv=5,
```

```
                                scoring=make_scorer(lambda yt, yp: mean_squared_error(yt, yp,
squared=False),
                                                    greater_is_better=False),
                                n_jobs=-1, random_state=2025)
        et_search.fit(X_train, y_train)
        et_cv = -et_search.best_score_
        results["ExtraTrees_tuned"] = (et_search.best_estimator_, et_cv)

…

    # ------------------------ Утіліти ------------------------
    def _fill_tree(self, tree: ttk.Treeview, df: pd.DataFrame):
        # Очистити
        tree.delete(*tree.get_children())
        # Стовпці
        cols = list(df.columns.astype(str))
        tree["columns"] = cols
        for c in cols:
            tree.heading(c, text=c)
            tree.column(c, width=max(120, int(800/len(cols))), anchor="center")
        # Рядки
        for _, row in df.iterrows():
            tree.insert("", "end", values=[row[c] for c in cols])

    def _append_tree(self, tree: ttk.Treeview, df: pd.DataFrame, title_row="---"):
        # вставити розділювач
        tree.insert("", "end", values=[title_row] + [""]*(len(tree["columns"])-1))
        # додати таблицю (пристосувати колонки якщо відрізняються)
        # якщо структура інша — створимо нові колонки під кінець
        if list(df.columns.astype(str)) != list(tree["columns"]):
            cols = list(df.columns.astype(str))
            tree["columns"] = cols
            for c in cols:
                tree.heading(c, text=c)
                tree.column(c, width=max(120, int(800/len(cols))), anchor="center")
            tree.insert("", "end", values=[""]*len(cols))
        for _, row in df.iterrows():
            tree.insert("", "end", values=[row[c] for c in df.columns])

# ----------------------------- Запуск -----------------------------
if __name__ == "__main__":
    # Для гарних шрифтів на Windows
    plt.rcParams.update({"font.size": 10})
    app = WasteForecastApp()
    app.mainloop()
```