Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі»

Виконав: студент 2 курсу, групи 1ІСТ-23м
спеціальності 126 – «Інформаційні системи та технології»
_____ Артем МАРУЩАК
Керівник: д.т.н., проф. каф. САІТ
_____ Олександр МОКІН
«05» 12 2024 р.
Опонент: д.т.н., проф. каф. КН
_____ Ярослав ІВАНЧУК
«03» 12 2024 р.

Допущено до захисту

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

«06» 12 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – другий (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 126 Інформаційні системи та технології
Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

«17» 09 2024 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Марущаку Артему Володимировичу

1.Тема роботи: «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі»,
керівник роботи: Олександр МОКІН, д.т.н., проф. каф. САІТ,
затверджені наказом ВНТУ від «17» 09 2024 року №310
2. Строк подання студентом роботи «01» 12 2024 року
3. Вихідні дані до роботи:
1) Kaggle Dataset «Crop Production» https://www.kaggle.com/datasets/imtkaggleteam/crop-production
2) Kaggle Dataset «Country Mapping - ISO, Continent, Region» https://www.kaggle.com/datasets/andradaolteanu/country-mapping-iso-continent-region

4. Зміст текстової частини:
- загальна характеристика об’єкту досліджень;
- налаштування технологій для розв’язання поставленої задачі;
- оброблення результатів спостережень та візуалізація даних у середовищі kaggle;
- розроблення інформаційної технології узагальнення даних вирощування сільськогосподарських культур у Європі;
- економічна частина.

5. Перелік ілюстративного матеріалу:
- теплова карта врожайності;
- графік поширення зернових культур;

– криві навчання для моделі передбачення даних;
– графік важливості ознак;
– алгоритм роботи інформаційної технології;

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
5	Олександр ЛЕСЬКО, к. е. н., проф. каф. ЕПВМ		
		10.11.24	30.11.24

7. Дата видачі завдання «17» 09 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітк
		початок	закінчення	
1	Загальна характеристика об'єкту дослідження	20.09	30.09	вик
2	Налаштування технологій для розв'язання поставленої задачі	01.10	15.10	вик.
3	Оброблення результатів спостережень та візуалізація даних у середовищі Kaggle	16.10	31.10	вик.
4	Розроблення інформаційної технології узагальнення даних вирощування сільськогосподарських культур у Європі	01.11	15.11	вик.
5	Економічна частина	16.11	25.11	вик
6	Оформлення матеріалів до захисту МКР	25.11	30.11	вик

Студент Артем МАРУЩАК

Керівник роботи Олександр МОКІН

АНОТАЦІЯ

УДК 004.85+631

Марущак А.В. Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2024. 125 с.

На укр. мові. Бібліогр.: 47 назв; рис.: 54; табл.: 13.

В магістерській кваліфікаційній роботі було здійснено розробку інформаційної технології аналізу врожайності сільськогосподарських культур на території Європи в період з 1961 по 2019 роки. Вхідний набір даних для виконання поставленої задачі було об’єднано з географічною прив’язкою. Проведено тестування розробленої системи на платформі Kaggle з використанням мови програмування Python. Виконано візуалізацію системного аналізу даних на прикладі побудови графіків, які показують критичні аспекти врожайності в країнах Європи. За допомогою методів машинного навчання було проведено навчання моделей на вхідних даних щодо врожайності сільськогосподарських культур, отримано позитивний результат.

Ілюстративна частина складається з 10 плакатів із розвідувальним аналізом та результатом тестування моделей.

У розділі економічної частини розглянуто питання про доцільність розробки та впровадження інформаційної технології передбачення врожайності сільськогосподарських культур на території Європи.

Ключові слова: Python, розвідувальний аналіз, машинне навчання, сільськогосподарська сфера, врожайність.

ABSTRACT

Marushchak A.V. Information technology of generalization of data on crop production in Europe. Master's qualification work in specialty 126 - Information systems and technologies, educational and professional program - Information technologies of data and image analysis. Vinnytsia: VNTU, 2024. 125 c.

In Ukrainian. Bibliography: 47 titles; figs. 54; tables: 13.

During the master's qualification work, the development of information technology for analyzing crop yields in Europe in the period from 1961 to 2019 was carried out. The input data set for the task was combined with a geographical reference. The developed system was tested on the Kaggle platform using the Python programming language. Visualization of the systematic data analysis was performed on the example of building graphs showing critical aspects of yields in European countries. Using machine learning methods, models were trained on the input data on crop yields, and a positive result was obtained.

The illustrative part consists of 10 posters with intelligence analysis and model testing results.

The economic section considers the feasibility of developing and implementing information technology for predicting crop yields in Europe.

Keywords: Python, intelligence analysis, machine learning, agricultural sector, yield.

ЗМІСТ

ВСТУП

Актуальність теми. Передбачення врожайності сільськогосподарських культур є ключовим інструментом для забезпечення продовольчої безпеки та стабільності ринків. Такі дані дозволяють фермерам та аграрним підприємствам планувати свої ресурси та операції, знижуючи ризики та оптимізуючи витрати. Завдяки точним прогнозам, можна уникнути надвиробництва або дефіциту. Сучасні технології, такі як супутниковий моніторинг та машинне навчання, значно підвищують точність прогнозів, сприяючи ефективнішому управлінню аграрним сектором.

У цьому контексті, дана робота присвячена дослідженню різних підходів для передбачення врожайності сільськогосподарських культур, їх переваг та недоліків.

Мета і завдання роботи. Метою дослідження є підвищення точності передбачення врожайності сільськогосподарських культур за рахунок використання методів машинного навчання.

Розроблення моделі для підвищення точності передбачення врожаю передбачає виконання таких завдань:

1. Аналіз важливості передбачення врожаю сільськогосподарської продукції у країнах Європи.
2. Вибір технологій та методів для розв’язування поставленої задачі.
3. Створення та налаштування нового проекту в середовищі роботи Kaggle.
4. Розроблення технології передбачення врожайності сільськогосподарських культур.

Об’єктом дослідження магістерської кваліфікаційної роботи є процес підвищення точності передбачення врожайності сільськогосподарських культур у Європі.

Предметом дослідження магістерської кваліфікаційної роботи є методи та інформаційна технологія аналізу врожайності, а також їх застосування для передбачення врожаю сільськогосподарських культур країн на території Європи.

Методи дослідження. У дослідженнях використовуються методи розвідувального аналізу, методи побудови моделей машинного навчання.

Новизна одержаних результатів полягає в подальшому розвитку інформаційної технології передбачення врожайності сільськогосподарських культур за рахунок удосконалення методів машинного навчання та розвідувального аналізу, що дозволяє підвищити точність цього передбачення.

Практичне значення. Отримані дані важливі та широко використовуються в аграрній сфері. Під час планування наступного врожаю, а також під час вирощування, керівництво заздалегідь може спланувати розподіл посівних сільськогосподарських культур з урахуванням отриманих даних в дослідженні. Такий підхід направлений для оптимізації використання ресурсів й отримання більших показників врожайності.

Апробація результатів магістерської кваліфікаційної роботи. Результати роботи подані на всеукраїнську науково-практичну інтернет-конференцію «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.).

Публікації результатів магістерської кваліфікаційної роботи. Опубліковано тези на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.) [1].

1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕНЬ

1.1 Аналіз стану методів сучасних технолой передбачення майбутнього врожаю сільськогосподарських культур

Рослини, які вважаються зерновими культурами – це рослини, що дають дрібне, тверде сухе насіння або плоди, які людина чи її домашні тварини споживають у їжу або переробляють для харчових чи промислових цілей.

Сільське господарство – це практика вирощування природних ресурсів для підтримки життя людини та забезпечення економічної вигоди. В цьому процесі поєднується креативність, уява та навички, пов'язані з посадкою сільськогосподарських культур і вирощуванням тварин із використанням сучасних методів виробництва та нових технологій.

Сільське господарство також є бізнесом, який забезпечує світову економіку товарами: основними товарами, що використовуються в торгівлі, такими як зерно, худоба, молочні продукти, клітковина та сировина для палива [2].

Людство залежить від сільського господарства не лише для забезпечення їжі, але й велику частину складає об'єм робочих місць та значення в глобальній економіці. Понад один мільярд людей, майже 30% усієї світової робочої сили, зайняті у сільському господарстві.

Пшениця, кукурудза та рис були одними з перших культур, які почали вирощувати. Пшениця є однією з перших відомих сільськогосподарських культур.

Виробництво продуктів харчування та сільськогосподарських культур безпосередньо визначає чисельність населення. Якщо не вистачає їжі, популяція не може рости. З удосконаленням сільського господарства населення зростає. У 1700-х роках населення Англії становило приблизно 5,7 мільйона і воно залишалося на цьому рівні протягом майже 2000 років [3]. Після винаходу нових

методів землеробства в 1750-х роках населення Англії майже потроїлося до 16,6 мільйонів лише за 100 років.

Демографічний вибух у Великій Британії та в усьому світі пояснюється покращенням виробництва сільськогосподарських культур.

Враховуючи важливість орного землеробства, дослідження способів передбачення врожайності сільськогосподарських культур мають важливе значення для подальшого стрімкого розвитку сфери.

Передбачення майбутнього врожаю включає спостереження та оцінку стану посівів зернових і інших сільськогосподарських культур, а також розрахунок необхідних вкладень і потенційного прибутку.

Ці моніторинги особливо важливі для інвесторів, які розглядають можливість вкласти кошти в агробізнес. Звіти агрохолдингів та інших сільськогосподарських підприємств часто містять дані про врожайність або пошкодження врожаю комахами, які можуть бути неточними. Причини може бути безліч від навмисного викривлення реальних даних до помилок, спричинених некомпетентністю керівництва. Незалежні моніторингові заходи допомагають об'єктивно оцінити можливі інвестиційні ризики і прогнозувати прибуток [4].

Моніторинг потрібен також самим аграріям, щоб якомога точніше спрогнозувати майбутній урожай, і в цілому – контролювати бізнес-процеси.

Більше половини зернових культур, що вирощуються в Європі, становить пшениця. Решта складаються з кукурудзи та ячменю, кожна з яких становить приблизно одну третину. Остання третина включає зернові культури, вирощені в менших кількостях, такі як жито, овес та полба. Зернові культури на території Європи переважно використовуються на корм тваринам, майже 70% загального врожаю, в той час як на споживання людиною залишається 30% і лише 3-5% використовується для виробництва біопалива [4]. На рисунку 1.1 показано динаміку врожайності пшениці на території України в період з 2018 по 2022 роки.

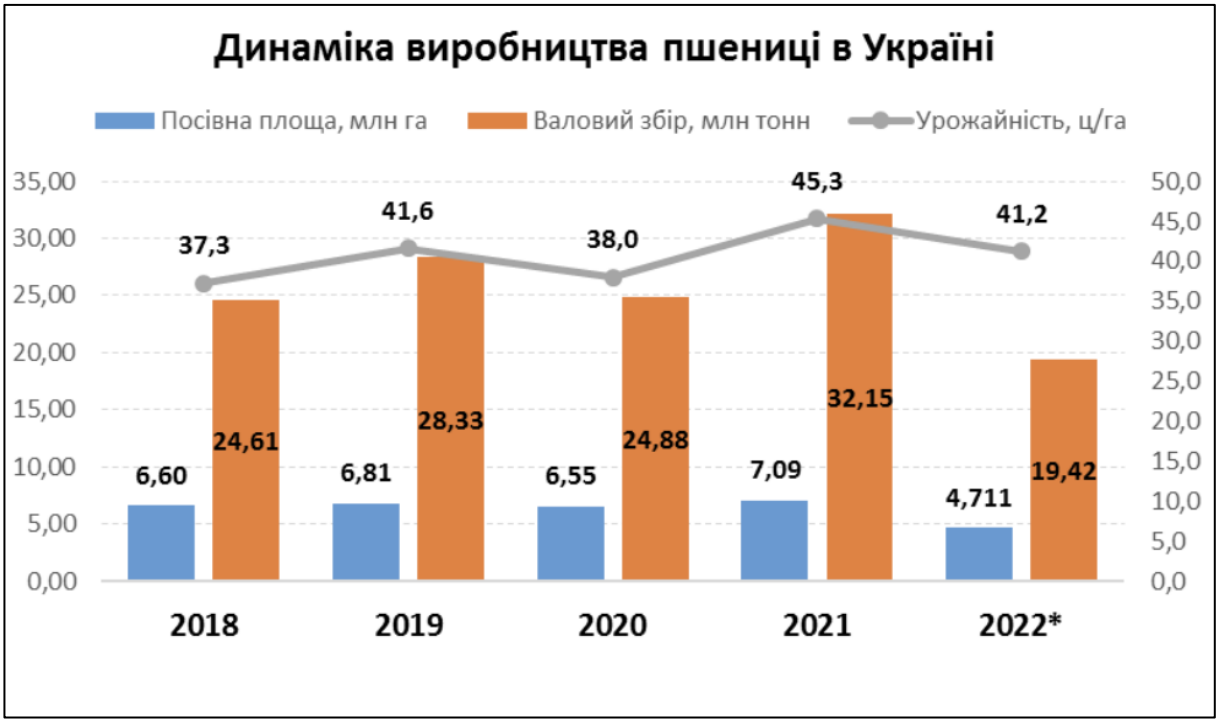


Рисунок 1.1 – Врожайність пшениці на території України [5]

Аналізуючи такі графіки, можна використати важливу та ключову інформацію для розробки моделей, які будуть прогнозувати майбутню врожайність пшениці на території України.

Варто зазначити, що соняшник також відіграє ключову роль у сільському господарстві, забезпечуючи високоякісну рослинну олію та корми для тварин, що сприяє економічній стабільності та розвитку аграрних регіонів. Його вирощування покращує структуру ґрунту і підвищує його родючість, а також сприяє розвитку біодизеля – як альтернативного джерела енергії. Це робить соняшник важливою культурою для харчової промисловості, тваринництва, економіки та екологічної стійкості [6].

Україна посідає перше місце у світі по об’єму вирощування даної зернової культури. На рисунку 1.2 показано карту об’єму зібраного врожаю соняшника у світі за 2021 рік.

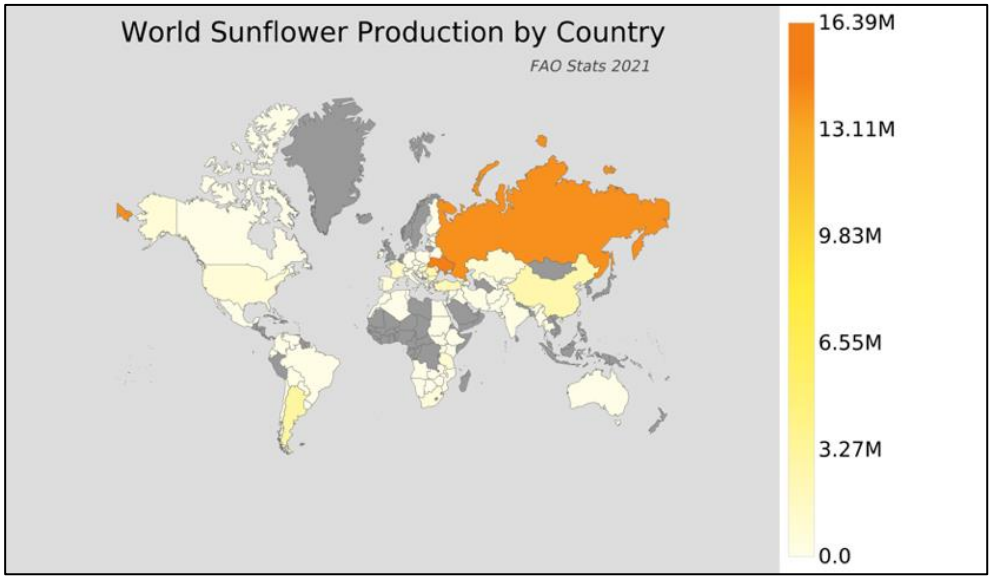


Рисунок 1.2 – Карта вирощування соняшника у світі [7]

Як можна побачити з рисунку, вирощування соняшника для України має важливу економічну позицію, адже більшість країн мають набагато менший валовий об’єм врожаю за один календарний рік. Отже, відбувається експорт як сировини, так і готової продукції, що позитивно впливає на фінансове становище країни. Завжди варто аналізувати вартість попередню та мати потужний інструмент передбачення ціни на певну культуру. На рисунку 1.3 показано інфографіку щомісячних змін вартості пшениці на світовому ринку в період 2020-2021 року.

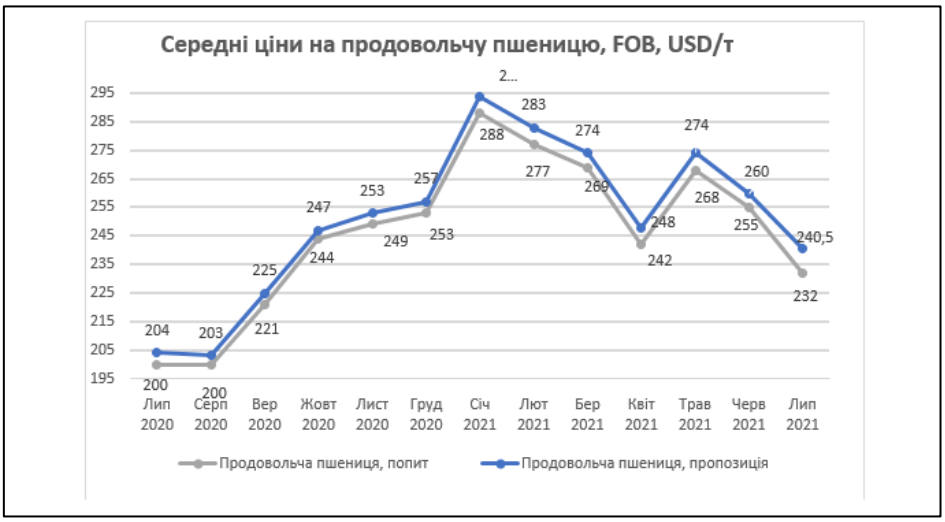


Рисунок 1.3 –Зміна вартості пшениці [8]

Після аналізу даних графіків було отримано результат, що врожайність зернових культури найбільше впливає на її майбутню вартість на ринку. Отже, для конкурентоспроможності, аграрії повинні користуватися технологіями для передбачення врожайності.

Прогнозування врожайності передбачає врахування низки факторів, таких як погодні умови, якість ґрунту, історичні дані та агротехнічні практики. Це дозволяє оцінити кількість сільськогосподарської продукції, яку можна отримати на певній території за визначений період. Сучасні алгоритми машинного навчання, зокрема методи регресії та нейронні мережі, значно підвищують точність прогнозів, що дозволяє фахівцям і аграрним компаніям більш точно передбачати врожайність [8].

Вплив таких технологій на сільське господарство є суттєвим. Точні прогнози врожайності допомагають фермерам ухвалювати обґрунтовані рішення щодо вибору культур для посіву, а також раціонального розподілу ресурсів. Крім того, це надає важливу інформацію для розробки політики, що сприяє забезпеченню продовольчої безпеки та ефективного управління ланцюгами постачання. Впровадження штучного інтелекту в точне землеробство дозволяє аграріям краще враховувати специфічні потреби кожного поля. Це в свою чергу сприяє ефективнішому використанню ресурсів і зменшенню негативного впливу на навколишнє середовище.

Очікується, що до 2050 року населення світу досягне 9,7 мільярдів, що створить величезне навантаження на виробництво продуктів харчування. Зміна клімату посилює проблему непередбачуваною погодою, посухою та екстремальними температурами. Прогнозування врожайності допомагає фермерам приймати ранні обґрунтовані рішення та оптимізувати ресурси [9].

На рисунку 1.4 показано візуалізацію зміну населення на планеті Земля.

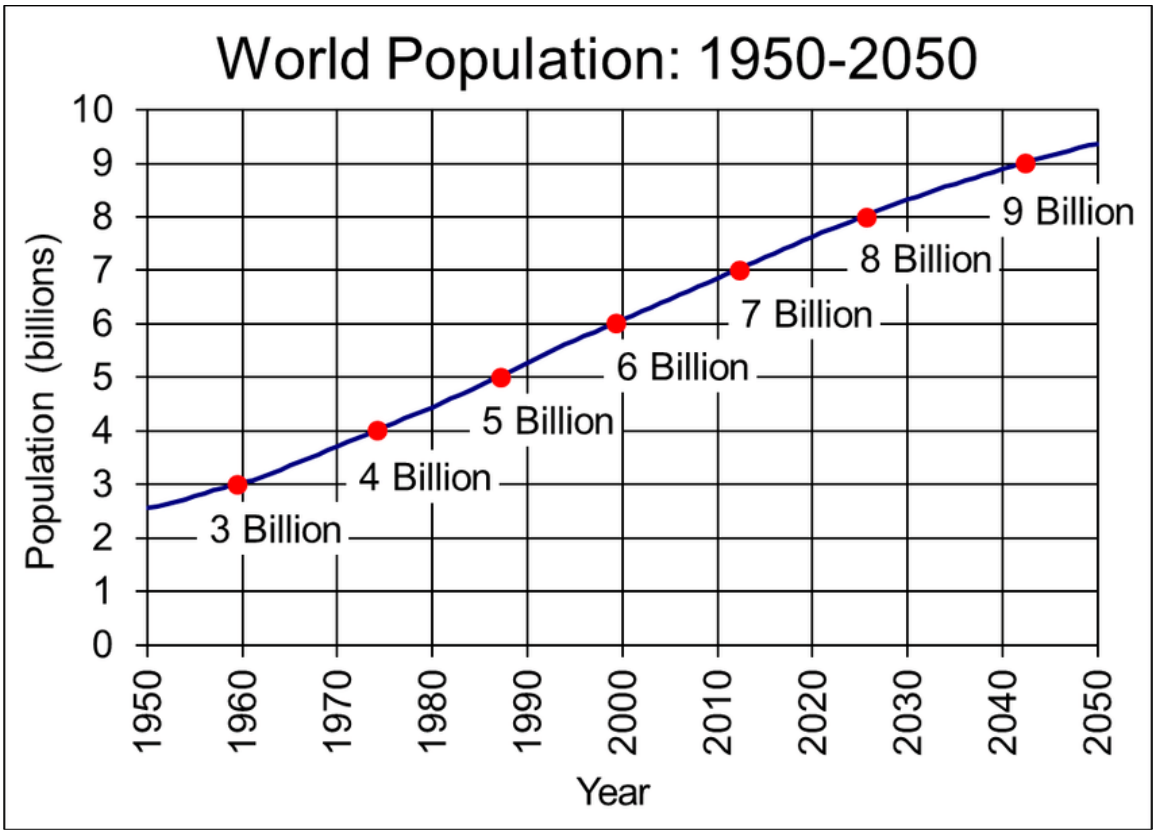


Рисунок 1.4 – Прогнозування популяції населення на планеті Земля [10]

Стале сільське господарство прагне задовольнити харчові потреби зростаючого населення, одночасно захищаючи екосистеми. Прогнозування врожайності сприяє цьому, допомагаючи фермерам ефективно використовувати ресурси та зменшувати їхній вплив на навколишнє середовище. Наука про дані та машинне навчання є ключовими у розробці цих моделей. Вони враховують різні фактори, такі як якість ґрунту, погода, сорти культур і шкідники. Це дозволяє фермерам передбачати виклики та готуватися до них, зменшуючи потребу у воді, добривах та пестицидах [11].

Успіх урожаю починається з якості ґрунту. Дослідження допомагає оцінити фізичні, хімічні та біологічні властивості ґрунту. Аналіз зразків земляного шару в лабораторіях виявляє необхідні поживні речовини для певних культур. Ці знання скеровують фермерів у виборі правильних добрив, зрошення та методів управління.

Вологість ґрунту також є ключовою для росту культур. Занадто багато або занадто мало води може зашкодити розвитку рослин і врожаю. Удосконалені датчики та системи поливу допомагають фермерам підтримувати правильний рівень вологості. Це сприяє здоровому росту, економить воду та знижує ризик втрати врожаю [11].

Методи на основі даних змінили прогнозування врожайності. Вони дозволяють фермерам і політикам приймати обґрунтовані рішення з точними прогнозами. Використовуючи аналітику великих даних і алгоритми ШІ, дослідники виявляють приховані закономірності. Вони отримують цінну інформацію з величезних наборів даних за роки та різноманітних джерел.

Наука про дані в сільському господарстві розвивається прогнозним моделюванням для оцінки врожайності. Алгоритми машинного навчання, такі як регресія та нейронні мережі, демонструють високу точність. Вони прогнозують врожайність на основі таких факторів, як погода, вологість ґрунту та показники рослинності [12].

Машинне навчання зробило революцію у прогнозуванні врожайності. Навчаючи моделі на величезних наборах даних, дослідники створюють надійні алгоритми. Ці алгоритми адаптуються до складності сільськогосподарських систем. Для прикладу можна відмітити [12]:

1. Прогнозування рівнів урожайності кукурудзи в поясі США за допомогою random forest та gradient boosting regressions.

2. Оцінка врожайності пшениці в Австралії за допомогою опорних векторних машин і нейронних мереж.

3. Прогнозування врожайності цитрусових і полуниці у Флориді за допомогою лінійної регресії та часткової регресії найменших квадратів.

4. Прогнозування врожайності рису в Китаї за допомогою згорткових нейронних мереж і моделей довготривалої короткочасної пам’яті (LSTM).

Великий успіх даних методів залежить від якості вхідних даних. Для підтримання сталого результату й отримання точних показників спеціалісти використовують засоби дистанційного зондування та пристрої Інтернет речей,

такий підхід покращує доступність даних і робить підвищення точності вихідного результату [13].

До технології використання дистанційного зондування відносять, супутникові зображення, аерофотознімки та геопросторові дані для відстеження стану розвитку необхідної культури на великих територіях.

До найпопулярніших індексів стану рослинності відносять:

1. NDVI (з англ. Normalized Difference Vegetation Index) – нормалізований відносний індекс рослинності, за яким можна робити висновки про розвиток біомаси рослин під час вегетації. Є різні методи визначення цього показника, і кожен має свої переваги та недоліки [14].

Зазначається, що зелене листя рослин поглинає електромагнітні хвилі в червоному діапазоні і відображає хвилі в ближньому інфрачервоному. Чим більша листкова поверхня рослин і чим більше хлорофілу в листі, тим сильніше рослини поглинають червоне світло, що потрапляє на них. Приклад NDVI аналізу показано на рисунку 1.5.

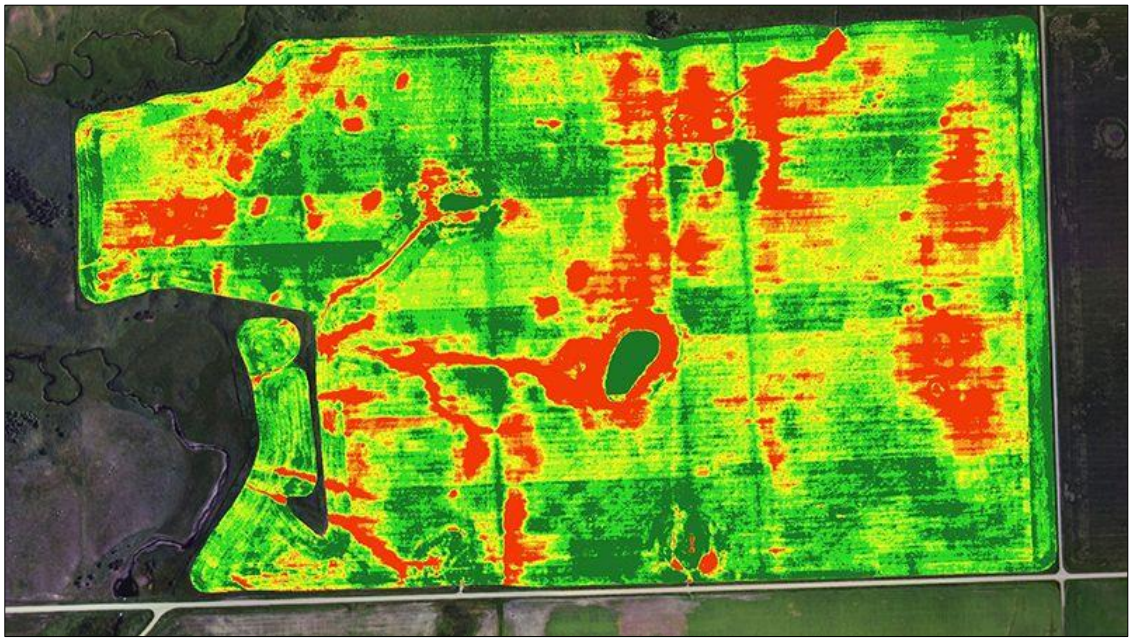


Рисунок 1.5 – NDVI аналіз вегетаційного стану рослинності на земельній ділянці [15]

2. VCI (з англ. Vegetation Condition Index) на рисунку 1.6 показано приклад виконаного VCI аналізу.

Рисунок 1.6 – Приклад VCI індексу на земельній ділянці [16]

Даний індекс використовується для оцінки стану рослинності та моніторингу змін у її розвитку протягом вегетаційного періоду. Цей індекс допомагає визначити, наскільки здорові та розвинуті рослини в конкретному регіоні в порівнянні з їхніми звичайними умовами [16].

3. Індекс рослинності з широким динамічним діапазоном або WDRVI, аналогічно вимірює рівні ближнього інфрачервоного та червоного світла, але є принаймні втричі точнішим ніж NDVI, оскільки він здатний вловити тонкі відмінності в рослинному покрові де висока густота рослинності, що особливо важливо для посівів із густим пологом і стиглих посівів [17]. Приклад використання WDRVI індексу показано на рисунку 1.7.

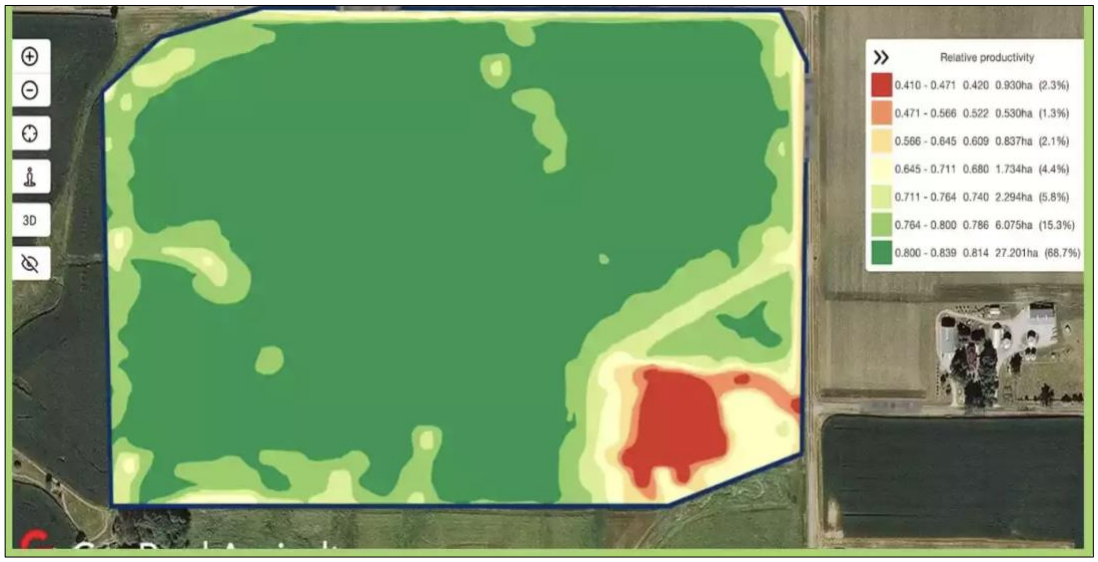


Рисунок 1.7 – Приклад аналізу індексу WDRVI [18]

Отже, умовах зміни клімату та зростання населення технології дистанційного зондування мають вирішальне значення для продовольчої безпеки та сталого сільського господарства. Ці інноваційні інструменти дають змогу фермерам і дослідникам підвищувати врожайність, зберігати ресурси та будувати стійку продовольчу систему на майбутнє.

Комп’ютерний зір має вирішальне значення для прогнозування врожайності, аналізуючи зображення високої роздільної здатності із супутників, дронів або камер. Удосконалені алгоритми виявляють і кількісно визначають показники здоров’я рослин, такі як індекс площі листя та вміст хлорофілу. Поєднуючи комп’ютерний зір із машинним навчанням, дослідники розробляють моделі, які точно оцінюють врожайність на основі візуальних особливостей і моделей росту. У таблиці 1.1 показано сфери застосування

технологій машинного навчання та результат, який можна отримати у сфері сільського господарства [19].

Таблиця 1.1 – Аналіз підходів обробки даних за допомогою ML

	Тип	**Застосування**	**Результат**
1	Регресія	Моделювання зв’язків між врожайністю та вхідними даними	Прогнозування врожайності на основі погоди, ґрунту
2	Дерева рішень й випадкові ліси	Обробка складних зв’язків і подій між функціями	Визначення основних факторів, що впливають на врожайність СГ культур
3	Векторні машини	Обробка даних великої розмірності та нелінійних зв’язків	Моделювання складних СГ систем
4	Нейронні мережі (CNN, RNN)	Отримання просторових часових залежностей у данних про культуру	Генерація даних часових рядів
5	Комп’ютерний зір	Аналіз зображення посівів	Виявлення показників вегетації рослини та оцінки врожайності

Інтеграція методологій і технологій ШІ трансформує сільськогосподарську галузь. Використовуючи наукові дані, фермери можуть приймати обґрунтовані рішення, оптимізувати ресурси та зменшити ризики.

1.2 Агрометеорологічний огляд

Серед метеорологічних факторів температура й опади мають особливе значення для врожайності та рівня виробництва. Урожайний 2022 рік в ЄС зазнав екстремальних погодних явищ з точки зору як температури, так і кількості опадів.

Серпневі дощі в більшості країн Північної та Центральної Європи призвели до деякої затримки початку посіву ріпаку. Тим не менш, загалом погодні умови дозволили завершити посів ріпаку у відповідний часовий проміжок у більшості частин Європи. Відповідний температурний режим і вологість ґрунту забезпечили появу та ранній розвиток сходів ріпаку, дозволили рослинам увійти в зимовий період у хорошому стані. Однак дуже сухі умови у вересні в деяких частинах центральної, східної та південної Європи, зокрема в Румунії, погіршили посів і проростання. Хоча опади в жовтні покращили умови зволоження ґрунту (особливо на півдні Румунії), рослини продовжували відставати у своєму розвитку до зими [20].

Початок посівної кампанії озимих зернових трохи затримався в північних країнах через складний урожай попередніх культур та надлишок опадів, але після цього посіви продовжилися. М’які температури та достатня вологість ґрунту сприяли швидкому виникненню сходів і гарному формуванню деревостану в більшості регіонів. Однак у Західній Європі, зокрема у Франції, переважання нижчих середніх температур з початку жовтня перешкоджало ранньому розвитку озимої пшениці, і лише кілька полів увійшли в стадію кущення до грудня. У Болгарії та Румунії посів спочатку був відкладений через посушливі умови, але добре просунувся після значних опадів у жовтні та листопаді, які в поєднанні зі сприятливими температурами створили адекватні умови для проростання та появи сходів [20].

Зимовий сезон 2021/2022 був теплішим, ніж зазвичай, майже в усіх частинах Європи; особливо це було помітно в північній Німеччині, Польщі, країнах Балтійського моря та Скандинавії, де середні температури перевищували

середні багаторічні (LTA) до 2°C. Ці умови дозволили навіть пізно посіяним озимим культурам добре прижитися і увійти у весну в майже хорошому стані в більшості країн Європи.

Найзначнішою холодною погодою взимку було різке зниження температури на більшій частині Балканського півострова та в східній Європі в останні дні січня, коли мінімальна денна температура опустилася до -24°C (у Румунії). Це тривало лише кілька днів, завдавши незначної шкоди посівам.

Все більше занепокоєння викликало прогресування сильної посухи на більшій частині Піренейського півострова, на півдні Франції та на північному заході Італії. Принаймні протягом зими прямий вплив на посіви залишався відносно обмеженим. Однак історично низький рівень водних резервуарів (тобто снігових покривів в Альпах і внутрішніх водоймах) викликав серйозне занепокоєння щодо наявності води для зрошення наприкінці весни та влітку.

Весна 2022 року відзначилася більш сухою, ніж зазвичай, умовою в більшості частин Європи [20].

Зазвичай у більшості регіонів квітень був холоднішим, ніж зазвичай. Виразне похолодання в першій половині квітня поширилося від центральної Іспанії до регіону Балтійського моря. Вплив на однорічні культури був обмеженим. Однак було завдано серйозної та незворотної шкоди квітучим фруктовим культурам, особливо в Іспанії.

Холодні квітневі умови призвели до певних затримок посіву ярих культур. Крім того, низький вміст води в ґрунті викликав занепокоєння щодо раннього розвитку кукурудзи та соняшнику у Франції, східній Німеччині, Угорщині та Румунії.

У південній Франції та північній Італії зберігалися та поширювалися умови посухи, що призвело до погіршення перспектив посівів літніх культур та очікуваної врожайності озимих та ярих зернових. В Іспанії та Португалії весна почалася зі сприятливих дощів, але вологість ґрунту залишалася низькою через попередню посуху; надзвичайно спекотні та сухі умови в травні серйозно знизили потенціал врожайності в основних регіонах виробництва.

У Румунії та Угорщині холодні та сухі умови в березні та квітні негативно вплинули на посіви озимих; тоді як постійний дефіцит дощів у поєднанні з високими температурами в наступні місяці ще більше знизив їх потенціал врожайності [20].

Літо 2022 року характеризувалося надзвичайно спекотними або сухими погодними умовами у значній частині Європи. Середньодобові максимальні температури були найвищими за всю історію спостережень у більшій частині Західної Європи та в значних частинах південної, центральної та північної Європи, а рекордно високі денні максимуми спостерігалися в багатьох із цих регіонів. На рисунку 1.8 показано температурну карту Європи у період 2022 року.

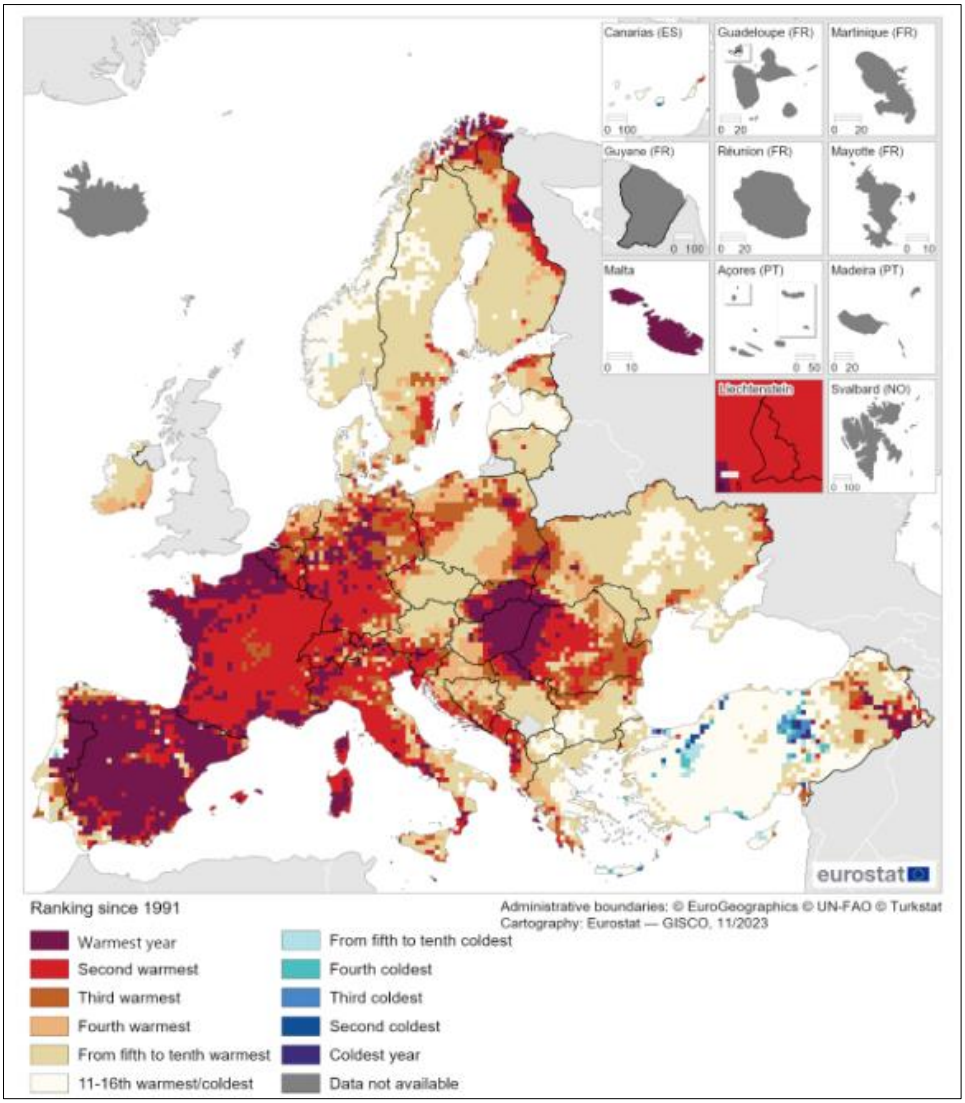


Рисунок 1.8 – Температурне відхилення у країнах Європи [21]

У більшості регіонів озимі та ярі зернові були вже надто просунуті у розвитку, щоб ці умови негативно вплинули на них – насправді теплі та сухі умови були сприятливими для збору врожаю. Однак у центральній Німеччині, Словенії, Хорватії, східній Словаччині та східній Угорщині постійний дефіцит опадів у поєднанні з хвилями спеки в липні скоротив налив зерна озимих культур і ярого ячменю, що негативно вплинуло на їхню врожайність.

Що стосується посівів ярових культур, то Угорщина, Румунія, Болгарія, Іспанія, Франція, центральна та північна Італія, Словенія, Хорватія, східна Словаччина та центральна Німеччина були одними з найбільш постраждалих регіонів. Періоди водного та теплового стресу частково збігалися з чутливими фазами цвітіння та наливання зерна росту рослин, що призвело до незворотної втрати потенціалу врожайності. Однак нестача опадів у поєднанні з іноді високими температурами також негативно вплинула на ярові культури в інших регіонах. Кілька країн запровадили заходи щодо обмеження використання води для зрошення. На рівні країни лише Польща, Данія та Швеція мали сприятливі умови для росту врожаю протягом літа [20].

На рівні ЄС найбільше постраждали посіви зернової кукурудзи, соняшнику, рису та сої. Цукровий буряк, картопля та зелена кукурудза, які здебільшого вирощуються в північних регіонах, зазнали меншого впливу та навіть трохи виграли від сонячних і тепліших, ніж зазвичай, умов, де зрошення було достатнім для задоволення потреб у воді [21].

У більшості постраждалих регіонів літня посуха закінчилася протягом серпня. Однак покращення погодних умов настало надто пізно, щоб істотно вплинути на літні культури.

1.3 Аналіз отриманого врожаю у країнах Європи

У 2022 році виробництво зернових (включно з рисом) в ЄС, за оцінками, склало 270,9 млн тон. Це на 26,7 млн тон менше порівняно з 2021 роком, що відповідає зниженню на 9,0%. Таким чином, рівень виробництва знизився навіть

нижче за рекордні 307,9 млн тон, зафіксовані у 2014 році, як детально показано на рисунку 1.9.

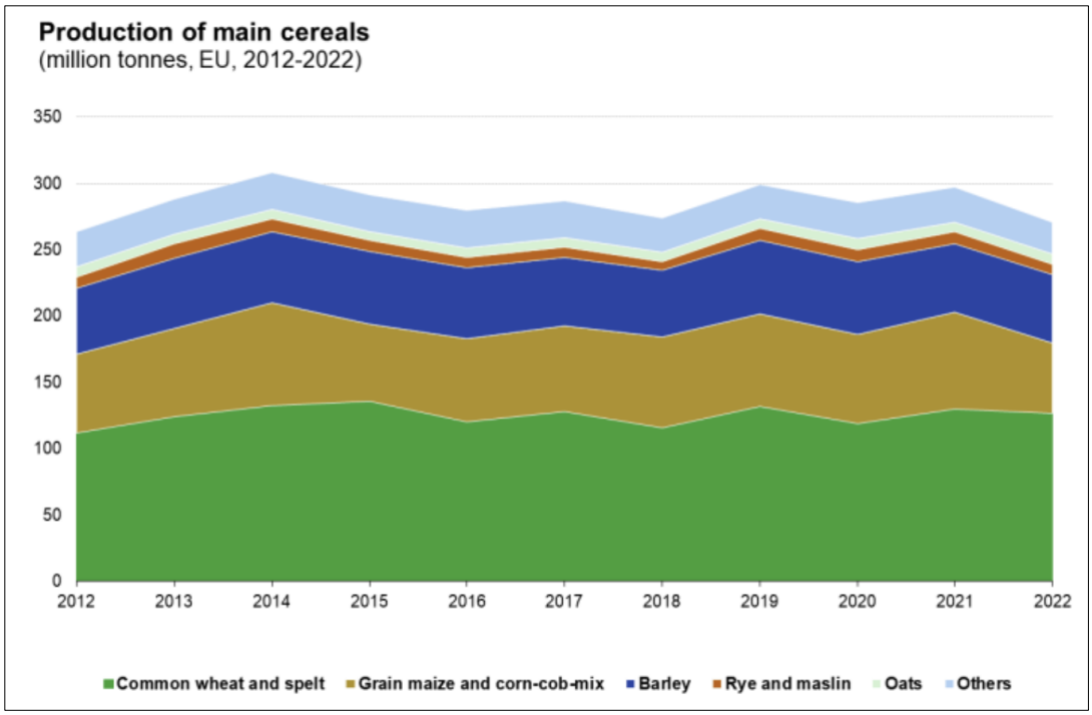


Рисунок 1.9 – Об’єми врожаю в період 2012 – 2022 роки у Європі [22]

У 2022 році Франція зібрала 59,9 млн тон зернових, що становило 22,1% від загального врожаю в ЄС. Німеччина зібрала 43,5 млн тон (16,1% від загального обсягу в ЄС), Польща — 35,0 млн тон (12,9%), Іспанія — 19,3 млн тон (7,1%), а Румунія — 18,9 млн тон (7,0%).

Загальне скорочення виробництва зернових у ЄС у 2022 році було спричинене посухою, що вразила такі країни, як Румунія (-32,1%, зменшення на 8,9 млн тон), Франція (-10,4%, скорочення на 7,0 млн тон), Іспанія (-24,4%, зниження на 6,2 млн тон) та Угорщина (-35,2%, зменшення на 4,9 млн тон). Лише в кількох країнах спостерігалося зростання врожаю, зокрема в Німеччині (+2,6%, приріст на 1,1 млн тон), Фінляндії (+39,1%, зростання на 1,0 млн тон після неврожаю у 2021 році) та Польщі (+2,9%, приріст на 1,0 млн тон) [22].

Нестабільність на світових ринках, підсилена війною в Україні, та загальне скорочення виробництва зернових у ЄС призвели до подальшого зростання цін на зернові культури, які у 2022 році зросли в середньому на 45,6% порівняно з

2021 роком у номінальному вираженні. На рисунку 1.10 представлена динаміка індексів вихідних цін на зернові.

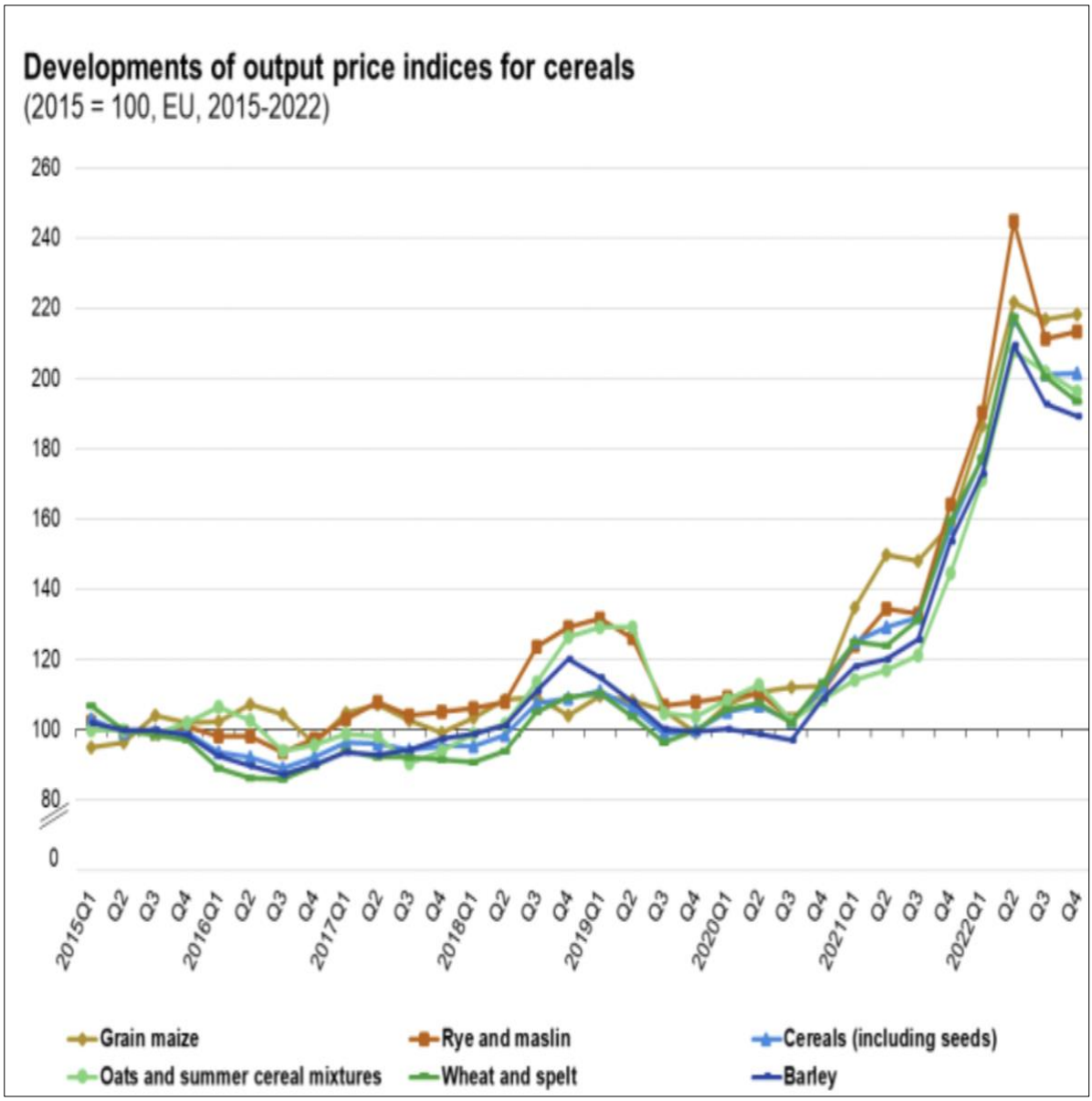


Рисунок 1.10 – Динаміка ціноутворення на зернові культури у Європі [23]

Прискорене зростання цін на зернові у 2022 році торкнулося всіх їх видів: вівса (+58,2%), жита та тритикале (+53,3%), ячменю (+47,9%), пшениці та полби (+46,3%) і кукурудзи (+41,0%). Загалом ціни на зернові в 2022 році були вдвічі вищими, ніж у 2015 році.

1.4 Огляд та аналіз відомих аналогів

Використання різних методів аналізу даних відбувається постійно і даний процес завжди піддається удосконаленню. Для роботи з даними та прогнозування врожаю аналітики використовують різного рівня складності алгоритми та підходи, щоб отримати бажаний результат. З одним з найпростіших підходів прогнозування є регресійний аналіз. Приклад графіку лінійної регресії показано на рисунку 1.11.

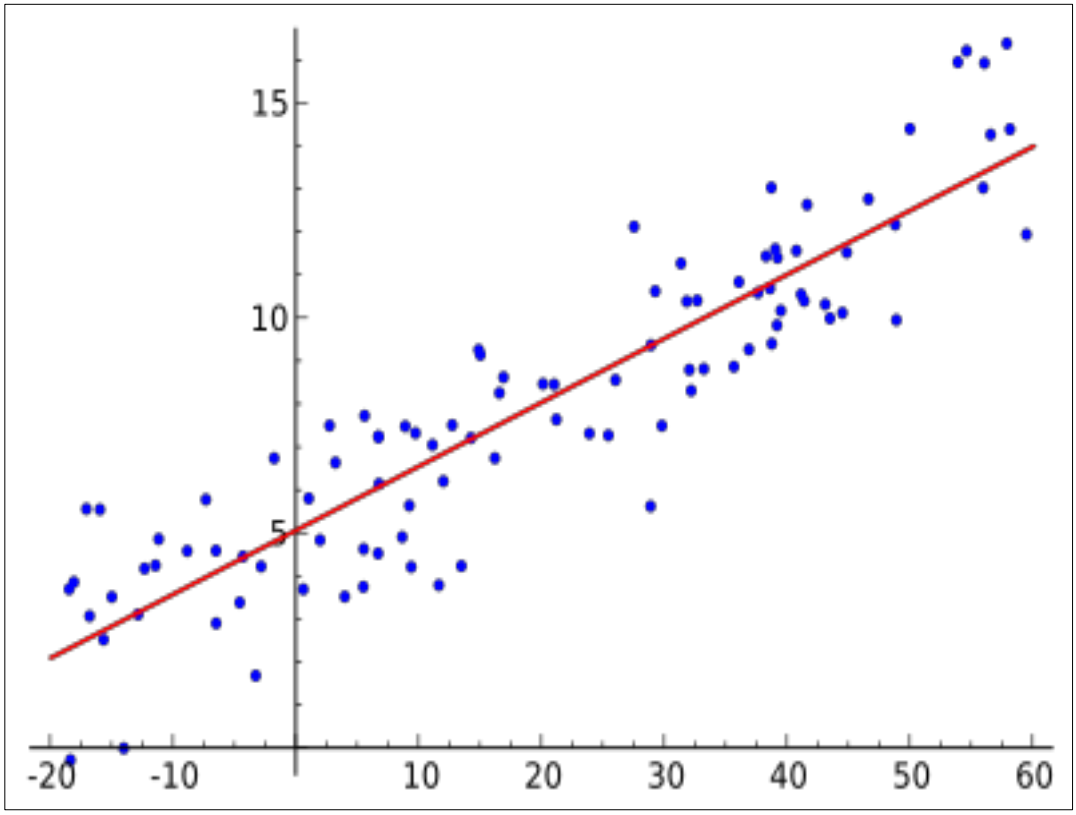


Рисунок 1.11 – Візуалізація даних лінійної регресії [24]

Являє собою статистичний метод, який використовується для визначення взаємозв'язку між змінними. Основна мета регресійного аналізу полягає в тому, щоб з'ясувати, як одна або кілька незалежних змінних впливають на залежну змінну. Це дозволяє прогнозувати значення залежної змінної на основі відомих значень незалежних змінних. До основних складових даного типу аналізу відноситься [24]:

Залежна змінна (Y) – це змінна, значення якої потрібно передбачити. В аграрній сфері можна вважати це значення як врожайність.

Незалежні змінні (X) – це змінні, які впливають на залежну змінну. Сюди можна віднести погодні умови, кількість добрив, врожай за минулий рік.

Регресійна модель – математичне рівняння, яке описує залежність між змінними. Приклад лінійної регресії показано на формулі:

$$Y = a + bX + \varepsilon, \tag{1.1}$$

де Y – залежна змінна;

a – константа;

b – коефіцієнт нахилу, що показує вплив X на Y;

X – незалежна змінна;

ε – помилка моделі.

Перевагами даного типу передбачення, є простота в інтерпретації та чіткість взаємодії змінних. Також даний тип передбачення можна використовувати як для прогнозування врожайності, так і для ціноутворення або ж аналітики ризиків [24].

До недоліків даного методу можна віднести, обмеженість лінійними випадками взаємозв’язку даних, для більш складніших залежностей потрібно використовувати потужніші методи такі як нейронні мережі.

LSTM (Long Short-Term Memory) — це тип рекурентної нейронної мережі (RNN), спеціально розроблений для роботи з часовими рядами та задачами, де важливо враховувати довготривалі залежності в даних. У прогнозуванні врожайності сільськогосподарської продукції LSTM може бути дуже корисним, оскільки дозволяє моделювати складні взаємозв’язки між багатьма факторами, які впливають на врожайність (наприклад, погодні умови, тип ґрунту, історичні дані про врожайність) [25].

На врожайність впливають не лише короткострокові фактори (наприклад, погода протягом останніх кількох днів), а й довготривалі тенденції, такі як стан

ґрунту за кілька місяців чи навіть років. LSTM дозволяє зберігати інформацію про ці довготривалі впливи в своїй внутрішній пам'яті. Дані про врожайність — це типовий часовий ряд, де є залежність між значеннями в різні моменти часу. LSTM добре працює з такими даними, оскільки може вчитися на послідовності історичних даних для прогнозування майбутніх значень [25].

Врожайність сільськогосподарських культур часто має сезонний характер (наприклад, весняні та осінні збори врожаю). LSTM здатен навчитися на цих сезонних патернах і точно прогнозувати результати на основі повторюваних сезонних закономірностей. На рисунку 1.12 показано алгоритм роботи методу LSTM.

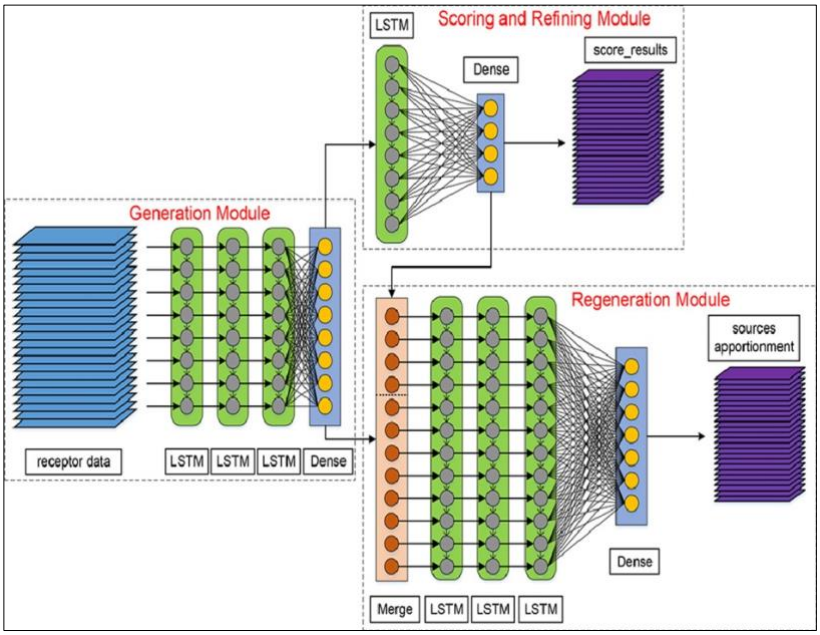


Рисунок 1.12 – Візуалізація роботи LSTM моделі [26]

Для кращого розуміння розберемо роботу моделі на прикладі: ми маємо історичні дані про врожайність пшениці, кількість опадів і температуру за останні 10 років. Ми можемо використовувати LSTM для створення моделі, яка прогнозує врожайність пшениці на основі минулих даних про погоду та врожай.

– дані: Послідовності погодних умов (температура, опади) за кожен рік, а також врожайність за кожен рік;

– LSTM модель: Модель буде навчена на цих даних для виявлення довготривалих залежностей між погодними факторами та врожайністю;

– прогнозування: Модель прогнозує врожайність на основі прогнозу погоди на наступний рік;

Отже, LSTM є потужним інструментом для прогнозування врожайності сільськогосподарської продукції завдяки здатності обробляти часові ряди та враховувати довготривалі залежності між даними.

APSIM (Agricultural Production Systems Simulator) — це програмне забезпечення, яке використовується для моделювання сільськогосподарських систем, зокрема для прогнозування врожайності сільськогосподарських культур під впливом різних факторів, таких як клімат, управлінські практики та ґрунтові умови. APSIM дозволяє оцінювати вплив змінних, що варіюються в часі та просторі, і є потужним інструментом для прийняття рішень у сільському господарстві, адаптації до змін клімату та оптимізації аграрних практик. Інтерфейс програмного забезпечення показано на рисунку 1.13.

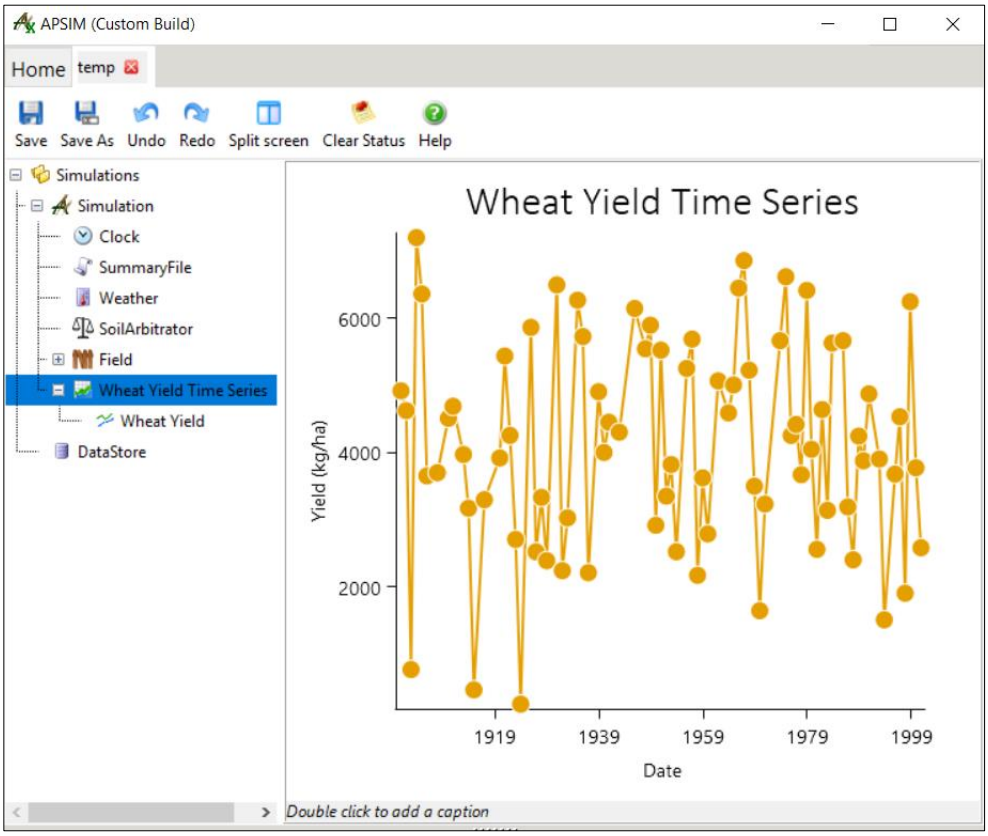


Рисунок 1.13 – Інтерфейс ПЗ APSIM [27]

APSIM може моделювати ріст та розвиток багатьох сільськогосподарських культур (наприклад, пшениця, кукурудза, рис, соя). Це дозволяє передбачати врожайність на основі різних факторів: кліматичні умови, агротехнічні прийоми, тип ґрунту, сценарії зміни клімату [27].

APSIM використовує історичні та прогнозовані кліматичні дані для моделювання того, як погодні умови (температура, опади, сонячне випромінювання) впливають на ріст культур. Це особливо важливо для передбачення потенційних впливів кліматичних змін на продуктивність сільського господарства. Отже, це потужний інструмент, який вміщує в собі комплекс алгоритмів та моделей машинного навчання й нейронних мереж, що в комплексному використанні дають високі показники передбачення необхідної інформації.

Використання APSIM не лише корисне для реального агробізнесу, але й для навчання. Вона дозволяє зрозуміти, як працює весь процес, без потреби чекати роками, поки буде результати в польових умовах. Це дуже ефективно для студентів, які хочуть заглибитися в технології управління сільськогосподарськими системами або провести наукові експерименти [27].

Основним недоліком даного програмного забезпечення, є неможливість використовувати його звичайним користувачам без корпоративної угоди між підприємствами.

1.5 Висновки

Під час виконання даного розділу було досліджено сферу вирощування сільськогосподарських культур, проаналізовано важливі аспекти даної роботи. Виконано агрометеорологічний огляд у період 2021-2022 років. На основі даних за попередні роки, було проаналізовано стан вирощування зернових культур і визначено, що передбачення даних під час планування майбутнього врожаю, є важливим процесом, який допоможе правильно розподілити ресурси підприємства.

Проаналізовано популярні методи та програмні забезпечення для роботи з даними у сфері аграрного бізнесу, визначено, що кожен алгоритм передбачення даних має свої переваги та недоліки. Отримані знання було використано під час проєктування власного програмного коду Python та кодування алгоритмів передбачення даних.

Отримано структуровані знання про дану сферу роботи, проаналізовані та опрацьовані дані будуть корисні для вибору оптимальних технологій під час здійснення передбачення даних.

2. НАЛАШТУВАННЯ ТЕХНОЛОГІЙ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Підготовка середовища розробки та мови програмування

У сучасному світі інформаційних технологій вибір мови програмування та середовища розробки є одними з найважливіших рішень, які впливають на успіх проекту. Різноманіття доступних мов і середовищ розробки може бути як перевагою, так і викликом для розробників. Правильний вибір залежить від багатьох факторів, включаючи тип проекту, вимоги до продуктивності, доступність ресурсів, досвід команди та специфічні потреби користувачів.

У цьому розділі ми розглянемо основні принципи вибору мови програмування, різні мови, їх переваги та недоліки, а також огляд популярних середовищ розробки, які можуть суттєво полегшити процес програмування. Ми також торкнемося того, як ефективний вибір мови і середовища може вплинути на продуктивність команди, якість коду та швидкість розробки.

Для аналізу даних та навчання моделей найпопулярнішими мовами програмування є:

1. Python – це високоабстрактна, інтерпретована мова програмування, яка відзначається простим та зрозумілим синтаксисом. Вона є дуже популярною у сфері науки про дані завдяки своїй універсальності та наявності великої кількості бібліотек для аналізу даних, статистики та машинного навчання. Перевагами даної мови програмування є простота у вивченні та використанні, величезна спільнота та безліч ресурсів для навчання, гнучкість у використанні для різних типів проектів [28].

2. R – це мова програмування та середовище для статистичних обчислень та графічної візуалізації. Вона широко використовується в академічних та бізнес-середовищах для аналізу даних, статистичних моделей та візуалізації. Найпопулярнішими бібліотеками даної мови є [28]:

– ggplot2: Потужна бібліотека для створення графіків та візуалізації даних за допомогою граматики графіків;

– dplyr: Інструмент для маніпуляції даними, який дозволяє зручно відбирати, фільтрувати та трансформувати набори даних;

– caret: Бібліотека для машинного навчання, яка спрощує процес тренування та оцінювання моделей;

– shiny: Дозволяє створювати інтерактивні веб-додатки для візуалізації даних.

R має сильну підтримку статистичних методів та тестування гіпотез, великий вибір пакетів для специфічних завдань, зручність для візуалізації даних.

3. SQL (Structured Query Language) — це мова, призначена для управління та обробки даних у реляційних базах даних. SQL дозволяє запитувати, вставляти, оновлювати та видаляти дані, а також виконувати складні запити для аналізу. У більшості випадків застосування даної мови зводиться до витягнення та агрегування даних з бази даних, маніпуляції з великими обсягами даних, що зберігаються в таблицях та підготовці даних до подальшого аналізу за допомогою інших мов програмування (Python або R).

Перевагами SQL є можливість оброки великих обсягів даних, зрозумілість та стандартність запитів, можливість роботи з різними реляційними СУБД (MySQL, PostgreSQL, Oracle) [28].

4. Julia – це мова програмування, яка спеціалізується на високопродуктивних обчисленнях. Вона поєднує в собі простоту Python з продуктивністю C, що робить її привабливою для наукових обчислень та машинного навчання. Популярними бібліотеками є DataFrames.jl: Аналог Pandas для роботи з табличними даними, Plots.jl: Для візуалізації даних, Flux.jl: Легка та гнучка бібліотека для створення нейронних мереж [28].

5. SAS (Statistical Analysis System) — це програмне забезпечення, яке надає інструменти для статистичного аналізу, управління даними та звітності. SAS використовується в бізнесі, наукових дослідженнях та охороні здоров'я.

Застосування:

– статистичний аналіз та моделювання даних;
– обробка великих обсягів даних;
– генерація звітів та візуалізація результатів.

Переваги:

– висока продуктивність при роботі з наборами даних;
– надійні та перевірені статистичні методи;
– розширена документація та підтримка.

У процесі аналізу та оцінки різних мов програмування, які можуть бути використані для аналізу даних та навчання моделей, було прийнято рішення обрати Python як основну мову для нашого проекту. Це рішення базується на кількох ключових факторах:

– простота та зрозумілість: Python має інтуїтивно зрозумілий синтаксис, що полегшує навчання та швидке освоєння, особливо для новачків у програмуванні. Це дозволяє розробнику зосередитися на реалізації алгоритмів, а не на труднощах коду.

– широкий спектр бібліотек: Python пропонує безліч потужних бібліотек для аналізу даних, візуалізації та машинного навчання, таких як Pandas, NumPy, Matplotlib, Scikit-learn, TensorFlow та PyTorch. Це забезпечує розробнику доступ до найсучасніших інструментів для виконання різноманітних завдань.

– активна спільнота: Велика та активна спільнота Python забезпечує наявність численних ресурсів, документації та підтримки, що спрощує вирішення проблем та пришвидшує процес розробки.

При виборі середовища розробки (IDE) для нашого проекту важливо врахувати кілька ключових факторів, які впливають на ефективність роботи, зручність та продуктивність команди. У даному описі буде розглянуто різні IDE, їх переваги та недоліки, щоб прийняти обґрунтоване рішення.

Створювати програми на Python дають можливість такі середовища, як PyCharm, VS Code, Jupyter Notebook, Spyder. Усі вище перераховані засоби для розробки програмного забезпечення вимагають обов'язкового встановлення локально на пристрій, а для доступу з мережі інтернет та хмарного сховища зазвичай для кінцевого користувача потрібно буде додаткова передплата з розблокуванням даного функціоналу [29].

Для забезпечення даних потреб існує онлайн сервіс Kaggle, це ще одне важливе середовище, яке варто розглянути для роботи з даними, особливо для аналітики та машинного навчання.

Детальніший опис Kaggle як платформи для розробки та аналізу даних [30]:

– доступ до наборів даних: Kaggle має величезну бібліотеку публічних наборів даних з різних галузей, що дозволяє дослідникам і розробникам легко знаходити дані для своїх проектів;

– інтерактивні середовища: Платформа пропонує інтерактивні Jupyter Notebooks, що дозволяє писати та виконувати код безпосередньо у веб-браузері. Це зручно для швидкого прототипування та експериментування;

– змагання з машинного навчання: Kaggle організовує змагання, де учасники можуть змагатися в розробці найкращих моделей для конкретних задач. Це чудовий спосіб отримати досвід, перевірити свої навички та здобути визнання;

– спільнота: Kaggle має активну спільноту, де користувачі можуть обмінюватися ідеями, задавати питання та ділитися ресурсами. Це створює можливості для навчання та співпраці;

– навчальні курси: Платформа також пропонує навчальні матеріали та курси з машинного навчання, аналізу даних та програмування на Python, що дозволяє новачкам швидко вчитися та покращувати свої навички;

Перевагами користування даної платформи є [30]:

– легкий доступ до даних: Можливість швидко знаходити і використовувати різноманітні набори даних;

– безкоштовність: Багато функцій доступні безкоштовно, що робить Kaggle привабливим для студентів та дослідників;

– практичний досвід: Участь у змаганнях і виконання проектів на платформі допомагає здобути практичні навички.

До недоліків можна віднести:

– обмеження ресурсів: Безкоштовна версія може мати обмеження щодо обчислювальних ресурсів, доступних для виконання моделей;

– не підходить для великих проектів: Для великих комерційних проектів або командної роботи Kaggle може не забезпечити всіх необхідних функцій.

Kaggle є чудовим інструментом для тих, хто займається аналізом даних та машинним навчанням. Завдяки своїм ресурсам, спільноті та змаганням, платформа забезпечує відмінну можливість для навчання, експериментів і розвитку навичок. Отже, для вирішення поставленої задачі у ході виконання магістерської кваліфікаційної роботи доцільно використовувати Kaggle у якості основного засобу для розробки програмного забезпечення з використанням мови програмування Python.

2.2 Налаштування обраних бібліотек мови Python

Під час вибору мови програмування було проаналізовано її основні бібліотеки, які будуть необхідні для вирішення поставленої задачі. До найпопулярніших бібліотек можна віднести:

1. Pandas є однією з найпопулярніших бібліотек для обробки та аналізу даних у Python. Вона забезпечує потужні структури даних, такі як DataFrame і Series, які дозволяють ефективно працювати з табличними даними. Операції, які можна виконати за допомогою даної бібліотеки: Читання та запис даних з/в різні формати (CSV, Excel, SQL, JSON тощо), маніпуляції з даними (фільтрація, групування, об'єднання), очищення даних (заповнення пропущених значень, видалення дублікатів), статистичний аналіз і агрегація даних [31]. На рисунку 2.1 показано приклад використання бібілотеки Pandas на практиці.

```
import pandas as pd

data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

Рисунок 2.1 – Використання бібліотеки Pandas [32]

Отже, дана бібліотека дає потужні можливості для ефективного керування вхідними даними й виконання подальших обробок для отримання бажаного результату.

2. NumPy (Numerical Python) – це бібліотека для наукових обчислень, яка забезпечує підтримку багатовимірних масивів і матриць, а також чисельних операцій над ними. Забезпечує можливість створення та маніпуляція з багатовимірними масивами (ndarray), виконання математичних операцій (арифметичні, тригонометричні, статистичні), лінійна алгебра, обробка сигналів і статистичні розрахунки [31]. На рисунку 2.2 показано приклад використання бібліотеки NumPy.

```
# Create a NumPy array from a list and tuple

p_list = np.array([900, 200, 300, 99, 54, 80])
print("This is a list into numpy array:\n", p_list)
print("Type: ", type(p_list))

p_tuple = np.array((50, 60, 90, 105, 90.6, 30.8))
print("This is a tuple into numpy array:\n", p_tuple)
print("Type: ", type(p_tuple))
```

```
This is a list into numpy array:
 [900 200 300  99  54  80]
Type:  <class 'numpy.ndarray'>
This is a tuple into numpy array:
 [ 50.   60.   90.  105.   90.6  30.8]
Type:  <class 'numpy.ndarray'>
```

Рисунок 2.2 – Використання бібліотеки NumPy [33]

Як показано на рисунку, за допомогою даної бібліотеки було виконано обробку масиву вхідних даних, таке розподілення даних дуже спрощує подальшу роботу з вхідними даними, особливо, якщо передбачається використовувати датасет в табличному поданні.

3. Matplotlib – це одна з найпопулярніших бібліотек для візуалізації даних у Python, яку широко використовують у наукових і технічних дослідженнях. Вона надає змогу створювати графіки різних типів – від простих лінійних графіків до складних 3D-візуалізацій. Її гнучкість дозволяє змінювати вигляд графіка до найдрібніших деталей, таких як кольори, маркери, підписи осей або масштабування.

Ця бібліотека є особливо корисною, оскільки її можна використовувати для аналізу результатів експериментів, моделювання фізичних процесів або навіть для побудови графічних звітів до лабораторних робіт. Її синтаксис інтуїтивно зрозумілий і легко інтегрується з іншими бібліотеками, такими як NumPy чи Pandas, що дає змогу працювати з великими наборами даних та отримувати графіки на основі обчислень.

Одна з головних переваг Matplotlib – її здатність створювати високоякісні статичні, анімовані або інтерактивні графіки. Наприклад, під час проведення дослідження можна не лише аналізувати залежності між величинами, але й демонструвати ці дані наочно, що значно полегшує розуміння складних

взаємозв’язків. Бібліотека активно оновлюється і підтримується спільнотою, тому в разі труднощів завжди можна знайти готові рішення або отримати допомогу в документації чи форумах [31]. На рисунку 2.3 показано приклад використання даної бібліотеки, яка входить в можливості мови програмування Python.

```
# Setup the subplot2grid layout
fig = plt.figure(figsize=(10, 5))
ax1 = plt.subplot2grid((2,4), (0,0))
ax2 = plt.subplot2grid((2,4), (0,1))
ax3 = plt.subplot2grid((2,4), (0,2))
ax4 = plt.subplot2grid((2,4), (0,3))
ax5 = plt.subplot2grid((2,4), (1,0), colspan=2)
ax6 = plt.subplot2grid((2,4), (1,2))
ax7 = plt.subplot2grid((2,4), (1,3))

# Input Arrays
n = np.array([0,1,2,3,4,5])
x = np.linspace(0,5,10)
xx = np.linspace(-0.75, 1., 100)

# Scatterplot
ax1.scatter(xx, xx + np.random.randn(len(xx)))
ax1.set_title("Scatter Plot")

# Step Chart
ax2.step(n, n**2, lw=2)
ax2.set_title("Step Plot")

# Bar Chart
ax3.bar(n, n**2, align="center", width=0.5, alpha=0.5)
ax3.set_title("Bar Chart")

# Fill Between
ax4.fill_between(x, x**2, x**3, color="steelblue", alpha=0.5);
ax4.set_title("Fill Between");

# Time Series
dates = pd.date_range('2018-01-01', periods = len(xx))
ax5.plot(dates, xx + np.random.randn(len(xx)))
ax5.set_xticks(dates[::30])
ax5.set_xticklabels(dates.strftime('%Y-%m-%d')[::30])
ax5.set_title("Time Series")

# Box Plot
ax6.boxplot(np.random.randn(len(xx)))
ax6.set_title("Box Plot")
```

Рисунок 2.3 – Використання бібліотеки Matplotlib [34]

З представленого програмного коду Python можна побачити, що бібліотека Matplotlib має розширений набір для візуалізації вихідних даних, які були оброблені користувачем. Кожен тип відображення даних має індивідуальні налаштування, отже Matplotlib є гнучким та невід'ємним інструментом для повноцінної роботи з машинним навчанням.

На рисунку 2.4 показано отриманий результат візуалізації даних за допомогою бібліотеки Matplotlib.

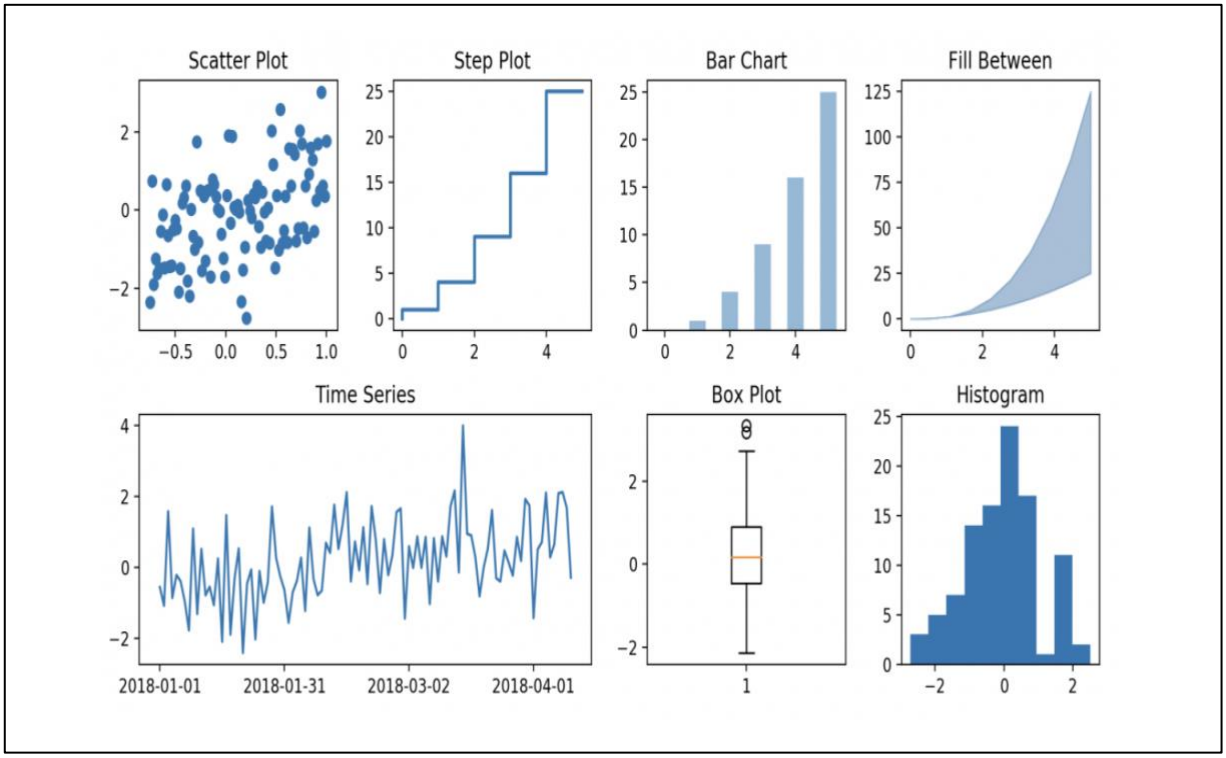
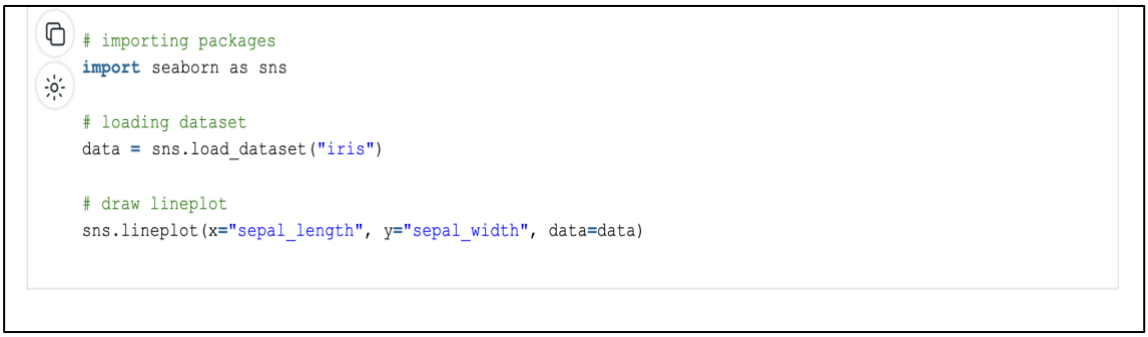


Рисунок 2.4 – Виконаний програмний код з використанням бібліотеки Matplotlib [34]

4. Seaborn – це бібліотека для візуалізації даних, яка побудована на основі Matplotlib і надає більш високий рівень абстракції, що спрощує створення красивих і зрозумілих графіків. Бібліотека Python Seaborn є популярним інструментом для візуалізації даних, який активно використовується в задачах обробки даних і машинного навчання. Вона побудована на базі бібліотеки matplotlib і дозволяє виконувати досліджувальний аналіз даних. Завдяки Seaborn можна створювати інтерактивні графіки для аналізу та відповіді на ключові питання про дані [31].

Для глибшого розуміння функціональності Seaborn і різних способів побудови графіків рекомендується працювати з кількома наборами даних, створюючи візуалізації для їх аналізу. На рисунку 2.5 показано приклад програмного коду з використанням бібліотеки Seaborn.



```
# importing packages
import seaborn as sns

# loading dataset
data = sns.load_dataset("iris")

# draw lineplot
sns.lineplot(x="sepal_length", y="sepal_width", data=data)
```

Рисунок 2.5 – Приклад використання бібліотеки Seaborn [35]

Кінцевий результат обробки вхідних даних та їх візуалізація за допомогою Seaborn, має спільну схожість з використанням бібліотеки Matplotlib. Але в той ж час, якщо переглянути програмний код, можна помітити, що для seaborn було написано менше коду й були мінімізовані початкові налаштування для правильного відображення результату. Отже, дану бібліотеку доцільно використовувати, коли користувачу необхідна швидкість й не потрібні точні налаштування отриманого результату. На рисунку 2.6 показано виконання програмного коду з рисунка 2.5.

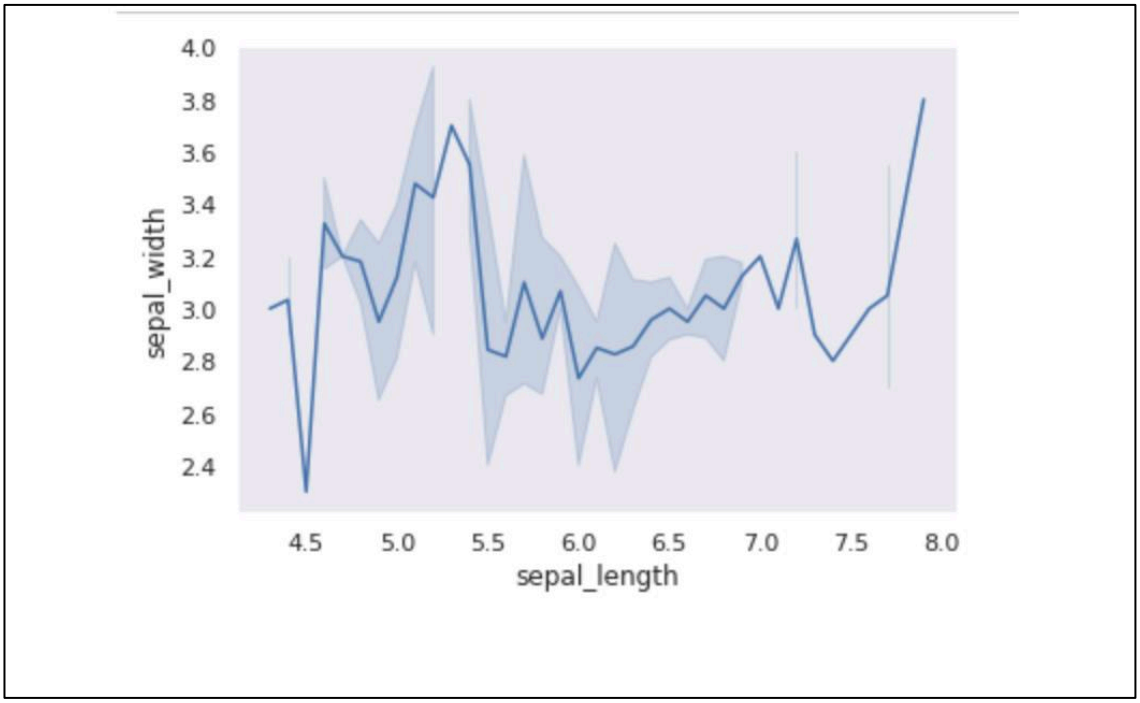


Рисунок 2.6 – Виконаний програмний код з використанням бібліотеки seaborn [35]

5. Бібліотека Scikit-learn є одним із найпопулярніших інструментів для роботи з моделями машинного навчання. Вона надає набір стандартних наборів даних, які можна використовувати для експериментів без необхідності завантаження з зовнішніх джерел. Це суттєво полегшує початкову роботу з бібліотекою. Після ознайомлення з даними слід підготувати їх для введення в модель, оскільки обробка даних є важливим етапом у машинному навчанні. Реальні дані часто містять пропуски, шум, помилки або викиди, які потрібно усунути перед навчанням моделі. Наприклад, використання інструмента StandardScaler дозволяє стандартизувати числові дані, щоб середнє значення кожної ознаки було близьким до нуля, а розподіл даних був уніфікованим [31].

Для навчання моделі важливо оцінити її ефективність на невидимих даних. Це досягається шляхом поділу початкового набору даних на навчальну та тестову вибірки. Функція train_test_split у Scikit-learn автоматизує цей процес, дозволяючи, наприклад, виділити 70% даних для навчання і 30% для тестування. Після цього можна створювати моделі, такі як логістична регресія, підтримка векторної машини чи дерево рішень, залежно від поставленої задачі.

Оцінка продуктивності моделі є завершальним етапом. Для цього Scikit-learn надає метрики, такі як точність, F1-оцінка, повнота тощо. Використовуючи функцію classification_report, можна отримати зведену таблицю з усіма ключовими показниками, що дозволяє оцінити якість моделі. Бібліотека Scikit-learn також пропонує широкий спектр інструментів для попередньої обробки даних, налаштування моделей і аналізу результатів, що робить її універсальним інструментом для задач машинного навчання. Для глибшого розуміння можливостей бібліотеки варто вивчати додаткові матеріали та документацію.

На рисунку 2.7 показано процес навчання моделі даних.

```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier

# Instnatiating the models
logistic_regression = LogisticRegression()
svm = SVC()
tree = DecisionTreeClassifier()

# Training the models
logistic_regression.fit(X_train_scaled, y_train)
svm.fit(X_train_scaled, y_train)
tree.fit(X_train_scaled, y_train)

# Making predictions with each model
log_reg_preds = logistic_regression.predict(X_test_scaled)
svm_preds = svm.predict(X_test_scaled)
tree_preds = tree.predict(X_test_scaled)
```

Рисунок 2.7 – Програмний код з використанням бібліотеки Scikit-learn [36]

З вищепоказаного рисунку можна зрозуміти, що бібліотека Scikit-learn має усі базові функції для проведення навчання моделі даних. Результат виконання програмного коду показано на рисунку 2.8.

```
Support Vector Machine Results:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        17
           1       1.00      1.00      1.00        25
           2       1.00      1.00      1.00        12

    accuracy                           1.00        54
   macro avg       1.00      1.00      1.00        54
weighted avg       1.00      1.00      1.00        54

Decision Tree Results:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94        17
           1       0.96      0.88      0.92        25
           2       0.86      1.00      0.92        12

    accuracy                           0.93        54
   macro avg       0.92      0.94      0.93        54
weighted avg       0.93      0.93      0.93        54
"""
```

Рисунок 2.8 – Результат навчання моделі даних за допомогою бібліотеки Scikit-learn [36]

6. TensorFlow — це відкрита бібліотека, яка підтримує Python і спрощує створення додатків машинного навчання та нейронних мереж. Розроблена Google Brain у 2015 році, вона дозволяє легко отримувати дані, навчати моделі, виконувати прогнози та вдосконалювати результати. TensorFlow об'єднує різні алгоритми машинного навчання, глибокого навчання та надає API для Python і JavaScript, а виконання відбувається на платформі, оптимізованій під C++. TensorFlow підтримує задачі класифікації зображень, перекладу тексту, обробки природної мови, а також диференціальних рівнянь. Моделі можна розгортати на локальних машинах, у хмарі чи на мобільних пристроях. Бібліотека також має попередньо навчені моделі та інструменти, як-от Keras API, що спрощують створення моделей [31].

TensorFlow 2.0, випущена у 2019 році, значно вдосконалила використання бібліотеки, інтегрувавши підтримку TensorFlow Lite для мобільних пристроїв і нові API для розподіленого навчання. Моделі можна розгортати за допомогою Docker, Kubernetes чи API REST. Ключовою перевагою TensorFlow є абстрагування технічних деталей, що дозволяє розробникам зосередитися на

логіці додатка. Інструмент TensorBoard забезпечує візуалізацію роботи графів і спрощує аналіз моделей [31].

7. PyTorch — це популярна бібліотека для створення і навчання моделей машинного та глибокого навчання, розроблена Facebook AI Research у 2016 році. Вона базується на мові Python і відома своєю гнучкістю, простотою у використанні та широкими можливостями. PyTorch дозволяє розробникам працювати з динамічними обчислювальними графами, що дає змогу змінювати структуру моделі «на льоту» під час виконання, на відміну від статичних графів у деяких інших бібліотеках. Це робить її ідеальним вибором для дослідницьких проєктів та експериментів із новими архітектурами нейронних мереж.

Особливістю PyTorch є його тісна інтеграція з Python, завдяки чому бібліотека підтримує інтуїтивно зрозумілий синтаксис і легку відладку. PyTorch використовує бібліотеку Torch для виконання високопродуктивних обчислень, таких як обробка даних на GPU. Вона підтримує широкий спектр моделей, зокрема згорткові нейронні мережі, рекурентні нейронні мережі та трансформери. Крім того, PyTorch має вбудовані інструменти для підготовки даних, що спрощує процес створення моделей, та надає можливості для розподіленого навчання, що підходить для роботи з великими наборами даних [31].

Для практичного використання PyTorch розробники отримують доступ до багатьох попередньо навчених моделей через бібліотеку TorchVision. Ці моделі можна легко адаптувати до нових завдань, скорочуючи час на навчання. PyTorch також активно використовується в академічних і промислових дослідженнях завдяки своїй прозорості й високій продуктивності. Завдяки своїм можливостям і зручності PyTorch став одним із провідних інструментів у сфері машинного навчання, що допомагає створювати інноваційні рішення в різних галузях.

Вибір бібліотеки залежить від конкретних потреб проекту. Для обробки даних зазвичай використовують Pandas і NumPy, для візуалізації – Matplotlib і Seaborn, а для машинного навчання – Scikit-learn, TensorFlow або PyTorch. Ці

бібліотеки разом утворюють потужний набір інструментів, які дозволяють ефективно аналізувати дані та будувати моделі.

2.3 Висновки

У сучасному світі інформаційних технологій вибір мови програмування та середовища розробки обумовлюється багатьма факторами, зокрема вимогами до продуктивності, доступністю ресурсів та досвідом розробника. У цьому контексті мова програмування Python виявилася найкращим вибором завдяки своїй простоті, універсальності та широкому спектру потужних бібліотек, таких як Pandas, NumPy, Matplotlib, Scikit-learn, TensorFlow та PyTorch, що дозволяють ефективно здійснювати обробку, аналіз та візуалізацію даних.

У другому розділі виконано підготовку для початку роботи з програмним кодом. Використано можливості онлайн-сервісу Kaggle. Підключено датасети у створений ноутбук й проведено тестову перевірку цілісності вхідних даних. Для виконання поставленої задачі магістерської кваліфікаційної роботи було зроблено підключення та попереднє налаштування обраних бібліотек мови програмування Python. Задано усі необхідні параметри, які будуть використовуватися в процесі розробки інформаційної технології.

У результаті, поєднання Python як основної мови програмування з бібліотеками для аналізу даних і Kaggle як платформою для розробки, було створено потужну основу для успішної реалізації проекту. Цей вибір забезпечить високий рівень продуктивності, зручності та якості виконання завдань у рамках магістерської кваліфікаційної роботи.

3. ОБРОБЛЕННЯ РЕЗУЛЬТАТІВ СПОСТЕРЕЖЕНЬ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ У СЕРЕДОВИЩІ KAGGLE

3.1 Попередня обробка вхідного набору даних Kaggle

В основі розв’язання поставленої задачі будуть використовуватися набір даних, який містить статистичну інформацію про рослинництво у 173 країнах в Африці, Америці, Азії, Європи та Океанії. Датасет є публічним та опублікованим на платформі Kaggle [37].

Датасет із Kaggle, пов'язаний з аналізом і прогнозуванням виробництва сільськогосподарських культур за допомогою машинного навчання, містить важливі дані для сільськогосподарської галузі. Він включає інформацію про вирощування різних культур, а також супутні фактори, такі як погодні умови, ґрунти, добрива, зрошення та інші ключові показники, що впливають на врожайність.

Аналіз цього датасету може допомогти виявити закономірності та тренди у виробництві сільськогосподарських культур, оцінити вплив різних чинників на врожайність і використовувати ці знання для створення моделей прогнозування. Використання машинного навчання для цього типу даних дозволяє підвищити точність прогнозів, що важливо для сільськогосподарського планування та управління ресурсами.

Для вирішення задачі, поставленої в ході виконання роботи будуть використовуватися дані країн Європейського розташування, які включають такі параметри [37]:

–«area_code» – унікальний послідовний ідентифікатор, який присвоюється кожній країні окремо;

–«area» – назва країни;

–«item_code» – унікальний ідентифікатор сільськогосподарської продукції;

–«item» – назва сільськогосподарської продукції;

–«element_code» – тип даних, що збережені в масиві, в даному випадку буде три типи під кодами: 5312 – площа збору врожаю, 5419 – врожайність, 5510 – зібрано тон;

–«element» – розшифрування назви типу даних до його коду, відповідно наявно три незалежних індивідуальних типи;

–«unit» – одиниці вимірювання;

–«Y*xxxx*» – масив даних, де *xxxx* – це рік, по якому представлена інформація. Датасет розпочинається з 1961 року (дані мають великий відсоток пропусків) і закінчуються по 2019 рік, де зібрано багато інформації по всіх представлених країнах. На рисунку 3.1 показано вхідний набір даних з датасету.

# Area Code	Area	# Item Code	Item	# Element Code	Element	Unit	# Y1961
3 – 273	Spain 3% Bulgaria 3% Other (9866) 93%	15 – 1841	Potatoes 1% Roots and Tubers, ... 1% Other (10287) 97%	5312 – 5510	Production 36% Area harvested 35% Other (3091) 29%	tonnes 36% ha 35% Other (3091) 29%	0
11	Austria	89	Buckwheat	5312	Area harvested	ha	224.00000
11	Austria	89	Buckwheat	5419	Yield	hg/ha	11429.000
11	Austria	89	Buckwheat	5510	Production	tonnes	256.00000
57	Belarus	89	Buckwheat	5312	Area harvested	ha	
57	Belarus	89	Buckwheat	5419	Yield	hg/ha	
57	Belarus	89	Buckwheat	5510	Production	tonnes	
80	Bosnia and Herzegovina	89	Buckwheat	5312	Area harvested	ha	
80	Bosnia and Herzegovina	89	Buckwheat	5419	Yield	hg/ha	
80	Bosnia and Herzegovina	89	Buckwheat	5510	Production	tonnes	
27	Bulgaria	89	Buckwheat	5312	Area harvested	ha	
27	Bulgaria	89	Buckwheat	5510	Production	tonnes	
98	Croatia	89	Buckwheat	5312	Area harvested	ha	
98	Croatia	89	Buckwheat	5419	Yield	hg/ha	
98	Croatia	89	Buckwheat	5510	Production	tonnes	
167	Czechia	89	Buckwheat	5312	Area harvested	ha	
167	Czechia	89	Buckwheat	5419	Yield	hg/ha	
167	Czechia	89	Buckwheat	5510	Production	tonnes	
51	Czechoslovakia	89	Buckwheat	5312	Area harvested	ha	889.00000

Рисунок 3.1 – Вхідний набір даних врожайності країн Європи

У наборі даних, який обрано для дослідження в ході виконання роботи представлено десять тисяч п’ятсот п’ятдесят вісім записів, і водночас кожен запис містить сто двадцять п’ять колонок інформації. Вважається, що такий масив інформації є об’ємним і виконане дослідження з прогнозованості

врожайності буде мати високі показники точності. Додатково набір даних містить таблиці, де знаходиться інформації про врожайність сільськогосподарських культур інших континентів, що показано на рисунку 3.2.

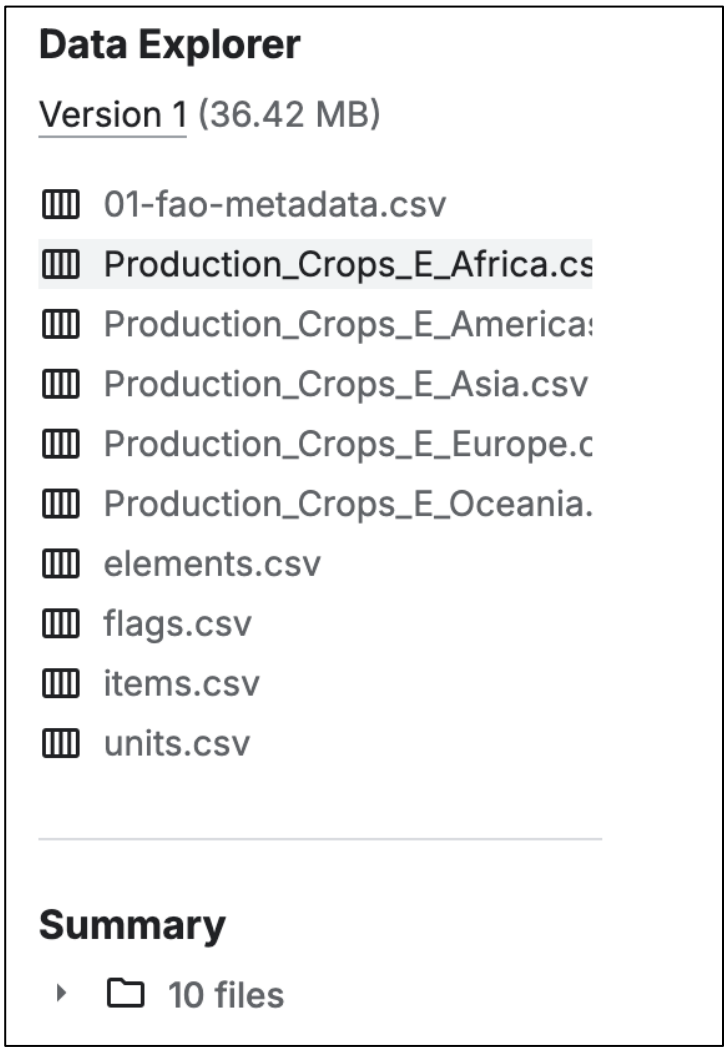


Рисунок 3.2 – Структура датасету «Crop Production»

На врожайність зернових культур впливає низка чинників, які варто детально контролювати. Погодні умови, пристосованість аграрних компаній до роботи із зерновою культурою, географічне розташування.

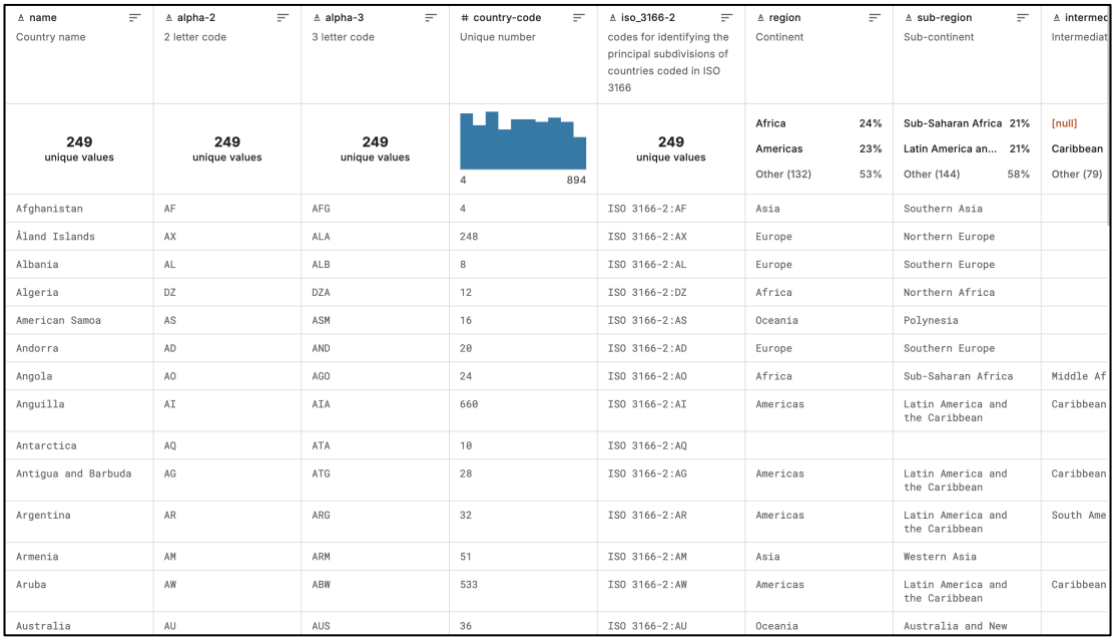
Для отримання прив’язки по розташуванню певної країни було використано відкриті дані з датасету, що міститься у Kaggle «Country Mapping - ISO, Continent, Region» [38]. Цей набір даних може бути використаний для аналізу географічного розподілу даних у різних дослідженнях, пов'язаних із

глобальними економічними, соціальними або демографічними аспектами. Структура датасету:

– «country» – назва країни;
– «ISO Alpha-2» – двосимвольний код країни за стандартом ISO;
– «ISO Alpha-3» – трисимвольний код країни за стандартом ISO;
– «ISO Numeric» – числовий код країни;
– «continent» – континент, на якому розташована країна;
– «region» – географічний регіон, до якого належить країна.

Набір даних можна використовувати для поділу даних за континентами або регіонами, допомагає у візуалізації даних на мапах світу з використанням відповідних кодів ISO. Важливою ознакою даного датасету є можливість інтеграції з іншими даними: Датасет легко інтегрується з іншими джерелами інформації для більш детального аналізу в різних сферах, як наприклад економіка, демографія, екологія. Саме ця властивість буде використана у ході виконання магістерської кваліфікаційної роботи.

Цей датасет є важливим інструментом для тих, хто хоче проводити регіональні дослідження або аналіз міжнародних даних, оскільки він забезпечує стандартизовану структуру даних для використання в широкому спектрі проектів. На рисунку 3.3 показано вміст датесету «Country Mapping - ISO, Continent, Region».



name Country name	alpha-2 2 letter code	alpha-3 3 letter code	country-code Unique number	iso_3166-2 codes for identifying the principal subdivisions of countries coded in ISO 3166	region Continent	sub-region Sub-continent	intermec Intermediat
249 unique values	249 unique values	249 unique values	4 894	249 unique values	Africa 24% Americas 23% Other (132) 53%	Sub-Saharan Africa 21% Latin America an... 21% Other (144) 58%	[null] Caribbean Other (79)
Afghanistan	AF	AFG	4	ISO 3166-2:AF	Asia	Southern Asia	
Åland Islands	AX	ALA	248	ISO 3166-2:AX	Europe	Northern Europe	
Albania	AL	ALB	8	ISO 3166-2:AL	Europe	Southern Europe	
Algeria	DZ	DZA	12	ISO 3166-2:DZ	Africa	Northern Africa	
American Samoa	AS	ASM	16	ISO 3166-2:AS	Oceania	Polynesia	
Andorra	AD	AND	20	ISO 3166-2:AD	Europe	Southern Europe	
Angola	AO	AGO	24	ISO 3166-2:AO	Africa	Sub-Saharan Africa	Middle Af
Anguilla	AI	AIA	660	ISO 3166-2:AI	Americas	Latin America and the Caribbean	Caribbean
Antarctica	AQ	ATA	10	ISO 3166-2:AQ			
Antigua and Barbuda	AG	ATG	28	ISO 3166-2:AG	Americas	Latin America and the Caribbean	Caribbean
Argentina	AR	ARG	32	ISO 3166-2:AR	Americas	Latin America and the Caribbean	South Ame
Armenia	AM	ARM	51	ISO 3166-2:AM	Asia	Western Asia	
Aruba	AW	ABW	533	ISO 3166-2:AW	Americas	Latin America and the Caribbean	Caribbean
Australia	AU	AUS	36	ISO 3166-2:AU	Oceania	Australia and New	

Рисунок 3.3 – Датасет «Country Mapping - ISO, Continent, Region»

В рамках дослідження для передбачення врожайності зернових культур було використано два датасети, доступні на платформі Kaggle. Перший датасет містить історичні дані про врожайність зернових культур за певний період. Він включає такі ключові ознаки, як рік збору врожаю, тип зернової культури, загальна площа посівів, обсяги врожайності, кліматичні показники (середньорічна температура, кількість опадів), а також дані про застосування агротехнічних практик.

Другий датасет використовується для геопросторової прив’язки даних до конкретних регіонів. Цей набір даних містить координати (широту і довготу), адміністративну приналежність території, а також додаткові регіональні характеристики, такі як тип ґрунту, рельєф, та інформація про середньорічну кількість сонячних днів. Використання такого підходу дозволяє інтегрувати кліматичні й геопросторові фактори, що суттєво впливають на врожайність.

Поєднання цих двох датасетів забезпечує багатовимірний аналіз даних, який враховує як часові, так і просторові фактори. Це створює основу для побудови моделей машинного навчання, що здатні враховувати складні взаємозв’язки між характеристиками регіонів, кліматичними умовами та агротехнічними практиками.

3.2 Візуалізація вхідних підготовлених даних

Для правильного та ефективного здійснення розвідувального аналізу та застосування алгоритмів для передбачення врожаю на основі попередніх даних потрібно створити ноутбук у онлайн системі Kaggle й завантажити усі необхідні для цього набори даних й бібліотеки Python. На рисунку 3.4 показано створений ноутбук з підключеними до нього датасетами для виконання роботи.

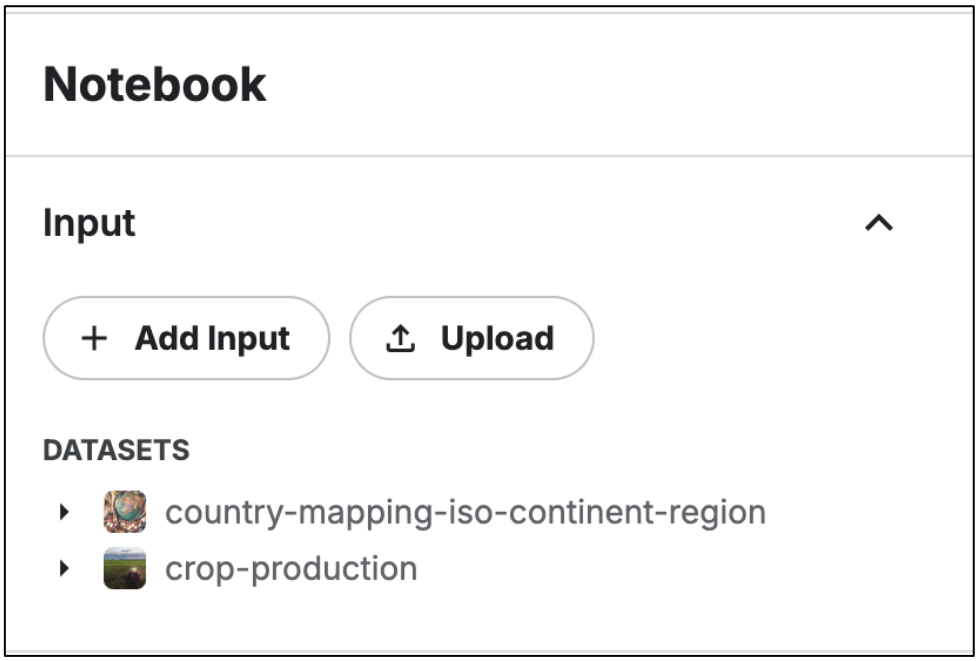


Рисунок 3.4 – Завантаження наборів даних для виконання задачі

Оскільки Kaggle надає зручне середовище для роботи з великими наборами даних і дозволяє інтерактивно виконувати кроки аналізу. Підготовка до роботи включала кілька основних етапів.

Спершу було імпортовано необхідні бібліотеки, які є стандартом для аналізу даних у Python. Для роботи з даними було використано Pandas та NumPy, що дозволяють ефективно обробляти великі таблиці та виконувати математичні операції. Для візуалізації результатів були використані Matplotlib і Seaborn, які забезпечують зручні інструменти для створення графіків та теплових карт. Оскільки дані включають геопросторову інформацію, для їх обробки було підключено бібліотеку GeoPandas, яка дозволяє працювати з географічними

координатами. Для побудови і оцінки моделей машинного навчання використовувалися інструменти з Scikit-learn. На рисунку 3.5 показано підключення бібліотеки у створеному ноутбуці Kaggle.

```
import pandas as pd

from arrow import now
from charset_normalizer import detect
from glob import glob

from warnings import filterwarnings
filterwarnings(action='ignore', category=FutureWarning)

CODES = '/kaggle/input/country-mapping-iso-continent-region/continents2.csv'
CROPS = '/kaggle/input/crop-production/Production_Crops_E_Europe.csv'
```

Рисунок 3.5 – Підключення та використання бібліотеки Pandas

Далі відбулося завантаження основних наборів даних із платформи Kaggle. Перший набір даних містить інформацію про врожайність зернових культур, а другий — про геопросторові дані, зокрема, координати, типи ґрунтів та кліматичні умови. Після завантаження даних була виконана перевірка на наявність пропущених значень або аномалій. Пропуски були заповнені середніми значеннями або модою в залежності від типу ознаки, а записи з некоректними або нечисловими значеннями були вилучені.

Один з важливих етапів — це об’єднання двох датасетів. Для цього використовувалася функція злиття (merge) в Pandas, що дозволила створити єдину таблицю для подальшого аналізу. Важливим моментом було нормалізувати географічні координати, щоб їх можна було зручно використовувати в моделюванні.

Для перевірки працездатності створеного ноутбуку та правильності підключення бібліотек було виконано тестовий запуск із кодом, який показано на рисунку 3.5.

```
import pandas as pd

from arrow import now
from charset_normalizer import detect
from glob import glob

from warnings import filterwarnings
filterwarnings(action='ignore', category=FutureWarning)

CODES = '/kaggle/input/country-mapping-iso-continent-region/continents2.csv'
CROPS = '/kaggle/input/crop-production/Production_Crops_E_Europe.csv'

def get_file(filename: str):
    with open(file=filename, mode='rb') as input_fp:
        encoding = detect(input_fp.read())['encoding']
        return pd.read_csv(filepath_or_buffer=filename, encoding=encoding)

time_start = now()
df = pd.concat(objs=[get_file(filename=input_file) for input_file in glob(CROPS)]).merge(right=pd.read_csv(filepath_or_buffe
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
        df[column] = df[column].fillna(value=0)
print('data read complete in {}'.format(now() - time_start))

df.info()
```

Рисунок 3.5 – Завантаження датасету в Notebook Jupyter

Результатом виконання було виведено інформацію, яка показує, що дані завантажуються швидко, отже ноутбук працює належним чином. Показано на рисунку 3.6.

```
data read complete in 0:00:00.495660
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8107 entries, 0 to 8106
Columns: 127 entries, Area Code to alpha-3
dtypes: float64(59), int64(3), object(65)
memory usage: 7.9+ MB
```

Рисунок 3.6 – Вихідний результат запущеного ноутбуку

Для перевірки роботи бібліотеки Pandas було запрограмовано тестовий код, який фільтрував дані датасету, залишаючи там лише інформацію, що стосується таких сільськогосподарських культур як «Пшениця» та «Соняшник».

На рисунку 3.7 (показано код Python), рисунку 3.8 (показано результат виконання фільтрації).

```python
# Read all files and combine them into a single DataFrame
df = pd.concat(objs=[get_file(filename=input_file) for input_file in glob(CROPS)])

# Read a file with codes and regions, leaving only the necessary columns
codes_df = pd.read_csv(filepath_or_buffer=CODES, usecols=['alpha-3', 'name'])

# Combine both DataFrames using merge, leaving only the rows with "wheat" in the "Item" column
df = df[df['Item'].isin(['Wheat', 'Sunflower seed'])].merge(right=codes_df, left_on='Area', right_on='name')

# Fill in the missing values in the appropriate columns
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
        df[column] = df[column].fillna(value=0)

# Display the time of operations
print('data read complete in {}'.format(now() - time_start))

# Filter the data, leaving only the rows where the 'Element' column is 'Yield'
df = df[df['Element'] == 'Yield']

# Print the first 5 lines of the resulting DataFrame
df.info()
df.head()
```

Рисунок 3.7 – Використання бібліотеки pandas

	Area Code	Area	Item Code	Item	Element Code	Element	Unit	Y1961	Y1961F	Y1962	...	Y2016	Y2016F	Y2017	Y2017F	Y2018	Y2018F	Y2019	Y2019F
1	3	Albania	267	Sunflower seed	5419	Yield	hg/ha	10000.0	Fc	10000.0	...	29000.0	Fc	22251.0	Fc	22803.0	Fc	23075.0	Fc
4	3	Albania	15	Wheat	5419	Yield	hg/ha	7731.0	Fc	10522.0	...	39000.0	Fc	40367.0	Fc	36927.0	Fc	40680.0	Fc
7	11	Austria	267	Sunflower seed	5419	Yield	hg/ha	19275.0	Fc	16765.0	...	32941.0	Fc	23336.0	Fc	28358.0	Fc	30372.0	Fc
10	11	Austria	15	Wheat	5419	Yield	hg/ha	25802.0	Fc	26122.0	...	62534.0	Fc	48712.0	Fc	46453.0	Fc	57372.0	Fc

Рисунок 3.8 – Результат компіляції програмного коду

Після очищення та інтеграції даних було здійснено першу візуалізацію для виявлення основних тенденцій і взаємозв'язків між різними параметрами. Для цього були створені графіки залежності врожайності від таких факторів, як температура, кількість опадів, а також інші кліматичні показники. Візуалізація допомогла краще зрозуміти вплив різних факторів на врожайність зернових культур у Європі та виявити ключові фактори, що потребують уваги при побудові моделі прогнозування.

Цей етап підготовки даних був необхідний для створення надійної основи для подальших етапів аналізу та моделювання, що дозволить отримати точніші

прогнози врожайності. На рисунку 3.9 показано кількість країн у яких вирощувався обрана культура в період з 1961 року по 2019 рік.

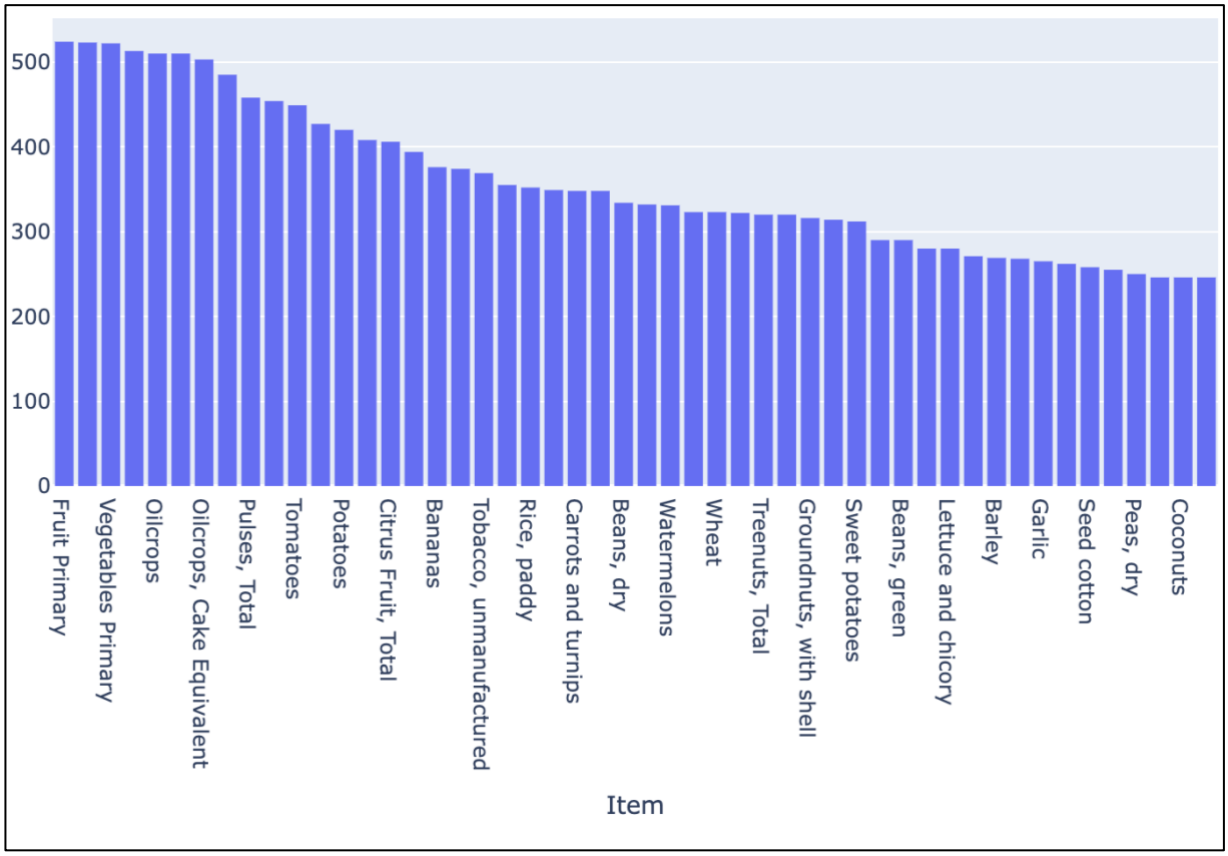


Рисунок 3.9 – Візуалізація кількості країн, які вирощували обраний продукт

З виконаної візуалізації можна спостерігати закономірність, що родина фруктових, найбільше поширена у світі, й вирощується понад як в 500 країнах світу, овочі знаходяться майже на такому самому рівні, а от часник вирощується лише в половини країн, що каже про те, що цей продукт не є таким поширеним у світі і для нього потрібні відповідні погодні умови, для отримання позитивного врожаю.

3.3 Висновки

Під час виконання третього розділу було обґрунтовано доцільність використання онлайн сервісу Kaggle. Оброблено вхідні набори даних з використанням мови програмування Python, здійснено попередню обробку інформації. Використано можливості бібліотеки Pandas та зроблено поєднання двох наборів даних, а також виконано налаштування фільтрації лише потрібних даних. На основі підготовленої інформації було виконано візуалізацію.

Отримана візуалізація показала, що датасет містить широкий набір актуальних даних про вирощування різноманітних сільськогосподарських культур й країн де вони збираються. Це свідчить про те, що вхідні дані є повноцінними для виконання поставленої задачі в магістерській кваліфікаційній роботі.

Підготовлена інформація буде в подальшому використовуватися для виконання розвідувального аналізу даних, а також для навчання моделей передбачення даних.

4. РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УЗАГАЛЬНЕННЯ ДАНИХ ВИРОЩУВАННЯ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР У ЄВРОПІ

4.1 Розвідувальний аналіз врожайності сільськогосподарських країн на території Європи

Було проведено розвідувальний аналіз, на основі якого буде виконано передбачення врожайності зернових культур на території Європи.

Вхідні дані було відфільтровано за такими параметрами, як: зернова культура – пшениця, ключова ознака – врожайність. Виконано перевірку між параметрами значень по роках. На рисунку 4.1 наведено Heatmap, можна спостерігати, що відношення кореляції даних є незначним від 1961 по 1991 роки. Велику зміну дана сфера зазнала після 2000-х років, спостерігається стрімке зростання врожайності, яке поступово зростає щороку.

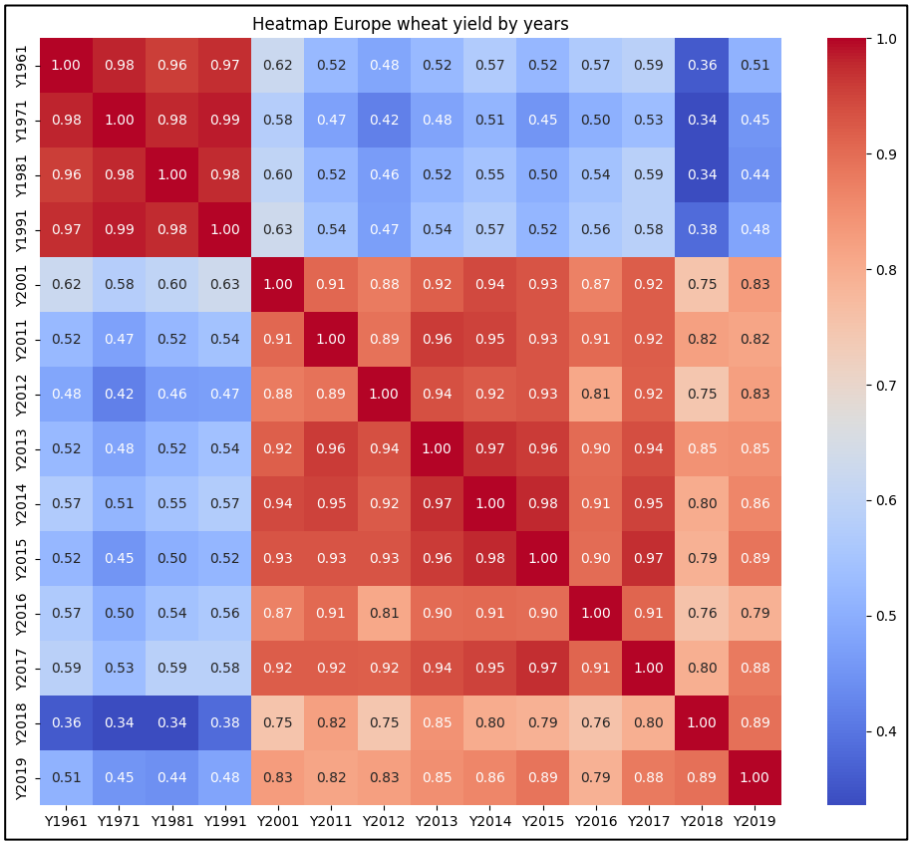


Рисунок 4.1 – Heatmap врожайності пшениці у Європі

На рисунку 4.2 зображено графік, який показує кількість різних культур, що вирощувалися за ввесь час в обраній країні. До країн, які мають найбільшу різноманітність вирощуваних сільськогосподарських культур відносяться: Іспанія, Італія, Греція, Франція, Україна.

Рисунок 4.2 – Кількість культур вирощуваних в країнах Європи

На рисунку 4.3 зображено відношення посівних площ соняшника та пшениці за 2019 рік на території Європи. На діаграмі чітко видно межу посівних площ соняшника та пшениці території України порівняно з усією Європою. Отже, Україна забезпечує більшу частину сировини на глобальному ринку.

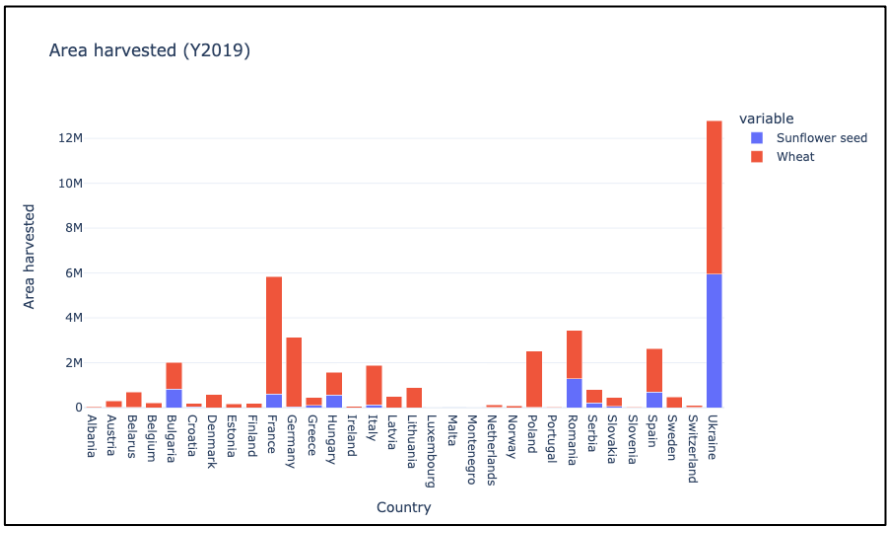


Рисунок 4.3 – Посівні площі соняшника та пшениці за 2019 рік

На рисунку 4.4 зображено порівняння посівної площі кукурудзи в Україні до всієї посівної площі кукурудзи країн, які територіально розташовані в Європі. Україна це аграрна країна, на з отриманого графіка можна побачити, що по збору кукурудзи Україна має високу конкуренцію і може прямо виставляти правила формування цінової політики кукурудзи на Європейському ринку.

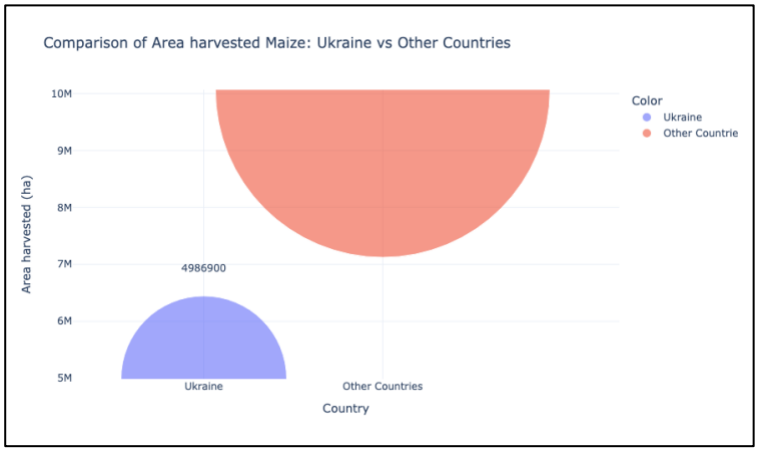


Рисунок 4.4 – Відношення посівних площ кукурудзи

На рисунку 4.5 зображено найвищий врожай ріпаку визначеного року. З графіка видно, що Бельгія є спроможним конкурентом на глобальному ринку і використовує новітні технології, що дають змогу отримати високий рівень врожаю ріпака.

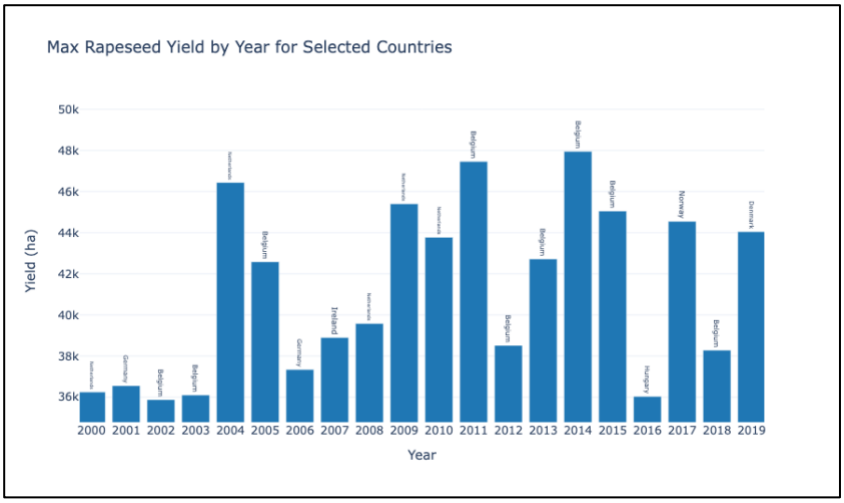


Рисунок 4.5 – Аналіз найвищого врожаю ріпаку за роками

На рисунку 4.6 показано порівняння популярності зернових культур, які вирощуються в Європі. Найпопулярнішою є пшениця, яка вирощується в усіх країнах Європи, на другому місці знаходиться ріпак, який вирощується у 28 країнах з 32, третє місця по кількості займає кукурудза – вирощується в 26 країнах з 32, і на останньому місця знаходиться соняшник, його вирощують лише 20 країн.

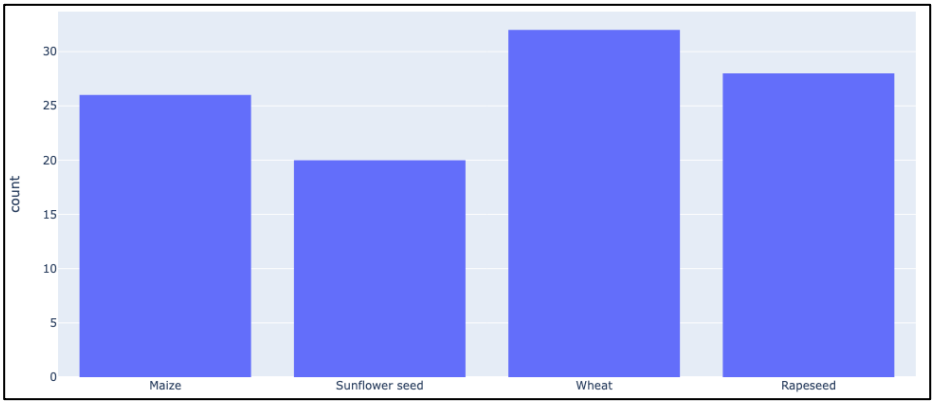


Рисунок 4.6 – Рейтинг поширення зернових культур в Європі

З використанням можливостей бібліотеки plotly та компоненту choropleth з вхідних даних датасету було побудовано візуалізацію, на якій показано посівні площі соняшника окремих країн Європи. Результат показано на рисунку 4.7. Можна побачити, що Україна обробляє велику площу землі та вирощує великі об’єми такої складної культури як соняшник. Зроблено висновок, що основне забезпечення сировиною та готовою продукцією в Європі соняшником дає Україна.

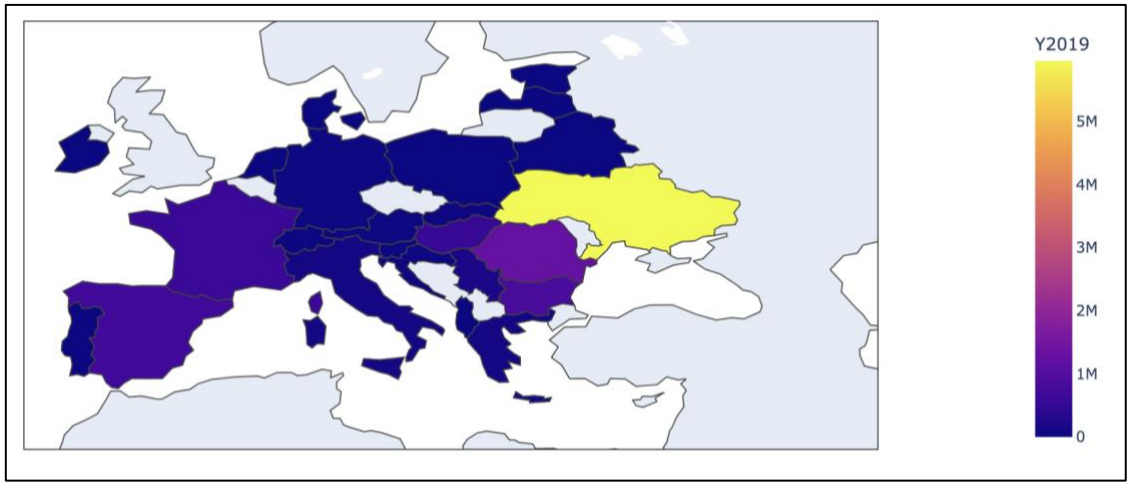


Рисунок 4.7 – Площі вирощування соняшника на території Європи

На рисунку 4.8 показано врожайність соняшника, яку отримують країни з посівних площ. Можна побачити, що результати України знаходяться середньому показнику в порівнянні з усіма країнами Європи. Це свідчить про те, що аграрні компанії використовують не такий якісний комплексний технологічний підхід, як в окремих країнах Європи, отже є стимул до навчання та опанування новітніх методів вирощування соняшника.

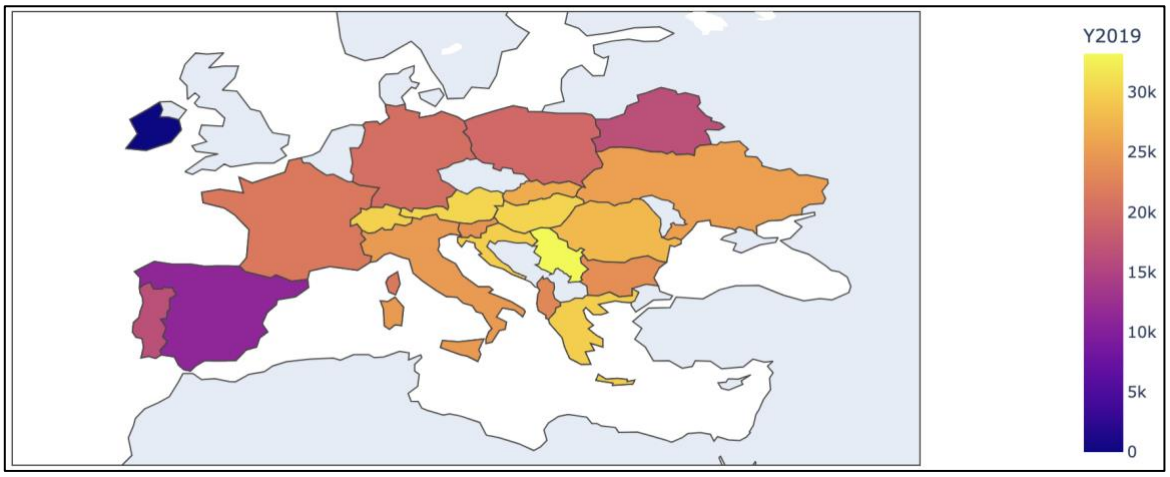


Рисунок 4.8 – Середня врожайність (ц) на гектар соняшника за 2019 рік

З датасету було візуалізовано інформацію про валовий збір врожаю соняшника за 2019 рік в країнах Європи, можна наглядно побачити, що Україна отримує своє місце на ринку і забезпечує всю територію Європи соняшниковою продукцією. Результат візуалізації показано на рисунку 4.9.

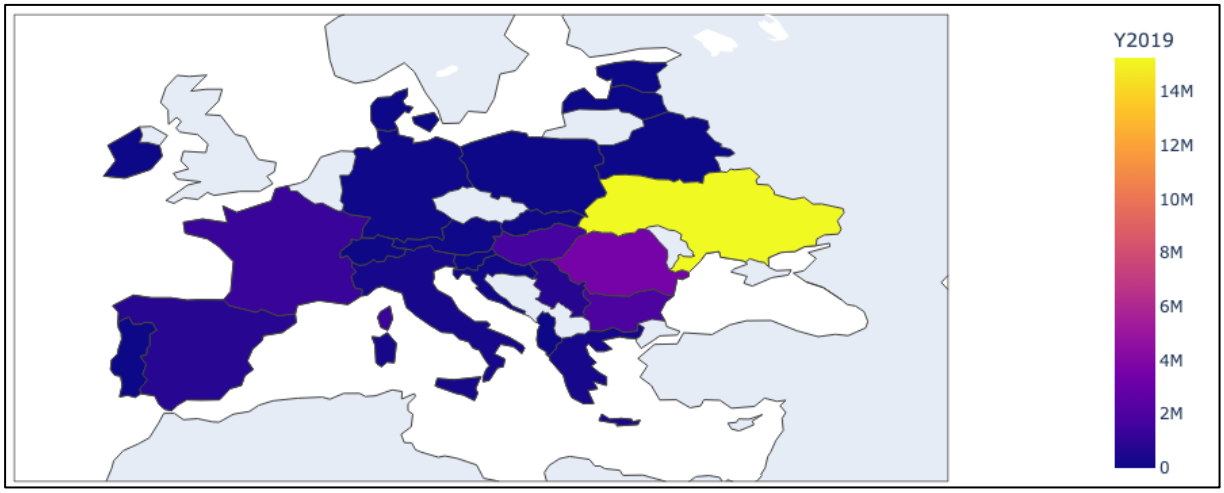


Рисунок 4.9 – Валовий збір соняшника за 2019 рік

На рисунку 4.10 показано середню врожайність пшениці країн Європи, що хронологічно розподілено по роках.

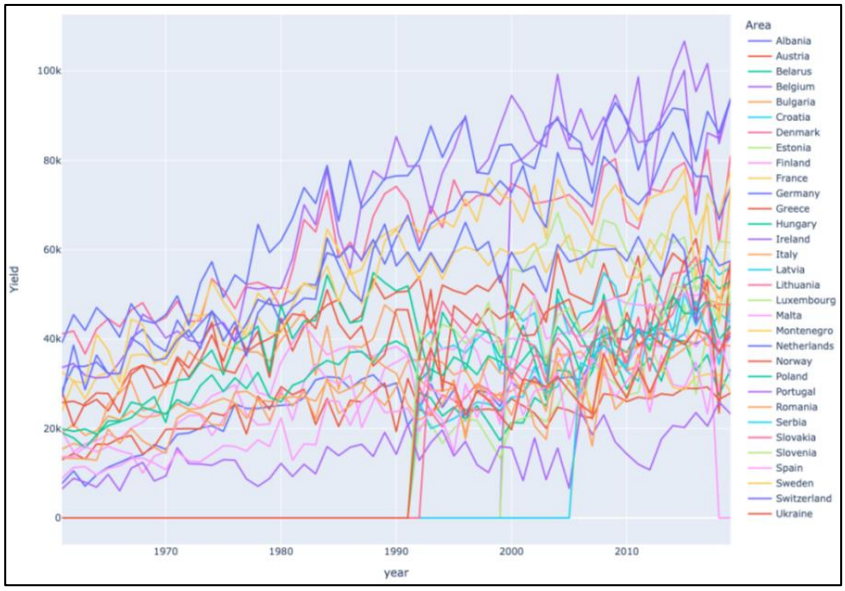


Рисунок 4.10 – Загальна середня врожайність пшениці країн Європи

З рисунка спостерігається наглядна закономірність графіків в кожної країни, які є подібними, це свідчить, що відбувається виснаження ґрунту за сівозмінну, і не можливо постійно отримувати високий показник врожаю, потрібно постійно змінювати попередника для відновлення спроможності ґрунтів, що й спостерігається на рисунку 4.10.

4.2 Вибір оптимальних моделей

Ефективне вирішення поставленої задачі магістерської роботи потребує правильне визначення оптимальної моделі для отримання точного передбачення даних. Було виконано аналіз наявних популярних моделей для машинного передбачення даних й визначено три моделі, які будуть використані в ході виконання роботи: XGBoostRegressor, DecisionTreeRegressor, RandomForestRegressor.

Extreme Gradient Boosting (XGBoost) [39] — це бібліотека з відкритим кодом, яка забезпечує ефективну реалізацію алгоритму градієнтного посилення. XGBoost став популярним методом і часто ключовим компонентом у розв'язанні низки проблем у змаганнях з машинного навчання. Алгоритм роботи даної бібліотеки показано на рисунку 4.11 [40].

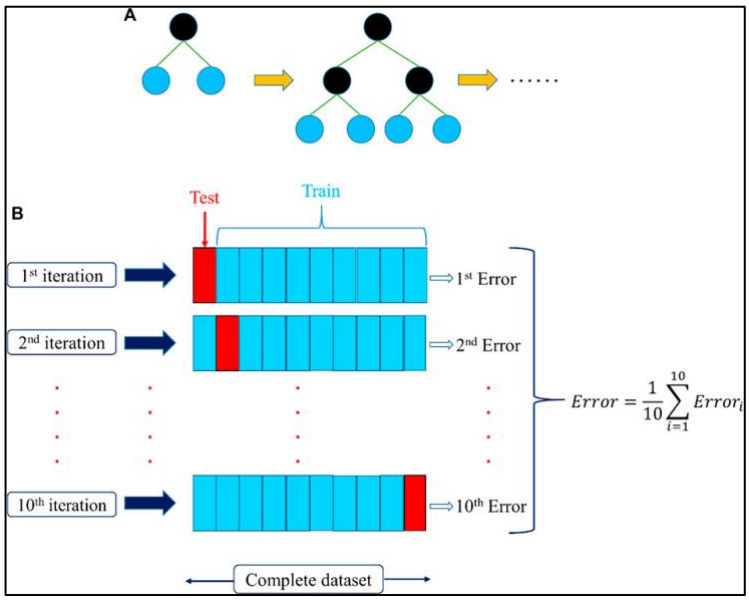


Рисунок 4.11 – Структура роботи моделі XGBoostRegressor

Підсилення градієнта відноситься до класу ансамблевих алгоритмів машинного навчання, які можна використовувати для задач класифікації або регресійного прогнозного моделювання [41].

Ансамблі будуються з моделей дерева рішень. Дерева додаються по одному в ансамбль і підходять для виправлення помилок передбачення, зроблених попередніми моделями. Це тип моделі ансамблевого машинного навчання, який називається форсуванням. Двома основними причинами використання XGBoost є швидкість виконання та продуктивність моделі.

XGBoost домінує в структурованих або табличних наборах даних щодо проблем класифікації та регресійного прогнозного моделювання [42].

DecisionTreeRegressor є моделлю машинного навчання, яка використовується для задачі регресії. Це означає, що вона призначена для передбачення числових значень, таких як ціна на нерухомість, кількість продуктів, проданих за день, або будь-який інший неперервний вимір. Основна ідея дерев рішень полягає в тому, щоб розділити набір даних на менші підгрупи, поки не буде досягнуто визначеного критерію зупинки, наприклад, максимальної глибини дерева чи мінімального числа зразків у вузлі. До основних характеристик даної моделі можна віднести [43]:

1. Розділення даних: Дерево рішень розділяє дані на групи на основі значень характеристик.

2. Критерії розділення: Використовуються різні критерії, наприклад, середньоквадратична помилка, для визначення способу розділення даних вузлів дерева.

3. Побудова дерева: Дерево будується зверху вниз, починаючи з кореневого вузла і розділяючи дані на кожному рівні.

4. Передбачення: Після побудови дерева можна використовувати його для передбачення нових значень, пройшовши від кореневого вузла до листового.

5. Параметризація: Дерева рішень мають багато параметрів для налаштування, таких як максимальна глибина дерева і мінімальна кількість вибірок у вузлі, що дозволяє оптимізувати модель.

Є частиною бібліотеки scikit-learn для мови програмування Python і є корисним інструментом для різних завдань регресії, де важливо здійснювати прогнози числових значень на основі вхідних даних.

RandomForestRegressor – це широко використовуваний алгоритм машинного навчання, розроблений Лео Брейманом та Адель Катлером, який комбінує результати кількох дерев рішень для отримання єдиного прогностичного результату. Його простота використання та гнучкість сприяли популярності, оскільки він ефективно вирішує завдання як класифікації, так і регресії. Алгоритми випадкового лісу мають три основні гіперпараметри, які необхідно встановити перед навчанням. До них належать розмір вузла, кількість дерев і кількість відібраних функцій. Звідти класифікатор випадкового лісу можна використовувати для вирішення проблем регресії або класифікації [44]. На рисунку 4.12 [44] показано алгоритм роботи моделі RandomForestRegressor.

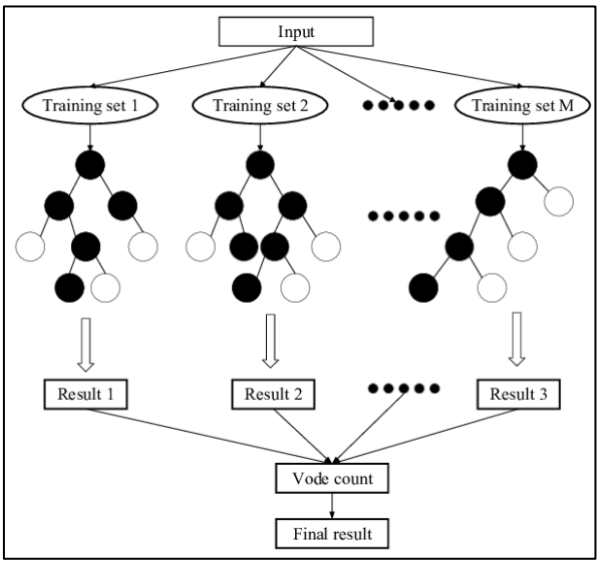


Рисунок 4.12 – Принцип роботи RandomForestRegressor

Алгоритм випадкового лісу складається з набору дерев рішень, і кожне дерево в ансамблі складається із вибірки даних, отриманої з навчального набору із заміною, яка називається початковою вибіркою. З цієї навчальної вибірки одна третина відкладена як тестові дані, відомі як вихідна вибірка. Інший випадок прогнозу вводиться через пакетування функцій, додаючи більше різноманітності до набору даних і зменшуючи кореляцію між деревами рішень. Залежно від типу проблеми визначення прогнозу буде різним. Модель надає можливість оцінити

важливість ознак в прогнозуванні, що дозволяє вибрати найбільш інформативні характеристики для покращення моделі [45].

4.3 Розроблення технології передбачення врожайності сільськогосподарських культур

Для розв’язання поставленої задачі потрібно виконати такі дії [46]:

– виконати аналіз та вибірку даних з датасету;

– зробити формування тренувального, валідаційного й тестового набору даних, які будуть використовуватися для обраних моделей передбачення;

– виконати тренування на навчальному датасеті. Провести передбачення за кожною моделлю окремо на основі валідаційних даних. Вибрати оптимальну модель;

– провести передбачення за ідентифікованою моделлю тестових даних;

– виконати оцінку точності кінцевого прогнозу, який було отримано після застосування моделей;

Дотримання вищеописаних пунктів гарантує, що в кінцевому результаті буде отримано позитивний результат з розв’язанням поставленої задачі в ході магістерської роботи. Формування висновку проведеного дослідження та його аналіз з виділенням ключових факторів для покращення роботи.

4.3.1 Вибір даних з датасету

Для передбачення врожайності сільськогосподарських культур потрібно підготувати вхідні дані, насамперед необхідно виділити цільову ознаку, виберемо дані врожайності за рік Y2019. Наступним кроком потрібно обрати ознаки, які найбільше впливають на цей параметр, основними чинниками є Item_Code та Element_Code, також потрібно додати до чинників дані за попередні роки врожайності, таким чином точність передбачення даних буде вищою. Адже після аналізу теплової карти було наглядно видно залежність врожайності по рокам. Отже, формування масиву даних та лістинг коду для навчальної та тестової вибірки даних показано на рисунках 4.13 та 4.14.

```
train = df[['Area Code', 'Item Code', 'Y2017', 'Y2018', 'Y2019']].copy()
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10557 entries, 0 to 10556
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Area Code  10557 non-null  int64
 1   Item Code  10557 non-null  int64
 2   Y2017      7626 non-null   float64
 3   Y2018      7963 non-null   float64
 4   Y2019      8006 non-null   float64
dtypes: float64(3), int64(2)
memory usage: 412.5 KB
```

Рисунок 4.13 – Лістинг коду на мові Python для вибору навчальної вибірки

```
test = df[['Area Code', 'Item Code', 'Y2017', 'Y2018']].copy()
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10557 entries, 0 to 10556
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Area Code  10557 non-null  int64
 1   Item Code  10557 non-null  int64
 2   Y2017      7626 non-null   float64
 3   Y2018      7963 non-null   float64
dtypes: float64(2), int64(2)
memory usage: 330.0 KB
```

Рисунок 4.14 – Лістинг коду на мові Python для вибору тестової вибірки даних

4.3.2 Формування навчального та валідаційного набору даних

На рисунку 4.15 зображено лістинг коду на мові Python з формуванням навчального та валідаційного набору даних.

```
# Вибір ознак (features) та цільової змінної (target)
features = ['Area Code','Item Code', 'Y2017', 'Y2018']
target = 'Y2019'
# Розділення даних на ознаки та цільову змінну
train_all = df[features]
target_all = df[target]
# Розділення на навчальні та тестові дані (80% на навчання, 20% на тестування)
train, valid, target_train, target_valid = train_test_split(train_all, target_all, test_size=0.2, random_state=42)
# Виведення результатів
print("Навчальні дані (X_train):")
print(train.info())
#print("\nНавчальні мітки (target_train):")
#print(target_train.info())
print("\nТестові дані (valid):")
print(valid.info())
#print("\nТестові мітки (target_valid):")
#print(target_valid.info())
```

```
Навчальні дані (X_train):
<class 'pandas.core.frame.DataFrame'>
Index: 8445 entries, 6275 to 7270
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Area Code  8445 non-null   int64
 1   Item Code  8445 non-null   int64
 2   Y2017      6136 non-null   float64
 3   Y2018      6386 non-null   float64
dtypes: float64(2), int64(2)
memory usage: 329.9 KB
None

Тестові дані (valid):
<class 'pandas.core.frame.DataFrame'>
Index: 2112 entries, 1950 to 1613
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Area Code  2112 non-null   int64
 1   Item Code  2112 non-null   int64
 2   Y2017      1490 non-null   float64
 3   Y2018      1577 non-null   float64
dtypes: float64(2), int64(2)
memory usage: 82.5 KB
None
```

Рисунок 4.15 – Формування навчального та валідаційного набору даних

4.3.3 Тренування та передбачення моделей

Для застосування обробки інформації та виконання завдання буде використано три моделі для передбачення даних за допомогою машинного навчання: XGBoostRegressor, DecisionTreeRegressor та RandomForestRegressor. Код, який відповідає за тренування та відтворення моделей показано на рисунках 4.16, 4.17, 4.18.

```
# Розбиваємо дані на навчальний та тестувальний набори
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)

# Ініціалізуємо модель XGBoostRegressor
model = XGBRegressor(
    max_depth=3,
    learning_rate=0.1,
    n_estimators=100,
    objective='reg:squarederror',
    booster='gbtree',
    random_state=42
)

# Навчаємо модель на навчальних даних
model.fit(X_train, y_train)

# Передбачаємо значення для навчальних та тестувальних даних
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Оцінюємо середньоквадратичну помилку (MSE) для навчальних та тестувальних даних
mse_train = mean_squared_error(y_train, y_train_pred)
mse_test = mean_squared_error(y_test, y_test_pred)
print(f"Train Mean Squared Error: {mse_train}")
print(f"Test Mean Squared Error: {mse_test}")

# Оцінюємо точність моделі (R²) для навчальних та тестувальних даних
r2_train = r2_score(y_train, y_train_pred)
r2_test = r2_score(y_test, y_test_pred)
accuracy_train = r2_train * 100
accuracy_test = r2_test * 100
print(f"Train Model Accuracy: {accuracy_train:.2f}%")
print(f"Test Model Accuracy: {accuracy_test:.2f}%")
```

Рисунок 4.16 – Тренування та передбачення XGBoostRegressor

```python
# Розділення даних на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Створення моделі DecisionTreeRegressor
model = DecisionTreeRegressor(random_state=42)

# Навчання моделі на навчальній вибірці
model.fit(X_train, y_train)

# Передбачення на навчальній та тестовій вибірках
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Оцінка моделі за допомогою метрики середньоквадратичної помилки (MSE) та R²
train_mse = mean_squared_error(y_train, y_train_pred)
train_r2 = r2_score(y_train, y_train_pred)
test_mse = mean_squared_error(y_test, y_test_pred)
test_r2 = r2_score(y_test, y_test_pred)

print(f"Test Mean Squared Error: {test_mse}")
print(f"Test R²: {test_r2:.2f} ({test_r2 * 100:.2f}%)")
```

Рисунок 4.17 – Тренування та передбачення DecisionTreeRegressor

```python
# Розподіл даних на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=0)

# Ініціалізація моделі RandomForestRegressor
model = RandomForestRegressor(n_estimators=100, random_state=0)

# Навчання моделі на навчальних даних
model.fit(X_train, y_train)

# Прогнозування на навчальних даних
y_train_pred = model.predict(X_train)

# Прогнозування на тестових даних
y_test_pred = model.predict(X_test)

# Оцінка точності моделі за допомогою середньоквадратичної помилки (MSE) для навчальних даних
train_mse = mean_squared_error(y_train, y_train_pred)
# Оцінка точності моделі за допомогою середньоквадратичної помилки (MSE) для тестових даних
test_mse = mean_squared_error(y_test, y_test_pred)
# Оцінка точності моделі за допомогою коефіцієнта детермінації (R²) для навчальних даних
train_r2 = r2_score(y_train, y_train_pred)
# Оцінка точності моделі за допомогою коефіцієнта детермінації (R²) для тестових даних
test_r2 = r2_score(y_test, y_test_pred)

# Переведення R² в точність у відсотках
train_accuracy = train_r2 * 100
test_accuracy = test_r2 * 100

print(f"Train Accuracy: {train_accuracy:.2f}%")
print(f"Test Accuracy: {test_accuracy:.2f}%")
```

Рисунок 4.18 – Тренування та передбачення RandomForestRegressor

4.3.4 Оцінювання точності передбачення

Для оцінки якості побудови моделей та визначення відсутності перенавчання було сформовано порівняльну таблицю 4.1, на якій показано результат виконання трьох моделей передбачення із введеними параметрами двадцять відсотків тестових даних для навчального та валідаційного набору.

Таблиця 4.1 – Результат моделей передбачення з 20% тестових даних

	Модель	**train_score**	**valid_score**
1	XGBoostRegressor	87.76%	95.94%
2	DecisionTreeRegressor	89.33%	93.86%
3	RandomForestRegressor	90.24%	91.61%

Для оцінки точності моделі у задачі регресії не підходить традиційне поняття «точності у відсотках», яке використовується у задачах класифікації. Однак, ми можемо скористатися коефіцієнтом детермінації (R^2), який показує, яка частка дисперсії залежної змінної пояснюється незалежними змінними моделі. Значення R^2 варіюється від 0 до 1, де 1 означає, що модель повністю пояснює варіативність даних.

У результаті зведення даних у таблиці 4.1 було отримано такі результати: найвища точність передбачення спостерігається під час використання XGBoostRegressor – 95.94%. Загалом результат трьох моделей має високу оцінку та стабільність результатів, отже перенавчання не відбувається. Параметри моделей та вхідні дані підібрано правильно. Для розширеного аналізу роботи моделей передбачення, було змінено відсоток тестових даних, отриманий результат наведено у таблиці 4.2.

Таблиця 4.2 – Точність моделей з різним відсотком тестових даних

	10%	20%	30%	40%
XGBoostRegressor	95.91	95.94	96.09	95.28
DecisionTreeRegressor	98.72	93.86	84.53	97.47
RandomForestRegressor	96.31	93.86	95.5	91.7

Зведені дані у таблиці показують, що моделі були добре навчені, також підлаштовуються під різний відсоток даних і дають високі показники точності.

Криві навчання (learning curves) – це графіки, які показують, як продуктивність моделі змінюється залежно від розміру навчального набору даних. Вони допомагають візуально оцінити стабільність і узагальнювальну здатність моделі, а також виявити проблеми, пов'язані з перенавчанням (overfitting) або недонавчанням (underfitting). На рисунку 4.19 показано графік, який демонструє ефективність навченої моделі XGBoost.

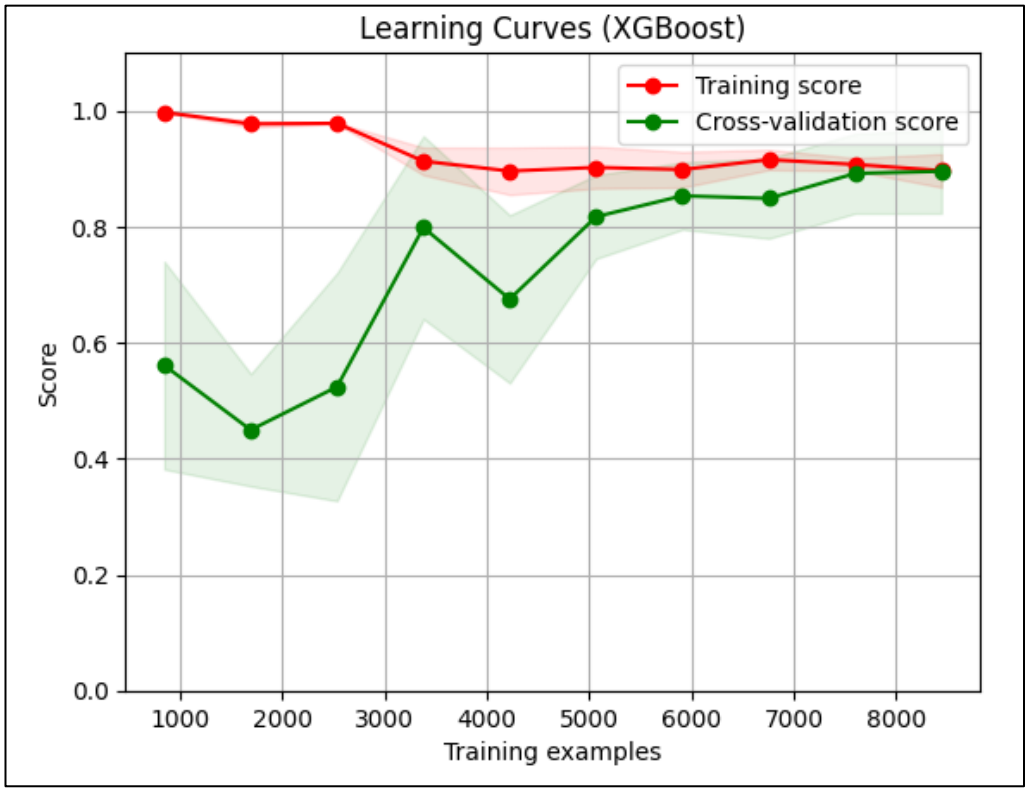


Рисунок 4.19 – Криві навчання для моделі XGBoostRegressor

Червона крива показує продуктивність моделі на навчальних даних. Зелена крива показує продуктивність моделі на валідаційних (крос-валідаційних) даних. Області навколо кривих показують стандартне відхилення продуктивності моделі на різних розбиттях даних. Графіки кривих є достатньо далеко один від одного, що свідчить про перенавчання досліджуваної моделі.

На рисунку 4.20 показано графіки розсіювання (scatter plots) для фактичних і передбачених врожайності сільськогосподарських культур на навчальних і тестових даних.

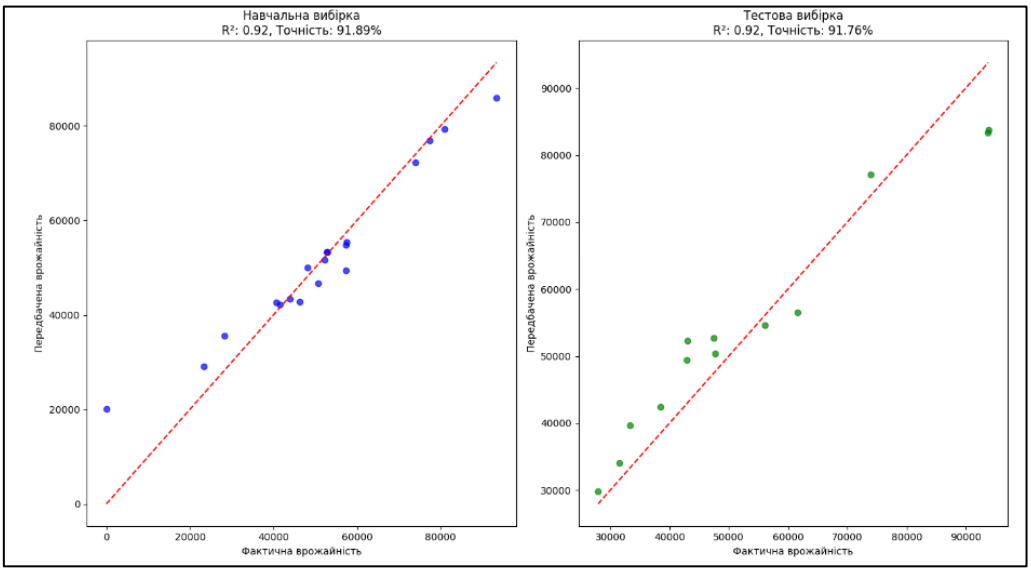


Рисунок 4.20 – Передбачення для валідаційних даних

Додано лінію ідеального передбачення (червона пунктирна лінія), де передбачені значення дорівнюють фактичним. Графіки допомагають візуально оцінити, наскільки добре модель RandomForestRegressor прогнозує врожайність на навчальних і тестових вибірках.

Результат навчання моделі DecisionTreeRegressor показано на рисунку 4.21.

–синя лінія – реальні значення цільової змінної (y_test);

–зелена лінія – прогнозовані значення (y_test_pred);

–візуалізація – легко порівняти реальні та передбачені значення для кожного індексу тестової вибірки.

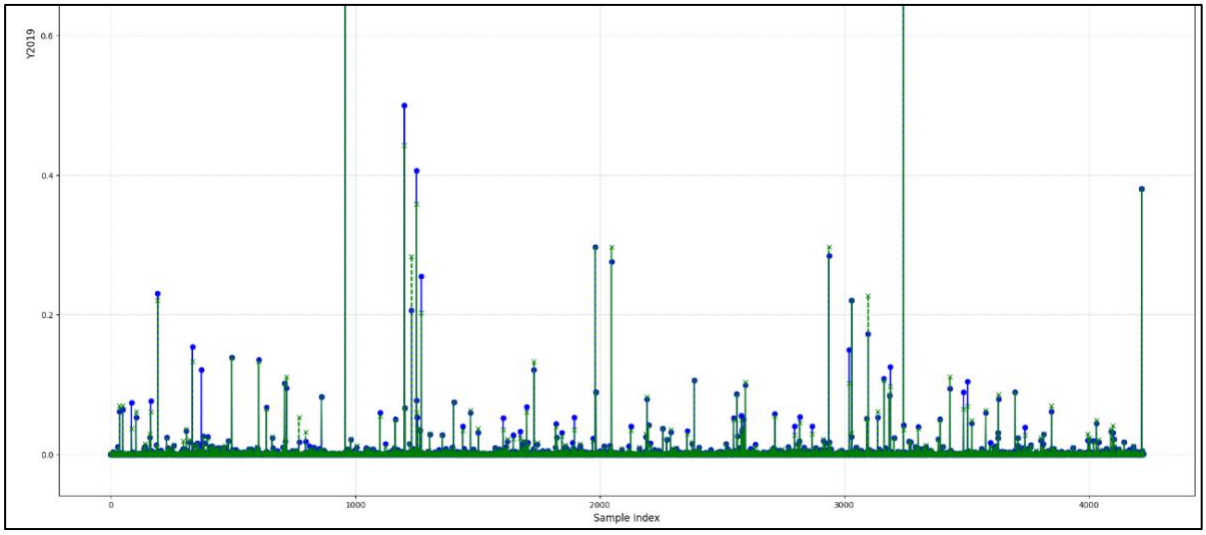


Рисунок 4.21 – Візуалізація результату навчання DecisionTreeRegressor

Графік важливості ознак візуалізує, які саме масиви даних вносять найбільший внесок у прийняття рішень моделлю XGBRegressor. Це важливо для розуміння того, як модель робить свої прогнози, а також для ідентифікації найбільш значущих факторів, що впливають на результат. Було протестовано передбачення на різних наборах даних і можна зробити висновок, що модель обирає схожий рік по врожайності, серед наявних і починає відтворювати прогноз даних. На рисунках 4.22, 4.23 показано два випадки, де візуально спостерігається така закономірність.

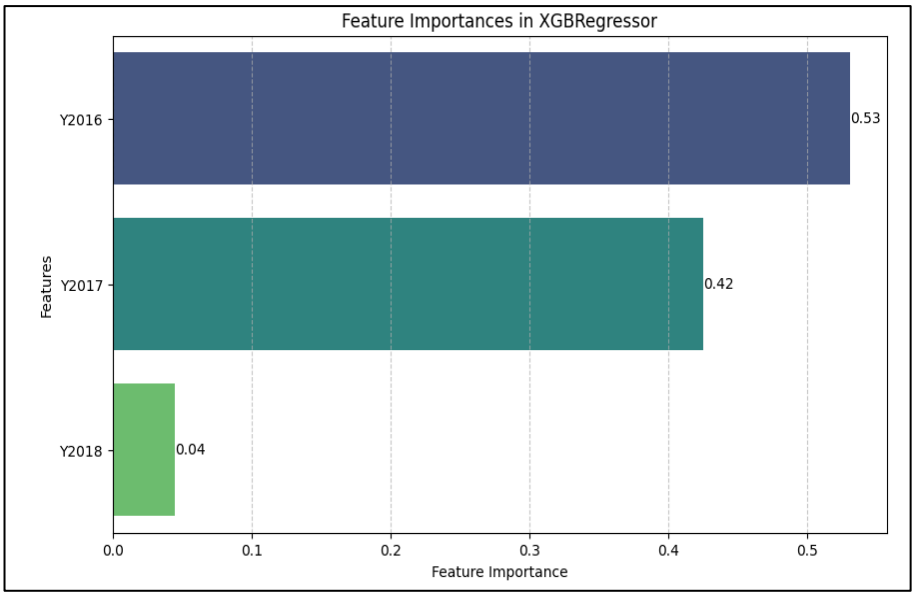


Рисунок 4.22 – Графік важливості ознак XGBRegressor (варіант 1)

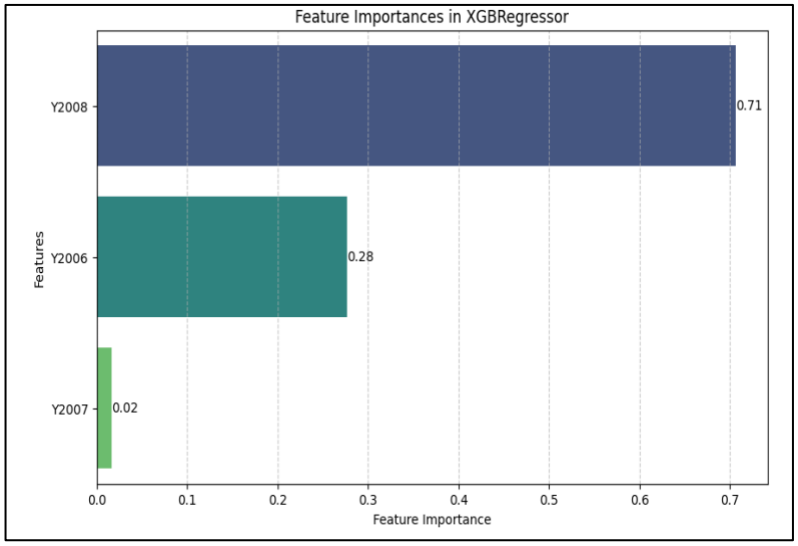


Рисунок 4.23 – Графік важливості ознак XGBRegressor (варіант 2)

4.4 Розроблення інформаційної технології узагальнення даних вирощування сільськогосподарських культур у Європі

Провелася робота із даними на платформі Kaggle, де вдалося проаналізувати значний обсяг інформації, пов'язаної з врожайністю зернових культур у різних регіонах Європи. Використовуючи Python та відповідні бібліотеки (Pandas, NumPy, Matplotlib, Seaborn), виконано попередню обробку даних, зокрема очищення, нормалізацію та виявлення ключових залежностей між параметрами.

Після аналізу створено три моделі машинного навчання, серед яких були регресійні моделі, алгоритми випадкових лісів (Random Forest) та градієнтного бустингу (XGBoost). Це дозволило отримати прогнози врожайності на основі наявних даних, а також оцінити точність та ефективність кожної з моделей.

Результати дослідження підтвердили можливість ефективного застосування інформаційної технології для узагальнення даних та прогнозування врожайності зернових культур.

Для представлення розробленого алгоритму було виконано формування UML-діаграми, зокрема діаграму діяльності, яка демонструє послідовність етапів роботи: від збору даних до створення прогнозів. Це допомогло

систематизувати розробку і показати логіку виконаних кроків. На рисунку 4.24 показано блок-схема алгоритму інформаційної технології.

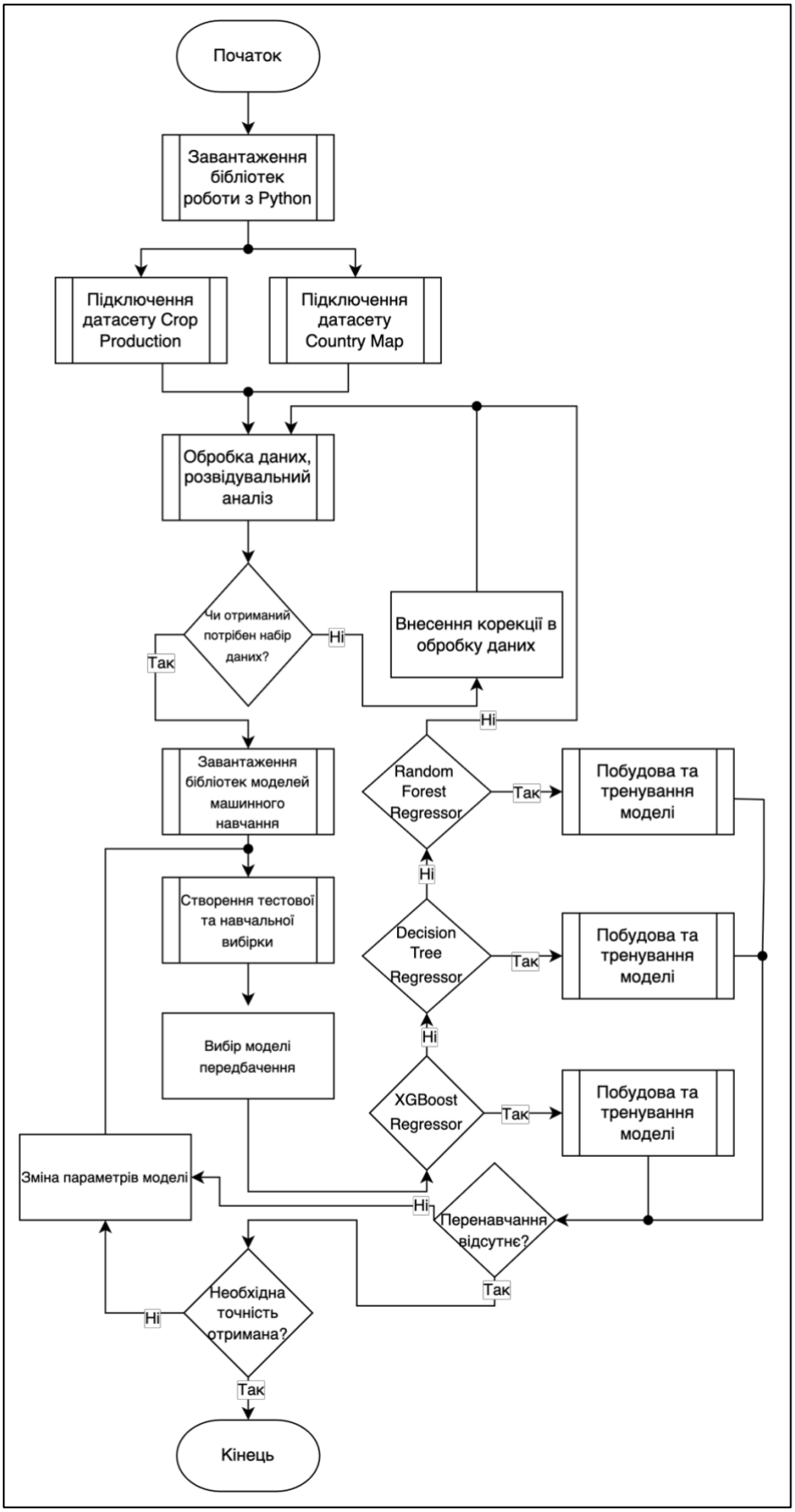


Рисунок 4.24 – Блок-схема алгоритму роботи інформаційної технології

Готова інформаційна технологія працює за таким алгоритмом (див. рисунок 4.24):

1. Відбувається завантаження усіх необхідних бібліотек Python та компонентів, які будуть використовуватися в процесі аназілу.

2. Підключення, налаштування та перевірка датасетів, які використовуються як вхідні дані для інформаційної технології.

3. Підготовка масиву даних, виконання розвідувального аналізу завантаженої інформації.

4. Завантаження бібіліотек для роботи з машинним навчанням.

5. Створення навчальної та тестової вибірки з підготовлених даних.

6. Вибір та запуск необхідної моделі машинного навчання.

7. Перевірка критеріїв відповідності отриманого результату, якщо все відповідає зазначеним вимогам, кінцевий результат можна використовувати в подальшому дослідженні, в іншому випадку користувач повинен провести зміну параметрів моделі і вернутися до пункту 5 алгоритму інформаційної технології.

4.5 Висновки

Під час виконання четвертого розділу магістерської роботи було зроблено розвідувальний аналіз обраної тематики й датасету. Сформовано теплову карту масиву даних, отримано наглядну інформацію, яка показує відношення врожайності пшениці у країнах Європи по роках. Згенеровано додаткові графіки, для розширеного розуміння вхідних даних.

Проведено аналіз регресійних моделей для машинного передбачення даних, визначено, що для отримання ефективного результату під час виконання поставленого завдання магістерської роботи необхідно використовувати такі моделі: XGBoostRegressor, DecisionTreeRegressor, RandomForestRegressor. Запрограмувавши моделі та вхідні дані з обраного датасету, було виконано навчання для отримання передбачення врожаю зернових культур країн Європи.

З яких оптимальною моделлю даних для виконання поставленої задачі є XGBoostRegressor – 95.94% за метрикою r2_score.

Графік важливості ознак показав (див. рисунок 4.22), що передбачуваний 2019 рік, більшою мірою залежить не від попереднього 2018 року (0,04), а саме через один рік назад, тобто для навчання моделі важливими були дані 2017 (0.46) та 2018 (0.53) років, які мали рівномірний вплив на точність результатів. Інший варіант графіку важливості ознак показує (див. рисунок 4.23), що на передбачення врожайності за 2014 рік вагомий вплив мали дані за 2008 рік (0.71). Тобто з отриманих результатів можна зробити висновок, що для ефективного навчання моделі передбачення даних потрібно мати великий масив інформації.

Результатом виконання роботи стала реалізація інформаційної технології, яка поєднує в собі усі розроблені операції з вхідними даними щодо урожайності сільськогосподарських культур в Європі. Показано UML-діаграму діяльності для кращого розуміння алгоритму виконання технології. Така розробка дає можливість використовувати вже готове напрацювання на вхідних даних не тільки країн Європи, а й територій Північної та Південної Америки й інших частин планети Земля для отримання передбачення інформації по урожайності.

5. ЕКОНОМІЧНА ЧАСТИНА

Виконання науково-дослідної роботи завжди передбачає отримання певних результатів і вимагає відповідних витрат. Результати виконаної роботи завжди дають нам нові знання, які в подальшому можуть бути використані для удосконалення та/або розробки (побудови) нових, більш продуктивних зразків техніки, процесів та програмного забезпечення.

Дослідження на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» може бути віднесено до фундаментальних і пошукових наукових досліджень і спрямоване на вирішення наукових проблем, пов’язаних з практичним застосуванням. Основою таких досліджень є науковий ефект, який виражається в отриманні наукових результатів, які збільшують обсяг знань про природу, техніку та суспільство, які розвивають теоретичну базу в тому чи іншому науковому напрямку, що дозволяє виявити нові закономірності, які можуть використовуватися на практиці.

Для цього випадку виконаємо такі етапи робіт:

1) здійснимо проведення наукового аудиту досліджень, тобто встановлення їх наукового рівня та значимості;

2) проведемо планування витрат на проведення наукових досліджень;

3) здійснимо розрахунок рівня важливості наукового дослідження та перспективності, визначимо ефективність наукових досліджень.

5.1 Оцінювання наукового ефекту

Основними ознаками наукового ефекту науково-дослідної роботи є новизна роботи, рівень її теоретичного опрацювання, перспективність, рівень розповсюдження результатів, можливість реалізації. Науковий ефект НДР на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» можна охарактеризувати двома показниками: ступенем наукової новизни та рівнем теоретичного опрацювання.

Значення показників ступеня новизни і рівня теоретичного опрацювання науково-дослідної роботи в балах наведені в таблиці 5.1 та 5.2.

Таблиця 5.1 – Показники ступеня новизни науково-дослідної роботи виставлені експертами

Ступінь новизни	Характеристика ступеня новизни	Значення ступеня новизни, бали		
		Експерти (ПІБ, посада)		
		1	2	3
Принципово нова	Робота якісно нова за постановкою задачі і ґрунтується на застосуванні оригінальних методів дослідження. Результати дослідження відкривають новий напрям в даній галузі науки і техніки. Отримані принципово нові факти, закономірності; розроблена нова теорія. Створено принципово новий пристрій, спосіб, метод	0	0	0
Нова	Отримана нова інформація, яка суттєво зменшує невизначеність наявних значень (по-новому або вперше пояснені відомі факти, закономірності, впроваджені нові поняття, розкрита структура змісту). Проведено суттєве вдосконалення, доповнення і уточнення раніше досягнутих результатів	57	53	56
Відносно нова	Робота має елементи новизни в постановці задачі і методах дослідження. Результати дослідження систематизують і узагальнюють наявну інформацію, визначають шляхи подальших досліджень; вперше знайдено зв’язок (або знайдено новий зв’язок) між явищами. В принципі відомі положення розповсюджені на велику кількість об’єктів, в результаті чого знайдено ефективне рішення. Розроблені більш прості способи для досягнення відомих результатів. Проведена часткова раціональна модифікація (з ознаками новизни)	0	0	0
Традиційна	Робота виконана за традиційною методикою. Результати дослідження мають інформаційний характер. Підтверджені або поставлені під сумнів відомі факти та твердження, які потребують перевірки. Знайдено новий варіант рішення, який не дає суттєвих переваг в порівнянні з існуючим	0	0	0
Не нова	Отримано результат, який раніше зафіксований в інформаційному полі, та не був відомий авторам	0	0	0
Середнє значення балів експертів		55,3		

Згідно отриманого середнього значення балів експертів ступінь новизни характеризується як нова, тобто отримана нова інформація, яка суттєво зменшує невизначеність наявних знань (по-новому або вперше пояснені відомі факти, закономірності, впроваджені нові поняття, розкрита структура змісту) та проведено суттєве вдосконалення, доповнення і уточнення раніше досягнутих результатів.

Таблиця 5.2 – Показники рівня теоретичного опрацювання науково-дослідної роботи виставлені експертами

Характеристика рівня теоретичного опрацювання	Значення показника рівня теоретичного опрацювання, бали		
	Експерт (ПІБ, посада)		
	1	2	3
Відкриття закону, розробка теорії	0	0	0
Глибоке опрацювання проблеми: багатоаспектний аналіз зв'язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу	75	73	77
Розробка способу (алгоритму, програми), пристрою, отримання нової речовини	0	0	0
Елементарний аналіз зв'язків між фактами та наявною гіпотезою, класифікація, практичні рекомендації для окремого випадку тощо	0	0	0
Опис окремих елементарних фактів, викладення досвіду, результатів спостережень, вимірювань тощо	0	0	0
Середнє значення балів експертів	75,0		

Згідно отриманого середнього значення балів експертів рівень теоретичного опрацювання науково-дослідної роботи характеризується як глибоке опрацювання проблеми: багатоаспектний аналіз зв'язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу.

Показник, який характеризує рівень наукового ефекту, визначаємо за формулою [47]:

$$E_{нау} = 0,6 \cdot k_{нов} + 0,4 \cdot k_{теор}, \quad (5.1)$$

де $k_{нов}, k_{теор}$ - показники ступеня новизни та рівня теоретичного опрацювання науково-дослідної роботи, $k_{нов} = 55,3, k_{теор} = 75,0$ балів;

0,6 та 0,4 – питома вага (значимість) показників ступеня новизни та рівня теоретичного опрацювання науково-дослідної роботи.

$$E_{нау} = 0,6 \cdot k_{нов} + 0,4 \cdot k_{теор} = 0,6 \cdot 55,3 + 0,4 \cdot 75,00 = 63,20 \text{ балів.}$$

Визначення характеристики показника $E_{нау}$ проводиться на основі висновків експертів виходячи з граничних значень, які наведені в таблиці 5.3.

Таблиця 5.3 – Граничні значення показника наукового ефекту

Досягнутий рівень показника	Кількість балів
Високий	70…100
Середній	50…69
Достатній	15…49
Низький (помилкові дослідження)	1…14

Відповідно до визначеного рівня наукового ефекту проведеної науково-дослідної роботи на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі», даний рівень становить 63,20 балів і відповідає статусу - середній рівень. Тобто у даному випадку можна вести мову про потенційну фактичну ефективність науково-дослідної роботи.

5.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.2.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [47]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.2)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 19000{,}00 \cdot 21 / 21 = 19000{,}00 \; грн.$$

Проведені розрахунки зведемо до таблиці 5.4.

Таблиця 5.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник дослідної роботи	19000,00	904,76	21	19000,00
Інженер-дослідник інформаційних систем	17500,00	833,33	21	17500,00
Експерт-агроном	17900,00	852,38	3	2557,14
Інженер-розробник програмного забезпечення	19500,00	928,57	15	13928,57
Фахівець 1-ї категорії	8200,00	390,48	15	5857,14
Всього				58842,86

Основна заробітна плата робітників:

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i \,, \tag{5.3}$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{М_М \cdot K_i \cdot K_c}{T_р \cdot t_{зм}} \,, \tag{5.4}$$

де $М_М$ – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $Мм$=8000,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [47];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об’єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_р$ – середнє число робочих днів в місяці, приблизно $T_р$ = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 8000{,}00 \cdot 1{,}10 \cdot 1{,}15 \,/\, (21 \cdot 8) = 60{,}24$ *грн.*

$З_{р1} = 60{,}24 \cdot 7{,}20 = 433{,}71$ *грн.*

Таблиця 5.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка робочого місця інженера-дослідника	7,20	2	1,10	60,24	433,71
Інсталяція програмного забезпечення середовища розробки і моделювання	7,30	3	1,35	73,93	539,68
Компіляція програмних блоків моделювання інформаційної системи	5,80	4	1,50	82,14	476,43
Підготовка локального серверного обладнання для проведення досліджень	6,00	5	1,70	93,10	558,57
Налагодження програмних блоків збору інформації	6,00	5	1,70	93,10	558,57
Тестування системи	8,00	2	1,10	60,24	481,90
Всього					3048,87

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (5.5)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 10%.

$З_{дод}$ = (58842,86 + 3048,87) · 10 / 100% = 6189,17 грн.

5.2.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (5.6)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (58842,86 + 3048,87 + 6189,17) · 22 / 100% = 14977,80 грн.

5.2.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі».

Витрати на матеріали на даному етапі проведення досліджень в основному пов’язані з використанням моделей елементів та моделювання роботи і досліджень за допомогою комп’ютерної техніки та створення експериментальних математичних моделей або програмного забезпечення, тому дані витрати формуються на основі витратних матеріалів характерних для офісних робіт.

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}\ , \quad (5.7)$$

де H_j – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j = 1,1 \ldots 1,15$);

$В_j$ – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

$М_1 = 3,0 \cdot 220,00 \cdot 1,05 - 0 \cdot 0 = 693,00$ *грн.*

Проведені розрахунки зведемо до таблиці 5.6.

Таблиця 5.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір А4 500	220,00	3,0	0	0	693,00
Папір для записів А4 250 (75%)	146,00	5,0	0	0	766,50
Органайзер офісний	163,00	2,0	0	0	342,30
Набір офісний (канцелярське приладдя)	203,00	3,0	0	0	639,45
Картридж для принтера	986,00	1,0	0	0	1035,30
Диск оптичний CD-R	22,70	3,0	0	0	71,51
Flesh-пам'ять 64 GB	265,00	1,0	0	0	278,25
Тека для паперів	66,00	4,0	0	0	277,20
Інші матеріали	150,00	1,0	0	0	157,50
Всього					4261,01

5.2.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі», відсутні.

5.2.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (5.8)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, (K_i = *1,10…1,12);*

k – кількість найменувань устаткування.

$B_{спец}$ = *10120,00 · 1 · 1,1 = 11132,00 грн.*

Отримані результати зведемо до таблиці 5.7:

Таблиця 5.7 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Маршрутизатор VPN TP-Link ER7212PC	1	10120,00	11132,00
Комп'ютерне обладнання для вирішення проблем оптимізаці їданих EOM HP Z6 G4 WKS Tower / Xeon Silver 4108 (6QP06EA)	1	105890,00	116479,00
Всього			127611,00

5.2.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (5.9)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань програмних засобів.

$B_{прг} = 2450,00 \cdot 1 \cdot 1,1 = 2695,00$ грн.

Отримані результати зведемо до таблиці 5.8:

Таблиця 5.8 – Витрати на придбання програмних засобів

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Платформа Kaggle	1	2450,00	2695,00
Pycharm	1	6820,00	7502,00
Бібліотеки мови програмування Python	1	1350,00	1485,00
Доступ до мережі Internet (високошвидкісний) грн/місяць	2	350,00	770,00
Моделі машинного навчання XGBoostRegressor, DecisionTreeRegressor, RandomForestRegressor	1	760,00	836,00
Всього			13288,00

5.2.7 Амортизація обладнання, програмних засобів та приміщень.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (5.10)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл} = (32800,00 \cdot 1) / (3 \cdot 12) = 911,11$ *грн.*

Проведені розрахунки зведемо до таблиці 5.9.

Таблиця 5.9 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника ПЗ	32800,0	3	1	911,11
Робоче місце інженера-програміста	7840,0	5	1	130,67
Пристрої передачі даних	6700,0	5	1	111,67
Оргтехніка	9500,0	5	1	158,33
Приміщення лабораторії розробки ПЗ	370000,0	45	1	685,19
ОС Windows	8320,0	3	1	231,11
Прикладний пакет Microsoft Office	7460,0	3	1	207,22
Всього				2435,30

5.2.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (5.11)$$

де W_{yi} – потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e$ = 10,98 грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$B_e = 0,42 \cdot 160,0 \cdot 10,98 \cdot 0,95 / 0,97 = 737,86$ *грн.*

Проведені розрахунки зведемо до таблиці 5.10.

Таблиця 5.10 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника ПЗ	0,42	160,0	737,86
Комп'ютерне обладнання для вирішення проблем оптимізації процесів EOM HP Z6 G4 WKS Tower / Xeon Silver 4108 (6QP06EA)	0,36	160,0	632,45
Робоче місце інженера-програміста	0,12	90,0	118,58
Пристрої передачі даних	0,10	90,0	98,82
Оргтехніка	0,56	3,0	18,45
Всього			1606,15

5.2.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \quad (5.12)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св} = 0\%$.

$B_{св} = (58842,86 + 3048,87) \cdot 0 / 100\% = 0,00$ грн.

5.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{Н_{сп}}{100\%}, \quad (5.13)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 0%.

$В_{сп}$ = (58842,86 + 3048,87) · 0 / 100% = 0,00 грн.

5.2.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (5.14)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$І_в$ = (58842,86 + 3048,87) · 50 / 100% = 30945,86 грн.

5.2.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов’язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (5.15)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$В_{нзв}$ = (58842,86 + 3048,87) · 100 / 100% = 61891,73 грн.
Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$В_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}, \quad (5.16)$$

$В_{заг}$= 58842,86 +3048,87 +6189,17 +14977,80 +4261,01 +0,00 +127611,00 + 13288,00 + 2435,30 +1606,15 +0,00 +0,00 +30945,86 +61891,73 = 325097,74 грн.

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{В_{заг}}{\eta}, \quad (5.17)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,9.

ЗВ = 325097,74 / 0,9 = 361219,71 грн.

5.3 Оцінювання наукової значимості науково-дослідної роботи

Оцінювання та доведення ефективності виконання науково-дослідної роботи фундаментального чи пошукового характеру є достатньо складним процесом і часто базується на експертних оцінках, тому має вірогідний характер.

Для обґрунтування доцільності виконання науково-дослідної роботи на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» використовується спеціальний комплексний показник, що враховує важливість, результативність роботи,

можливість впровадження її результатів у виробництво, величину витрат на роботу.

Комплексний показник K_P рівня науково-дослідної роботи може бути розрахований за формулою:

$$K_P = \frac{I^n \cdot T_C \cdot R}{B \cdot t}, \tag{5.18}$$

де I – коефіцієнт важливості роботи. Приймемо $I = 4$;

n – коефіцієнт використання результатів роботи; $n = 0$, коли результати роботи не будуть використовуватись; $n = 1$, коли результати роботи будуть використовуватись частково; $n = 2$, коли результати роботи будуть використовуватись в дослідно-конструкторських розробках; $n = 3$, коли результати можуть використовуватись навіть без проведення дослідно-конструкторських розробок. Приймемо n=3;

T_C – коефіцієнт складності роботи. Приймемо $T_C = 2$;

R – коефіцієнт результативності роботи; якщо результати роботи плануються вище відомих, то $R = 4$; якщо результати роботи відповідають відомому рівню, то $R = 3$; якщо нижче відомих результатів, то $R = 1$. Приймемо $R = 4$;

B – вартість науково-дослідної роботи, тис. грн. Приймемо B =361219,71 грн;

t – час проведення дослідження. Приймемо t = 0,08 років, (1 міс.).

Визначення показників *I, n, T_C, R, B, t* здійснюється експертним шляхом або на основі нормативів [47].

$$K_P = \frac{I^n \cdot T_C \cdot R}{B \cdot t} = 4^3 \cdot 2 \cdot 4 \,/\, 361{,}2 \cdot 0{,}08 = 17{,}01.$$

Якщо $K_P > 1$, то науково-дослідну роботу на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» можна вважати ефективною з високим науковим, технічним і економічним рівнем.

5.4 Висновки

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» складають 361219,71 грн. Відповідно до проведеного аналізу та розрахунків рівень наукового ефекту проведеної науково-дослідної роботи на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» є середній, а дослідження актуальними, рівень доцільності виконання науково-дослідної роботи $K_P > 1$, що свідчить про потенційну ефективність з високим науковим, технічним і економічним рівнем.

ВИСНОВКИ

Магістерська кваліфікаційна робота присвячена розробці інформаційної технології узагальнення даних вирощування сільськогосподарських культур у Європі.

У першому розділі охактеризовано сферу виробництва сільськогосподарських культур, виконано комплексний огляд об’єкту дослідження, проаналізовано важливість та актуальність проведення передбачення даних врожаю зернових культур. Проведено агрометеорологічний аналіз даних в країнах Європи. На основі отриманої інформації було підтверджено актуальність майбутнього дослідження, яке передбачає використання методів машинного навчання для передбачення даних.

У другому розділі виконано підготовку для початку роботи з програмним кодом. Використано можливості онлайн-сервісу Kaggle. Підключено датасети у створений ноутбук й проведено тестову перевірку цілісності вхідних даних. Для виконання поставленої задачі магістерської кваліфікаційної роботи було виконано підключення та попереднє налаштування обраних бібліотек мови програмування Python. Задано усі необхідні параметри, які будуть використовуватися в процесі розробки інформаційної технології.

Під час виконання третього розділу було використано можливості бібліотек Python та зроблено поєднання двох наборів даних, а також виконано налаштування фільтрації лише потрібних даних. На основі підготовленої інформації було виконано візуалізацію, яка показала, що датасет містить широкий набір актуальних даних про вирощування різноманітних сільськогосподарських культур й країн де вони збираються.

Під час виконання четвертого розділу магістерської роботи було зроблено розвідувальний аналіз обраної тематики й датасету. Сформовано теплову карту масиву даних, отримано наглядну інформацію, яка показує відношення врожайності пшениці у країнах Європи по роках. Згенеровано додаткові графіки, для розширеного розуміння вхідних даних. Проведено аналіз регресійних моделей для машинного передбачення даних, визначено, що для отримання

ефективного результату під час виконання поставленого завдання магістерської роботи необхідно використовувати такі моделі: XGBoostRegressor, DecisionTreeRegressor, RandomForestRegressor. Запрограмувавши моделі та вхідні дані з обраного датасету, було виконано навчання для отримання передбачення врожаю зернових культур країн Європи. З яких оптимальною моделлю даних для виконання поставленої задачі є XGBoostRegressor – 95.94% за метрикою r2_score. Графік важливості ознак показав, що передбачуваний 2019 рік, більшою мірою залежить не від попереднього 2018 року (0,04), а саме – через один рік назад, тобто для навчання моделі важливими були дані 2017 (0.46) та 2018 (0.53) років, які мали рівномірний вплив на точність результатів.

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі» складають 361219,71 грн. Відповідно до проведеного аналізу та розрахунків рівень наукового ефекту проведеної науково-дослідної роботи є середній, а дослідження актуальними, рівень доцільності виконання науково-дослідної роботи $K_P > 1$, що свідчить про потенційну ефективність з високим науковим, технічним і економічним рівнем.

Результатом виконання роботи стала реалізація інформаційної технології, яка поєднує в собі усі розроблені операції з вхідними даними щодо урожайності сільськогосподарських культур в Європі. Показано UML-діаграму діяльності для кращого розуміння алгоритму виконання технології. Така розробка дає можливість використовувати вже готове напрацювання на вхідних даних не тільки країн Європи, а й територій Північної та Південної Америки й інших частин планети Земля для отримання передбачення інформації по урожайності.

Апробація та публікація результатів магістерської кваліфікаційної роботи. Результати роботи подані на всеукраїнську науково-практичну інтернет-конференцію «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.) [1].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Марущак А. В., Мокін В. Б., Розроблення інформаційної технології передбачення врожаю сільськогосподарських культур. Всеукраїнська науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.). URL: https://conferences.vntu.edu.ua/index.php/. (Дата звернення: 10.11.2024).

2. Kevin HANNIGA, Why farmers struggle with harvest forecasting. 2023. URL: https://cutt.ly/UesFgAbU. (Дата звернення: 22.09.2024).

3. Robert GRAYBOSCH, Encyclopedia of Food Grains (Second Edition). 2016. URL: https://cutt.ly/EeaviuHc. (Дата звернення: 22.09.2024).

4. Jakaria SHAWON, Cereals, oilseeds, protein crops and rice. 2023. URL: https://cutt.ly/mesFgEhm. (Дата звернення: 27.09.2024).

5. Микола ПИЛИПЧУК, Пшеничні реалії: вимушений експорт до ЄС. 2022. URL: https://cutt.ly/ves1BHVc. (Дата звернення: 30.09.2024).

6. HSAT, Прогнозування культур і врожайності. 2023. URL: https://cutt.ly/wePmyppm. (Дата звернення: 02.10.2024).

7. Eurostat, Crop production in EU standard humidity. 2024. URL: https://cutt.ly/mePn7GTx. (Дата звернення: 02.10.2024).

8. Eurostat, Price indices of agricultural products, output (2015 = 100) - quarterly data. 2023. URL: https://cutt.ly/QePn5z7D. (Дата звернення: 02.10.2024).

9. Martina LIEBERMANN, The world urban population | Infographics. 2016. URL: https://cutt.ly/ReZxNBBn. (Дата звернення: 10.10.2024).

10. Luca SELLA, World Population Growth. 2018. URL: https://cutt.ly/CeZxMKrn. (Дата звернення: 11.10.2024).

11. Петро КОГУТ, Стале Сільське Господарство: Методи Та Їх Переваги. 2024. URL: https://cutt.ly/EeZx0wXl. (Дата звернення: 11.10.2024).

12. Олексій ПАВЛЕНКО,Чому IoT, AI та Machine Learning – це майбутнє сільського господарства. 2018. URL: https://cutt.ly/0eZx05cQ. (Дата звернення: 15.10.2024).

13. Remote sensing and GIS in agriculture. 2024. URL:

https://cutt.ly/heZx9uXG. (Дата звернення: 15.10.2024).

14. Jim REILLY, Landsat Normalized Difference Vegetation Index. 2020. URL: https://cutt.ly/ueZx91RW. (Дата звернення: 15.10.2024).

15. Методи визначення NDVI — названо переваги та недоліки. 2023. URL: https://cutt.ly/seZx3zvZ (Дата звернення: 17.10.2024).

16. Edvinas STONEVICIUS, Vegetation condition index. 2017. URL: https://cutt.ly/KeZx8kXB. (Дата звернення: 17.10.2024).

17. Granular Insights, What is the difference between NDVI and WDRVI? 2022. URL: https://cutt.ly/9eZx4uXq. (Дата звернення: 22.10.2024).

18. Steve LAROCQUE, Which vegetation index is better to use in Precision Agriculture? 2022. URL: https://cutt.ly/9eZx42Y0. (Дата звернення: 22.10.2024).

19. Ersin ELBASI, Nour MOSTAFA, Optimizing Agricultural Data Analysis Techniques through AI-Powered Decision-Making Processes. 2024. URL: https://cutt.ly/LeZx5thP. (Дата звернення: 23.10.2024).

20. Eurostat, Production of main cereals, EU, 2012-2022. 2022. URL: https://cutt.ly/vePn6xgL. (Дата звернення: 23.10.2024).

21. Eurostat, Temperature map. 2022. URL: https://cutt.ly/VePmqcyf. (Дата звернення: 24.10.2024).

22. Eurostat, Agricultural production – crops. 2022. URL: https://cutt.ly/jePmtQrm. (Дата звернення: 25.10.2024).

23. Leonardo VELASCO, World Sunflower Production by Country. 2021. URL: https://cutt.ly/ues0aiMh. (Дата звернення:25.10.2024).

24. Joe DION, Linear Regression. 2020. URL: https://cutt.ly/NeZceufp. (Дата звернення: (Дата звернення: 25.10.2024).

25. Wei WANG, Weiman XU, What is LSTM – Long Short-Term Memory. 2024. URL: https://cutt.ly/7eZct2Ql. (Дата звернення: 25.10.2024).

26. Ruoyu MA, Shuai DENG, Self-feedback LSTM regression model for real-time particle source apportionment. 2022. URL: https://cutt.ly/ZePn4byD. (Дата звернення: (Дата звернення: 27.10.2024).

27. What is the difference between APSIM 7.10 and APSIM Next Gen? 2024.

URL: https://cutt.ly/PePn8zLd. (Дата звернення: 27.10.2024).

28. Stephen BIGELOW, What's the best programming language for machine learning? 2024. URL: https://cutt.ly/JeZcvkae. (Дата звернення: 27.10.2024)

29. Aryan GUPTA, 20 Most Popular Python IDEs in 2024: Code Like a Pro. 2024. URL: https://cutt.ly/leZcnRL7. (Дата звернення: 27.10.2024).

30. Caglar USLU, What is Kaggle?. 2022. URL: https://cutt.ly/yeZcmmVn. (Дата звернення: 01.11.2024).

31. Наука про дані: машинне навчання та інтелектуальний аналіз даних : електронний навчальний посібник комбінованого (локального та мережевого) використання [Електронний ресурс] / В. Б. Мокін, М. В. Дратований – Вінниця : ВНТУ, 2024. – 258 с. – Режим доступу: https://docs.vntu.edu.ua/card.php?id=8163. (Дата звернення: 03.11.2024).

32. Pandas DataFrames. 2024. URL: https://cutt.ly/meZcW19o. (Дата звернення: 03.11.2024).

33. Brian MUTEA, NumPy tutorial (Everything you need to know about NumPy with examples). 2024. URL: https://cutt.ly/weZcESFp. (Дата звернення: 05.11.2024).

34. Matplotlib Tutorial – A Complete Guide to Python Plot with Examples. 2020. URL: https://cutt.ly/geZcT8db. (Дата звернення: 05.11.2024).

35. Python Seaborn Tutorial. 2023. URL: https://cutt.ly/ieZcOJ3r. (Дата звернення: 05.11.2024).

36. Python Machine Learning: Scikit-Learn Tutorial. 2023. URL: https://cutt.ly/SeZcAv6S. (Дата звернення: 05.11.2024).

37. Mohamadreza MOMENI Crop Production. 2024. URL: https://cutt.ly/5eZcDTvJ. (Дата звернення: 07.11.2024).

38. Country Mapping - ISO, Continent, Region. 2019. URL: https://cutt.ly/pePn4wt7. (Дата звернення: 07.11.2024).

39. Anzhelika DANIELKIEVYCH, Top 6 Machine Learning Techniques for Predictive Modeling and Data Analysis Explained. 2023. URL: https://cutt.ly/wesFg4h2. (Дата звернення: 11.11.2024).

40. Denys BREUS, Olga YEVTUSHENKO, Svetlana SKOK, Surveying

Geology & Mining Ecology Management (SGEM). 2020. с. 523. URL: https://cutt.ly/9esFfCkZ. (Дата звернення: 12.11.2024).

41. Muhammad SHAHANI, Sec. Geohazards and Georisks Volume 9. 2021. URL: https://cutt.ly/0esFhdiR. (Дата звернення: 12.11.2024).

42. Muhammad SHAHANI, Developing an XGBoost Regression Model for Predicting. 2023. URL: https://cutt.ly/Res0hcn7. (Дата звернення: 13.11.2024).

43. Anna KANDAS, Python | Decision Tree Regression using sklearn. 2023. URL: https://cutt.ly/sesFgyEi. (Дата звернення: 13.11.2024).

44. Carolina BRIGIDA, What is Random Forest?. 2023. URL: https://cutt.ly/Les1MfVw. (Дата звернення: 13.11.2024).

45. Ruibo ZHANG, Enhancing Signal Recognition Accuracy in Delay-Based Optical Reservoir Computing: A Comparative Analysis of Training Algorithms. 2024. URL: https://cutt.ly/zes0j5kl. (Дата звернення: 17.11.2024)

46. James COUSINS, 6 steps to build a predictive model. 2022. URL: https://cutt.ly/desFhnzJ. (Дата звернення: 17.11.2024)

47. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

Додаток А
Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«___» _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ УЗАГАЛЬНЕННЯ ДАНИХ ВИРОЩУВАННЯ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР У ЄВРОПІ»
08-34.МКР.002.02.000.ТЗ

Керівник: д.т.н., проф. каф. САІТ
_____ Олександр МОКІН
«___» _____ 2024 р.

Розробив: студент гр. 1ІСТ-23м
_____ Артем МАРУЩАК
«___» _____ 2024 р.

Вінниця 2024

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № __ по ВНТУ від «__» _____ 2024 р., та індивідуальне завдання на МКР, затверджене протоколом № __ засідання кафедри САІТ від «__» _____ 2024 р.

2. Джерела розробки:

- Наука про дані: машинне навчання та інтелектуальний аналіз даних : електронний навчальний посібник комбінованого (локального та мережевого) використання [Електронний ресурс] / В. Б. Мокін, М. В. Дратований – Вінниця : ВНТУ, 2024. – 258 с. – Режим доступу: https://docs.vntu.edu.ua/card.php?id=8163
- Robert GRAYBOSCH, Encyclopedia of Food Grains (Second Edition). 2016. URL: https://cutt.ly/EeaviuHc
- Noel Varela, Jesus Silva, Omar Bonerge Pineda, Danelys Cabrera, Prediction of the Corn Grains Yield through Artificial Intelligence. 2022. URL: https://cutt.ly/leZbIEWv

3. Мета і призначення роботи: є підвищення точності передбачення врожайності сільськогосподарських культур за рахунок використання методів машинного навчання.

4.Вихідні дані для проведення робіт:

1) Kaggle Dataset «Crop Production» https://www.kaggle.com/datasets/imtkaggleteam/crop-production

2) Kaggle Dataset «Country Mapping - ISO, Continent, Region» https://www.kaggle.com/datasets/andradaolteanu/country-mapping-iso-continent-region

5.Методи дослідження:

1. Розвідувальний аналіз;
2. Методи машинного навчання.

6. Етапи роботи і терміни їх виконання:

1. Загальна характеристика об’єкту досліджень ..... ____ – ____
2. Налаштування технологій розв’язання поставленої задачі____ – ____
3. Оброблення результатів спостережень та візуалізація даних у середовищі Kaggle ..... ____ – ____
4. Розроблення інформаційної технології узагальнення даних вирощування сільськогосподарських культур у Європі..... ____ – ____
5. Економічна частина..... ____ – ____
6. Оформлення матеріалів до захисту МКР. ..... ____ – ____

7. Очікувані результати та порядок реалізації:

Інформаційна технологія передбачення врожаю сільськогосподарських культур в країнах на території Європи, результатом якої є підвищення точності передбачення за рахунок використання методів машинного навчання.

8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання магістерських кваліфікаційних робіт для студентів спеціальності 126 «Інформаційні системи та технології» (освітня програма «Інформаційні технології аналізу даних та зображень»)

9. Порядок приймання роботи

Публічний захист..... «___» _____ 2024 р.

Початок розробки ..... «___» _____ 2024 р.

Граничні терміни виконання МКР..... «___» _____ 2024 р.

Розробив студент групи 1ІСТ-23м _____ Артем МАРУЩАК

Додаток Б

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія узагальнення даних вирощування сільськогосподарських культур у Європі»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ
Керівник: Олександр МОКІН, д.т.н., проф. каф. САІТ

Показники звіту подібності Turnitin

Оригінальність 95 % Загальна схожість 5 %

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

○ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
○ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи _____ Артем МАРУЩАК
(підпис)

Опис прийнятого рішення

Робота допускається до захисту

Особа, відповідальна за перевірку _____ Сергій ЖУКОВ
(підпис)

Додаток В
Лістинг програми

```
import pandas as pd

from arrow import now
from charset_normalizer import detect
from glob import glob

from warnings import filterwarnings
filterwarnings(action='ignore', category=FutureWarning)

CODES = '/kaggle/input/country-mapping-iso-continent-region/continents2.csv'
CROPS = '/kaggle/input/crop-production/Production_Crops_E_Europe.csv'

def get_file(filename: str):
    with open(file=filename, mode='rb') as input_fp:
        encoding = detect(input_fp.read())['encoding']
        return pd.read_csv(filepath_or_buffer=filename, encoding=encoding)

time_start = now()
df = pd.concat(objs=[get_file(filename=input_file) for input_file in
glob(CROPS)]).merge(right=pd.read_csv(filepath_or_buffer=CODES,
usecols=['alpha-3', 'name']), left_on='Area', right_on='name')
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
        df[column] = df[column].fillna(value=0)
print('data read complete in {}'.format(now() - time_start))

df.info()
# Зчитуємо всі файли і об'єднуємо їх у один DataFrame
df = pd.concat(objs=[get_file(filename=input_file) for input_file in glob(CROPS)])

# Зчитуємо файл з кодами і регіонами, залишаючи лише потрібні стовпці
codes_df = pd.read_csv(filepath_or_buffer=CODES, usecols=['alpha-3', 'name'])

# Об'єднуємо обидва DataFrames за допомогою merge, залишаючи лише рядки з
"wheat" у колонці "Item"
df = df[df['Item'].isin(['Wheat', 'Sunflower seed'])].merge(right=codes_df,
left_on='Area', right_on='name')

# Заповнюємо пропущені значення у відповідних стовпцях
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
```

```
        df[column] = df[column].fillna(value=0)

# Виводимо час виконання операцій
print('data read complete in {}'.format(now() - time_start))

# Фільтруємо дані, залишаючи лише рядки, де колонка 'Element' має значення
'Yield'
df = df[df['Element'] == 'Yield']

# Виводимо перші 5 рядків результатуючого DataFrame
df.info()
df.head()
import seaborn as sns
import matplotlib.pyplot as plt

# Підготовка даних для heatmap
heatmap_data = df[['Y1961','Y1971','Y1981','Y1991','Y2001','Y2011','Y2012',
                   'Y2013','Y2014','Y2015','Y2016','Y2017','Y2018','Y2019']]

# Побудова heatmap
plt.figure(figsize=(12, 10))  # Задаємо розміри графіка
sns.heatmap(heatmap_data.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Heatmap Europe wheat yield by years')  # Заголовок графіка
plt.show()
df.head()

import pandas as pd
import plotly.express as px
from arrow import now
from charset_normalizer import detect
from glob import glob
from warnings import filterwarnings
filterwarnings(action='ignore', category=FutureWarning)

CODES = '/kaggle/input/country-mapping-iso-continent-region/continents2.csv'
CROPS = '/kaggle/input/crop-production/Production_Crops_E_Europe.csv'

def get_file(filename: str):
    with open(file=filename, mode='rb') as input_fp:
        encoding = detect(input_fp.read())['encoding']
        return pd.read_csv(filepath_or_buffer=filename, encoding=encoding)

time_start = now()

# Зчитуємо всі файли і об'єднуємо їх у один DataFrame
```

```
df = pd.concat(objs=[get_file(filename=input_file) for input_file in glob(CROPS)])

# Зчитуємо файл з кодами і регіонами, залишаючи лише потрібні стовпці
codes_df = pd.read_csv(filepath_or_buffer=CODES, usecols=['alpha-3', 'name'])

# Об'єднуємо обидва DataFrames за допомогою merge, залишаючи лише рядки з
"wheat" у колонці "Item"
#df = df[df['Item'].isin(['Wheat'])].merge(right=codes_df, left_on='Area',
right_on='name')

# Заповнюємо пропущені значення у відповідних стовпцях
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
        df[column] = df[column].fillna(value=0)

# Фільтруємо дані, залишаючи лише рядки, де колонка 'Unit' має значення
'Yield'
df = df[df['Element'] == 'Yield']

# Підрахунок кількості культур для кожної країни
country_crops = df.groupby('Area')['Item'].nunique().reset_index()
country_crops.columns = ['Country', 'Unique Crops']

# Створюємо новий DataFrame для візуалізації
viz_data = pd.DataFrame({
    'Index': range(len(country_crops)),
    'Country - Unique Crops': [f"{row['Country']} - {row['Unique Crops']}" for _, row
in country_crops.iterrows()]
})

# Додаємо колонку з кольорами
country_crops['Color'] = ['red' if country == 'Ukraine' else 'blue' for country in
country_crops['Country']]

# Створення стовпчастого графіку за допомогою plotly
fig = px.bar(country_crops, x='Country', y='Unique Crops',
             title='Кількість різних культур, вирощених у кожній країні',
             labels={'Country': 'Країна', 'Unique Crops': 'Кількість культур'},
             height=600,
             color='Color',
             color_discrete_map={'red': 'red', 'blue': 'blue'})

# Показуємо графік
fig.show()
```

```
import pandas as pd
import plotly.express as px
from arrow import now
from charset_normalizer import detect
from glob import glob

from warnings import filterwarnings
filterwarnings(action='ignore', category=FutureWarning)

CODES = '/kaggle/input/country-mapping-iso-continent-region/continents2.csv'
CROPS = '/kaggle/input/crop-production/Production_Crops_E_Europe.csv'

def get_file(filename: str):
    with open(file=filename, mode='rb') as input_fp:
        encoding = detect(input_fp.read())['encoding']
        return pd.read_csv(filepath_or_buffer=filename, encoding=encoding)

time_start = now()

# Зчитуємо всі файли і об'єднуємо їх у один DataFrame
df = pd.concat(objs=[get_file(filename=input_file) for input_file in glob(CROPS)])

# Зчитуємо файл з кодами і регіонами, залишаючи лише потрібні стовпці
codes_df = pd.read_csv(filepath_or_buffer=CODES, usecols=['alpha-3', 'name'])

# Об'єднуємо обидва DataFrames, залишаючи лише рядки з "wheat" або
"Sunflower seed" у колонці "Item"
df = df[df['Item'].isin(['Wheat', 'Sunflower seed'])].merge(right=codes_df,
left_on='Area', right_on='name')

# Заповнюємо пропущені значення у відповідних стовпцях
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
        df[column] = df[column].fillna(value=0)

# Виводимо час виконання операцій
print('data read complete in {}'.format(now() - time_start))

# Залишаємо лише необхідні колонки
df = df[['Area Code', 'Area', 'Item Code', 'Item', 'Element Code', 'Element', 'Unit',
'Y2019']]

# Фільтруємо дані, залишаючи лише рядки, де колонка 'Unit' має значення
'Yield'
df = df[df['Element'] == 'Area harvested']
```

```
df.head()

# Фільтруємо дані, залишаючи лише рядки, де колонка 'Unit' має значення 'hg/ha'
#df = df[df['Unit'] == 'hg/ha']
# Підготовка даних для графіка
pivot_df = df.pivot(index='Area', columns='Item', values='Y2019').reset_index()

# Побудова графіка за допомогою plotly
fig = px.bar(pivot_df, x='Area', y=['Sunflower seed', 'Wheat'],
             title='Area harvested (Y2019)',
             labels={'value': 'Area harvested (ha)', 'Area': 'Country'},
             barmode='stack')

# Налаштування макета графіка
fig.update_layout(template='plotly_white', xaxis_title='Country', yaxis_title='Area harvested (ha)')
fig.show()

import pandas as pd
import plotly.express as px
from arrow import now
from charset_normalizer import detect
from glob import glob

from warnings import filterwarnings
filterwarnings(action='ignore', category=FutureWarning)

CODES = '/kaggle/input/country-mapping-iso-continent-region/continents2.csv'
CROPS = '/kaggle/input/crop-production/Production_Crops_E_Europe.csv'

def get_file(filename: str):
    with open(file=filename, mode='rb') as input_fp:
        encoding = detect(input_fp.read())['encoding']
        return pd.read_csv(filepath_or_buffer=filename, encoding=encoding)

time_start = now()

# Зчитуємо всі файли і об'єднуємо їх у один DataFrame
df = pd.concat(objs=[get_file(filename=input_file) for input_file in glob(CROPS)])

# Зчитуємо файл з кодами і регіонами, залишаючи лише потрібні стовпці
codes_df = pd.read_csv(filepath_or_buffer=CODES, usecols=['alpha-3', 'name'])
```

```
# Об'єднуємо обидва DataFrames, залишаючи лише рядки з "wheat" або
"Sunflower seed" у колонці "Item"
df = df[df['Item'].isin(['Maize'])].merge(right=codes_df, left_on='Area',
right_on='name')

# Заповнюємо пропущені значення у відповідних стовпцях
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
        df[column] = df[column].fillna(value=0)

# Виводимо час виконання операцій
print('data read complete in {}'.format(now() - time_start))

# Залишаємо лише необхідні колонки
df = df[['Area Code', 'Area', 'Item Code', 'Item', 'Element Code', 'Element', 'Unit',
'Y2019']]

# Фільтруємо дані, залишаючи лише рядки, де колонка 'Unit' має значення
'Yield'
df = df[df['Element'] == 'Area harvested']

# Визначення врожайності для України та всіх інших країн
ukraine_yield = df[df['Area'] == 'Ukraine']['Y2019'].values[0]
other_countries_yield = df[df['Area'] != 'Ukraine']['Y2019'].sum()

# Створення DataFrame для графіка
bubble_data = {
    'Area': ['Ukraine', 'Other Countries'],
    'Yield': [ukraine_yield, other_countries_yield],
    'Color': ['Ukraine', 'Other Countrie']
}
bubble_df = pd.DataFrame(bubble_data)

# Побудова бульбашкового графіка за допомогою plotly
fig = px.scatter(bubble_df, x='Area', y='Yield', size='Yield', color='Color',
text='Yield',
                 title='Comparison of Area harvested Maize: Ukraine vs Other Countries',
                 labels={'Yield': 'Yield (ha)', 'Area': 'Country'})


# Налаштування макета графіка та видалення текстових міток
fig.update_traces(marker=dict(opacity=0.6, sizemode='diameter'), textposition='top
center')
fig.update_layout(template='plotly_white', xaxis_title='Country', yaxis_title='Area
harvested (ha)')
```

```
fig.show()

import pandas as pd
import plotly.express as px
from arrow import now
from charset_normalizer import detect
from glob import glob
import plotly.graph_objects as go

from warnings import filterwarnings
filterwarnings(action='ignore', category=FutureWarning)

CODES = '/kaggle/input/country-mapping-iso-continent-region/continents2.csv'
CROPS = '/kaggle/input/crop-production/Production_Crops_E_Europe.csv'

def get_file(filename: str):
    with open(file=filename, mode='rb') as input_fp:
        encoding = detect(input_fp.read())['encoding']
        return pd.read_csv(filepath_or_buffer=filename, encoding=encoding)

time_start = now()

# Зчитуємо всі файли і об'єднуємо їх у один DataFrame
df = pd.concat(objs=[get_file(filename=input_file) for input_file in glob(CROPS)])

# Зчитуємо файл з кодами і регіонами, залишаючи лише потрібні стовпці
codes_df = pd.read_csv(filepath_or_buffer=CODES, usecols=['alpha-3', 'name'])

# Об'єднуємо обидва DataFrames, залишаючи лише рядки з "wheat" або
"Sunflower seed" у колонці "Item"
df = df[df['Item'].isin(['Rapeseed'])].merge(right=codes_df, left_on='Area',
right_on='name')

# Заповнюємо пропущені значення у відповідних стовпцях
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
        df[column] = df[column].fillna(value=0)

# Виводимо час виконання операцій
print('data read complete in {}'.format(now() - time_start))

# Залишаємо лише необхідні колонки
#df = df[['Area Code', 'Area', 'Item Code', 'Item', 'Element Code', 'Element', 'Unit',
'Y2019']]
```

```
# Фільтруємо дані, залишаючи лише рядки, де колонка 'Unit' має значення 'Yield'
df = df[df['Element'] == "Yield"]

# Визначення країни з найвищим урожаєм за роками
max_yield_country_2000 = df.loc[df['Y2000'].idxmax(), 'Area']
max_yield_country_2001 = df.loc[df['Y2001'].idxmax(), 'Area']
max_yield_country_2002 = df.loc[df['Y2002'].idxmax(), 'Area']
max_yield_country_2003 = df.loc[df['Y2003'].idxmax(), 'Area']
max_yield_country_2004 = df.loc[df['Y2004'].idxmax(), 'Area']
max_yield_country_2005 = df.loc[df['Y2005'].idxmax(), 'Area']
max_yield_country_2006 = df.loc[df['Y2006'].idxmax(), 'Area']
max_yield_country_2007 = df.loc[df['Y2007'].idxmax(), 'Area']
max_yield_country_2008 = df.loc[df['Y2008'].idxmax(), 'Area']
max_yield_country_2009 = df.loc[df['Y2009'].idxmax(), 'Area']
max_yield_country_2010 = df.loc[df['Y2010'].idxmax(), 'Area']
max_yield_country_2011 = df.loc[df['Y2011'].idxmax(), 'Area']
max_yield_country_2012 = df.loc[df['Y2012'].idxmax(), 'Area']
max_yield_country_2013 = df.loc[df['Y2013'].idxmax(), 'Area']
max_yield_country_2014 = df.loc[df['Y2014'].idxmax(), 'Area']
max_yield_country_2015 = df.loc[df['Y2015'].idxmax(), 'Area']
max_yield_country_2016 = df.loc[df['Y2016'].idxmax(), 'Area']
max_yield_country_2017 = df.loc[df['Y2017'].idxmax(), 'Area']
max_yield_country_2017 = df.loc[df['Y2017'].idxmax(), 'Area']
max_yield_country_2018 = df.loc[df['Y2018'].idxmax(), 'Area']
max_yield_country_2019 = df.loc[df['Y2019'].idxmax(), 'Area']

# Визначення максимального урожаю за роками
max_yield_2000 = df['Y2000'].max()
max_yield_2001 = df['Y2001'].max()
max_yield_2002 = df['Y2002'].max()
max_yield_2003 = df['Y2003'].max()
max_yield_2004 = df['Y2004'].max()
max_yield_2005 = df['Y2005'].max()
max_yield_2006 = df['Y2006'].max()
max_yield_2007 = df['Y2007'].max()
max_yield_2008 = df['Y2008'].max()
max_yield_2009 = df['Y2009'].max()
max_yield_2010 = df['Y2010'].max()
max_yield_2011 = df['Y2011'].max()
max_yield_2012 = df['Y2012'].max()
max_yield_2013 = df['Y2013'].max()
max_yield_2014 = df['Y2014'].max()
max_yield_2015 = df['Y2015'].max()
max_yield_2016 = df['Y2016'].max()
```

```
max_yield_2017 = df['Y2017'].max()
max_yield_2018 = df['Y2018'].max()
max_yield_2019 = df['Y2019'].max()

# Побудова графіку
fig = go.Figure()

fig.add_trace(go.Bar(x=['2000','2001','2002','2003','2004','2005',
                        '2006','2007','2008','2009','2010','2011','2012',
                        '2013','2014','2015','2016','2017', '2018', '2019'], y=[max_yield_2000,
                                                                                max_yield_2001,
                                                                                max_yield_2002,
                                                                                max_yield_2003,
                                                                                max_yield_2004,
                                                                                max_yield_2005,
                                                                                max_yield_2006,
                                                                                max_yield_2007,
                                                                                max_yield_2008,
                                                                                max_yield_2009,
                                                                                max_yield_2010,
                                                                                max_yield_2011,
                                                                                max_yield_2012,
                                                                                max_yield_2013,
                                                                                max_yield_2014,
                                                                                max_yield_2015,
                                                                                max_yield_2016,
                                                                                max_yield_2017,
                                                                                max_yield_2018,
                                                                                max_yield_2019],

                    text=[max_yield_country_2000,
                          max_yield_country_2001,
                          max_yield_country_2002,
                          max_yield_country_2003,
                          max_yield_country_2004,
                          max_yield_country_2005,
                          max_yield_country_2006,
                          max_yield_country_2007,
                          max_yield_country_2008,
                          max_yield_country_2009,
                          max_yield_country_2010,
                          max_yield_country_2011,
                          max_yield_country_2012,
                          max_yield_country_2013,
                          max_yield_country_2014,
```

```
                max_yield_country_2015,
                max_yield_country_2016,
                max_yield_country_2017,
                max_yield_country_2018,
                max_yield_country_2019],
            textposition='outside',
            textangle=90,
            textfont=dict(size=200),

# Побудова кривих навчання
train_sizes, train_scores, test_scores = learning_curve(
    model, X, y, cv=5, scoring='r2', n_jobs=-1, train_sizes=np.linspace(0.1, 1.0, 10))

# Обчислення середнього та стандартного відхилення
train_scores_mean = np.mean(train_scores, axis=1)
train_scores_std = np.std(train_scores, axis=1)
test_scores_mean = np.mean(test_scores, axis=1)
test_scores_std = np.std(test_scores, axis=1)

# Побудова графіка кривих навчання
plt.figure()
plt.title("Learning Curves (XGBoost)")
plt.xlabel("Training examples")
plt.ylabel("Score")
plt.ylim(0, 1.1)
plt.grid()

# Побудова кривих з обмеженнями на стандартне відхилення
plt.fill_between(train_sizes, train_scores_mean - train_scores_std,
            train_scores_mean + train_scores_std, alpha=0.1, color="r")
plt.fill_between(train_sizes, test_scores_mean - test_scores_std,
            test_scores_mean + test_scores_std, alpha=0.1, color="g")
plt.plot(train_sizes, train_scores_mean, 'o-', color="r", label="Training score")
plt.plot(train_sizes, test_scores_mean, 'o-', color="g", label="Cross-validation score")

plt.legend(loc="best")
plt.show()

import pandas as pd
import numpy as np
from arrow import now
from charset_normalizer import detect
from glob import glob
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from warnings import filterwarnings
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import GridSearchCV, KFold
from sklearn.metrics import r2_score

filterwarnings(action='ignore', category=FutureWarning)

CODES = '/kaggle/input/country-mapping-iso-continent-region/continents2.csv'
CROPS = '/kaggle/input/crop-production/Production_Crops_E_Europe.csv'

def get_file(filename: str):
    with open(file=filename, mode='rb') as input_fp:
        encoding = detect(input_fp.read())['encoding']
        return pd.read_csv(filepath_or_buffer=filename, encoding=encoding)
time_start = now()

# Зчитуємо файл з кодами і регіонами, залишаючи лише потрібні стовпці
codes_df = pd.read_csv(filepath_or_buffer=CODES, usecols=['alpha-3', 'name'])

# Зчитуємо всі файли і об'єднуємо їх у один DataFrame
df = pd.concat(objs=[get_file(filename=input_file) for input_file in glob(CROPS)])

# Об'єднуємо обидва DataFrames за допомогою merge, залишаючи лише рядки з
"wheat" у колонці "Item"
df = df[df['Item'].isin(['Wheat'])].merge(right=codes_df, left_on='Area',
right_on='name')

# Заповнюємо пропущені значення у відповідних стовпцях
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
        df[column] = df[column].fillna(value=0)

# Видаляємо колонки, що закінчуються на 'F'
df = df.drop(columns=[col for col in df.columns if col.endswith('F')])

# Фільтруємо дані, залишаючи лише рядки, де колонка 'Unit' має значення
'Yield'
df = df[df['Element'] == 'Yield']

#train = df[['Area Code', 'Item Code', 'Y2017', 'Y2018', 'Y2019']].copy()
X = df[['Area Code', 'Item Code', 'Y2017', 'Y2018']].copy()
y = df[['Y2019']].copy()
```

```
# Розподіл даних на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
random_state=0)

# Ініціалізація моделі RandomForestRegressor
model = RandomForestRegressor(n_estimators=100, random_state=0)

# Навчання моделі на навчальних даних
model.fit(X_train, y_train)

# Передбачення на навчальних даних
y_train_pred = model.predict(X_train)

# Передбачення на тестових даних
y_test_pred = model.predict(X_test)

# Оцінка точності моделі за допомогою середньоквадратичної помилки (MSE)
для навчальних даних
train_mse = mean_squared_error(y_train, y_train_pred)
# Оцінка точності моделі за допомогою середньоквадратичної помилки (MSE)
для тестових даних
test_mse = mean_squared_error(y_test, y_test_pred)
# Оцінка точності моделі за допомогою коефіцієнта детермінації (R²) для
навчальних даних
train_r2 = r2_score(y_train, y_train_pred)
# Оцінка точності моделі за допомогою коефіцієнта детермінації (R²) для
тестових даних
test_r2 = r2_score(y_test, y_test_pred)

# Переведення R² в точність у відсотках
train_accuracy = train_r2 * 100
test_accuracy = test_r2 * 100

print(f"Train Accuracy: {train_accuracy:.2f}%")
print(f"Test Accuracy: {test_accuracy:.2f}%")


# Візуалізація результатів
plt.figure(figsize=(14, 8))

# Графік для навчальних даних
plt.subplot(1, 2, 1)
plt.scatter(y_train, y_train_pred, alpha=0.7, color='blue')
```

```
plt.plot([y_train.min(), y_train.max()], [y_train.min(), y_train.max()], '--r')
plt.xlabel('Фактична врожайність')
plt.ylabel('Передбачена врожайність')
plt.title(f'Навчальна вибірка\nR²: {train_r2:.2f}, Точність: {train_accuracy:.2f}%')

# Графік для тестових даних
plt.subplot(1, 2, 2)
plt.scatter(y_test, y_test_pred, alpha=0.7, color='green')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], '--r')
plt.xlabel('Фактична врожайність')
plt.ylabel('Передбачена врожайність')
plt.title(f'Тестова вибірка\nR²: {test_r2:.2f}, Точність: {test_accuracy:.2f}%')

plt.tight_layout()
plt.show()

import pandas as pd
import numpy as np
from arrow import now
from charset_normalizer import detect
from glob import glob
import seaborn as sns
import matplotlib.pyplot as plt
from warnings import filterwarnings
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import GridSearchCV, KFold
from sklearn.metrics import r2_score

filterwarnings(action='ignore', category=FutureWarning)

CODES = '/kaggle/input/country-mapping-iso-continent-region/continents2.csv'
CROPS = '/kaggle/input/crop-production/Production_Crops_E_Europe.csv'

def get_file(filename: str):
    with open(file=filename, mode='rb') as input_fp:
        encoding = detect(input_fp.read())['encoding']
        return pd.read_csv(filepath_or_buffer=filename, encoding=encoding)
time_start = now()

# Зчитуємо файл з кодами і регіонами, залишаючи лише потрібні стовпці
codes_df = pd.read_csv(filepath_or_buffer=CODES, usecols=['alpha-3', 'name'])
```

```
# Зчитуємо всі файли і об'єднуємо їх у один DataFrame
df = pd.concat(objs=[get_file(filename=input_file) for input_file in glob(CROPS)])

# Об'єднуємо обидва DataFrames за допомогою merge, залишаючи лише рядки з
"wheat" у колонці "Item"
#f = df[df['Item'].isin(['Wheat'])].merge(right=codes_df, left_on='Area',
right_on='name')

# Заповнюємо пропущені значення у відповідних стовпцях
for column in df.columns:
    if column.startswith('Y') and not column.endswith('F'):
        df[column] = df[column].fillna(value=0)

# Видаляємо колонки, що закінчуються на 'F'
df = df.drop(columns=[col for col in df.columns if col.endswith('F')])

# Фільтруємо дані, залишаючи лише рядки, де колонка 'Unit' має значення
'Yield'
#df = df[df['Element'] == 'Yield']

#train = df[['Area Code', 'Item Code', 'Y2017', 'Y2018', 'Y2019']].copy()
#X = df[['Y2010', 'Y2011', 'Y2012', 'Y2013', 'Y2014', 'Y2015', 'Y2016', 'Y2017',
'Y2018']].copy()
X = df[['Y2016', 'Y2017', 'Y2018']].copy()
y = df[['Y2019']].copy()

# Розділення даних на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
random_state=42)

# Створення моделі DecisionTreeRegressor
model = DecisionTreeRegressor(random_state=42)

# Навчання моделі на навчальній вибірці
model.fit(X_train, y_train)

# Передбачення на навчальній та тестовій вибірках
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Оцінка моделі за допомогою метрики середньоквадратичної помилки (MSE)
та R²
train_mse = mean_squared_error(y_train, y_train_pred)
train_r2 = r2_score(y_train, y_train_pred)
```

```
test_mse = mean_squared_error(y_test, y_test_pred)
test_r2 = r2_score(y_test, y_test_pred)

print(f"Test Mean Squared Error: {test_mse}")
print(f"Test R²: {test_r2:.2f} ({test_r2 * 100:.2f}%)")


# Візуалізація важливості ознак
feature_importances = pd.Series(model.feature_importances_, index=X.columns)
feature_importances = feature_importances.sort_values(ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(x=feature_importances, y=feature_importances.index)
plt.xlabel('Feature Importance')
plt.ylabel('Features')
plt.title('Feature Importances in DecisionTreeRegressor')
plt.grid(True)
plt.show()
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ УЗАГАЛЬНЕННЯ ДАНИХ ВИРОЩУВАННЯ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР У ЄВРОПІ

Нормоконтроль: к.т.н., доцент
_____ Сергій ЖУКОВ
«___» _____ 2024 р.

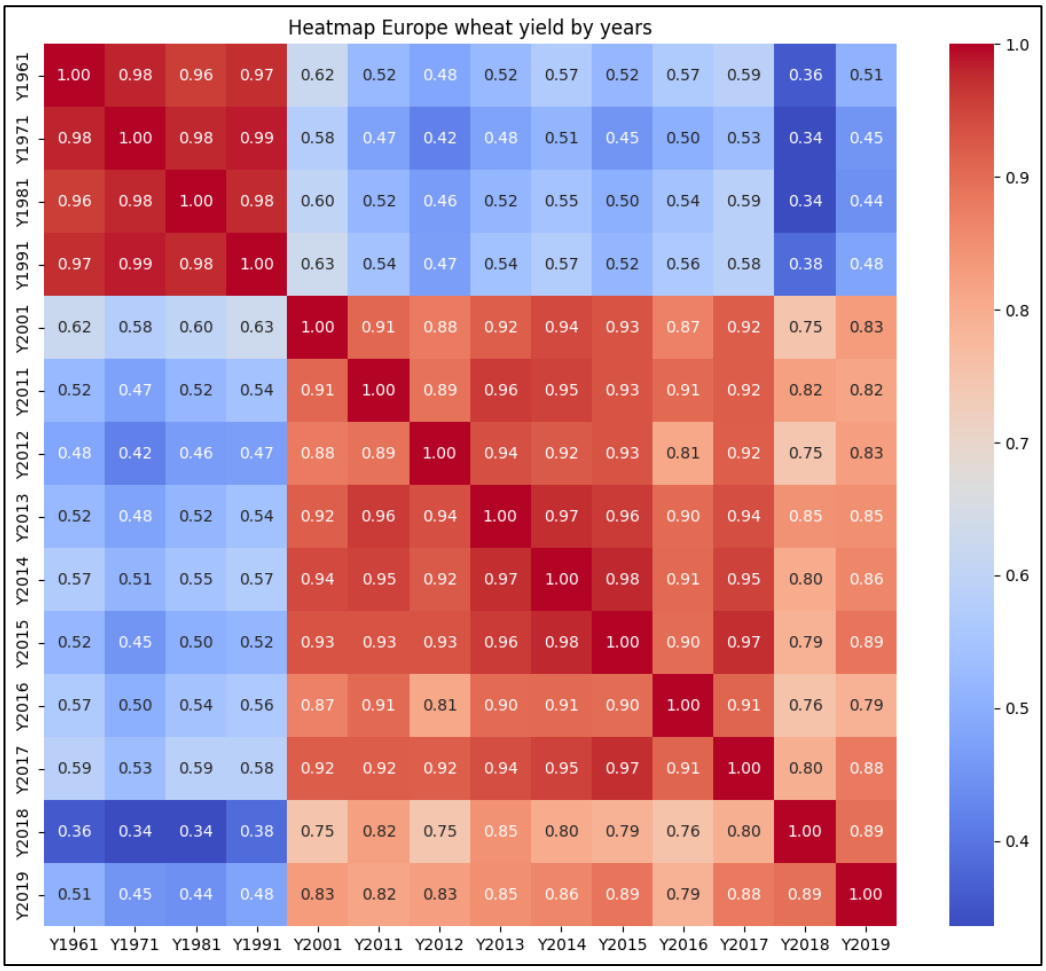


Рисунок Г.1 – Heatmap врожайності пшениці у Європі

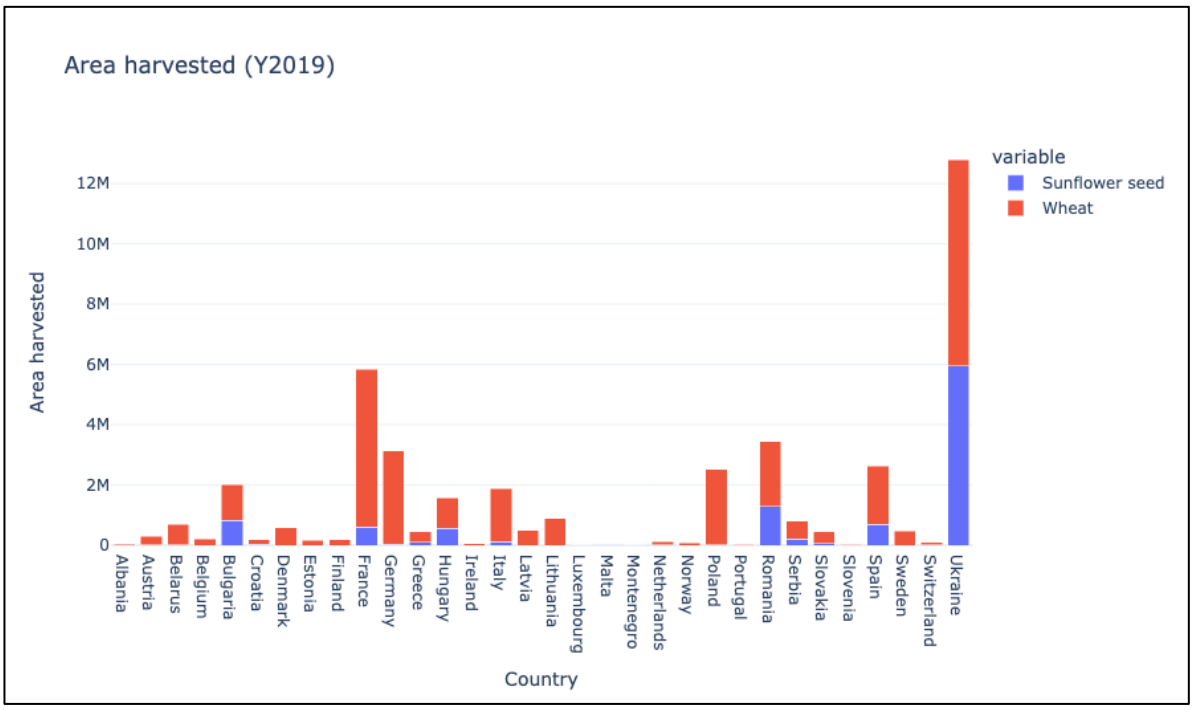


Рисунок Г.2 – Посівні площі соняшника та пшениці за 2019 рік

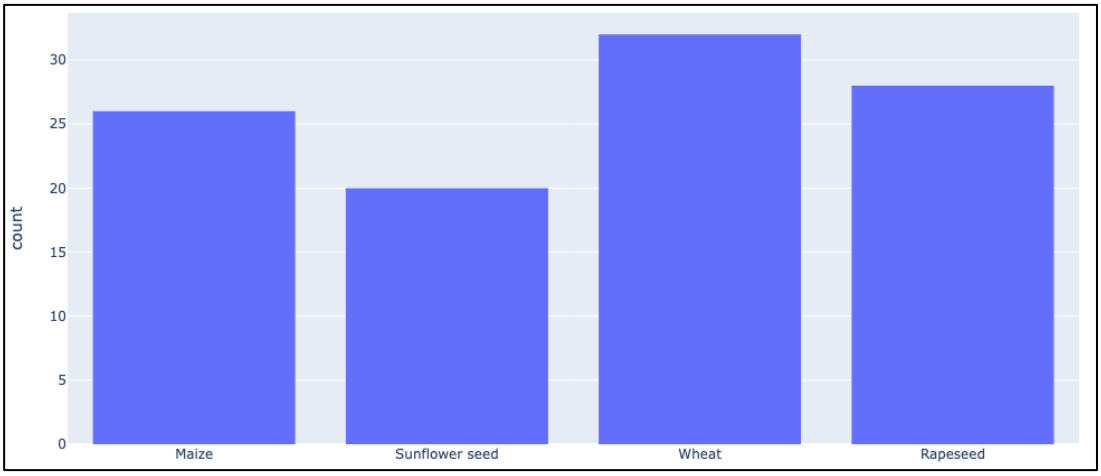


Рисунок Г.3 – Рейтинг поширення зернових культур в Європі

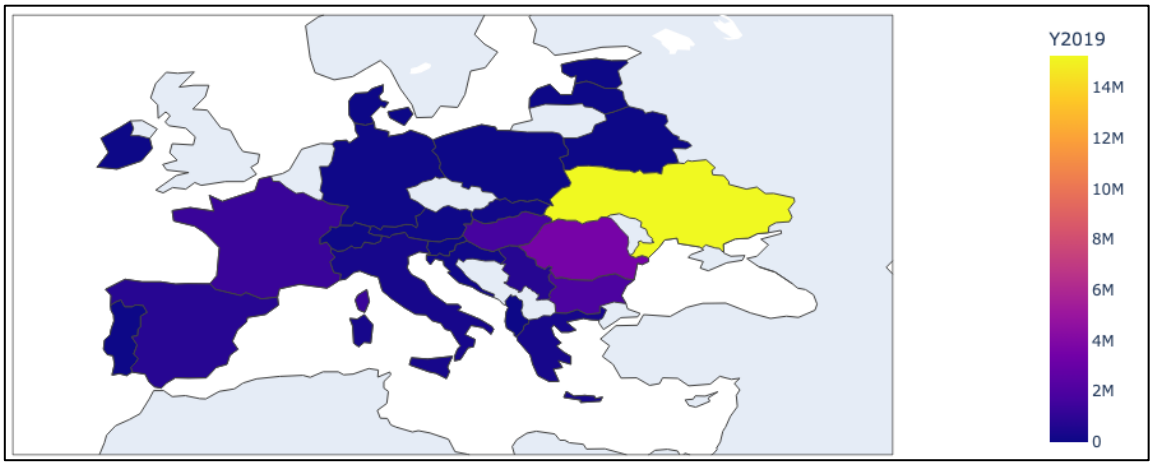


Рисунок Г.4 – Валовий збір соняшника за 2019 рік

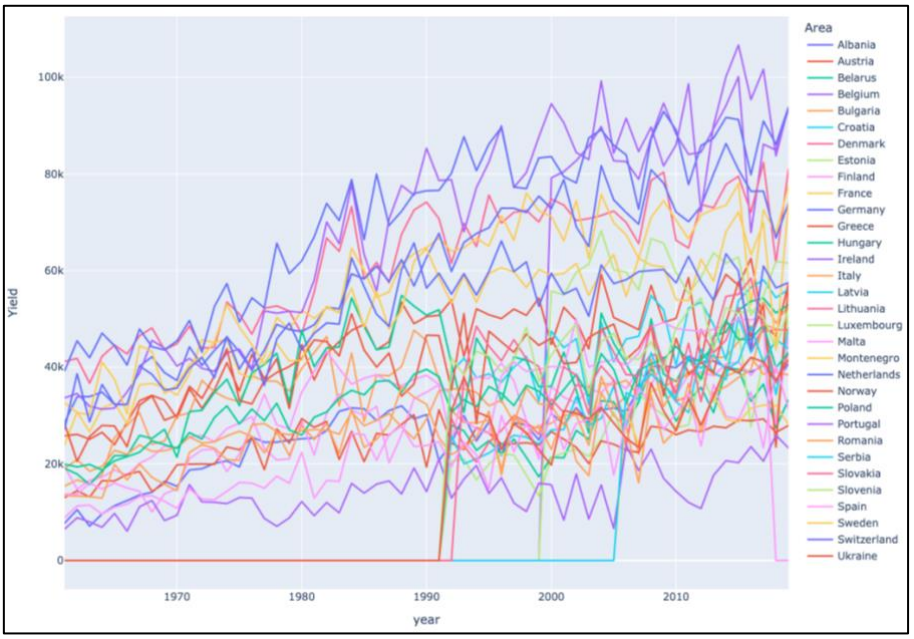


Рисунок Г.5 – Загальна середня врожайність пшениці країн Європи

```python
# Вибір ознак (features) та цільової змінної (target)
features = ['Area Code','Item Code', 'Y2017', 'Y2018']
target = 'Y2019'
# Розділення даних на ознаки та цільову змінну
train_all = df[features]
target_all = df[target]
# Розділення на навчальні та тестові дані (80% на навчання, 20% на тестування)
train, valid, target_train, target_valid = train_test_split(train_all, target_all, test_size=0.2, random_state=42)
# Виведення результатів
print("Навчальні дані (X_train):")
print(train.info())
#print("\nНавчальні мітки (target_train):")
#print(target_train.info())
print("\nТестові дані (valid):")
print(valid.info())
#print("\nТестові мітки (target_valid):")
#print(target_valid.info())
```

```
Навчальні дані (X_train):
<class 'pandas.core.frame.DataFrame'>
Index: 8445 entries, 6275 to 7270
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Area Code  8445 non-null   int64
 1   Item Code  8445 non-null   int64
 2   Y2017      6136 non-null   float64
 3   Y2018      6386 non-null   float64
dtypes: float64(2), int64(2)
memory usage: 329.9 KB
None

Тестові дані (valid):
<class 'pandas.core.frame.DataFrame'>
Index: 2112 entries, 1950 to 1613
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Area Code  2112 non-null   int64
 1   Item Code  2112 non-null   int64
 2   Y2017      1490 non-null   float64
 3   Y2018      1577 non-null   float64
dtypes: float64(2), int64(2)
memory usage: 82.5 KB
None
```

Рисунок Г.6 – Формування навчального та валідаційного набору даних

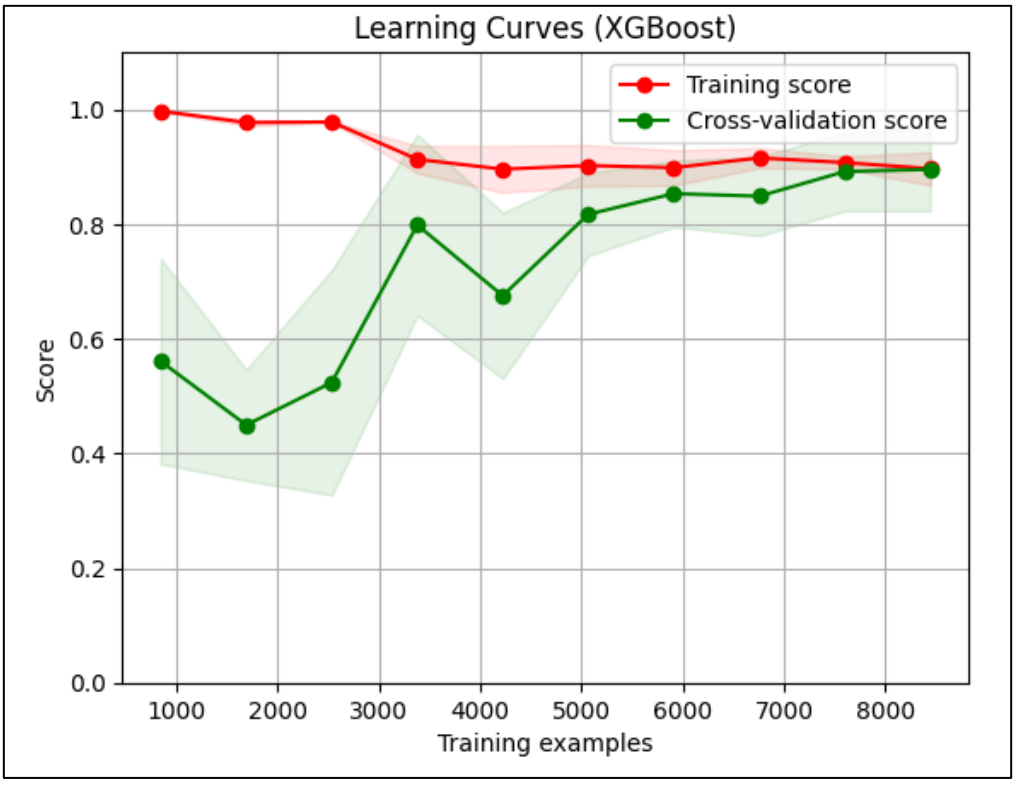


Рисунок Г.7 – Криві навчання для моделі XGBoostRegressor

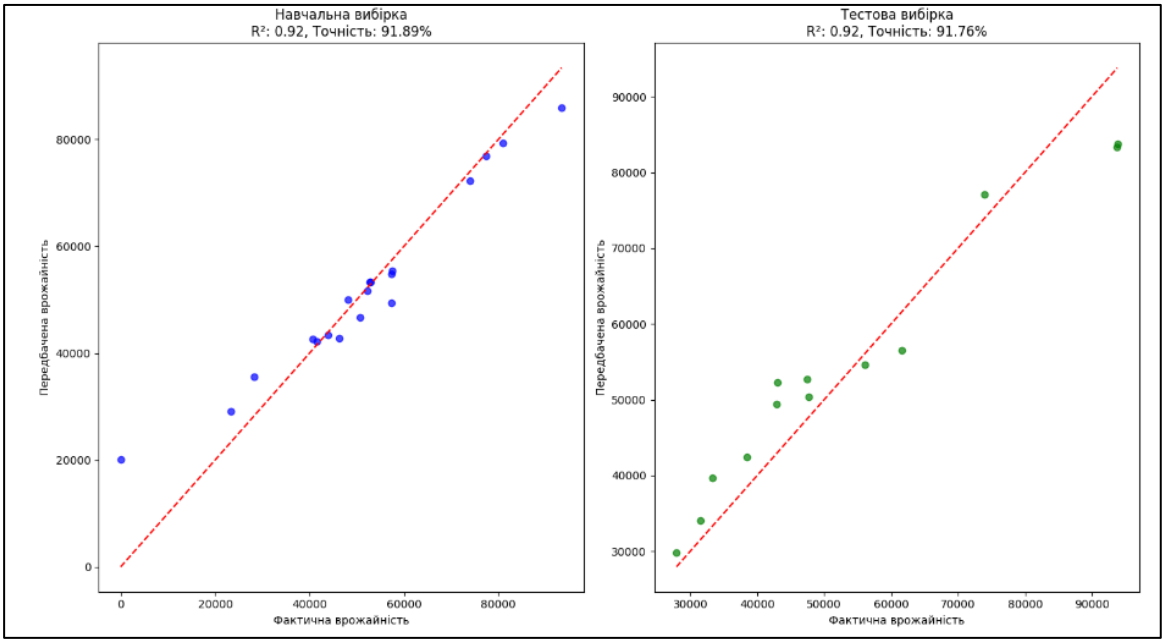


Рисунок Г.8 – Передбачення для валідаційних даних

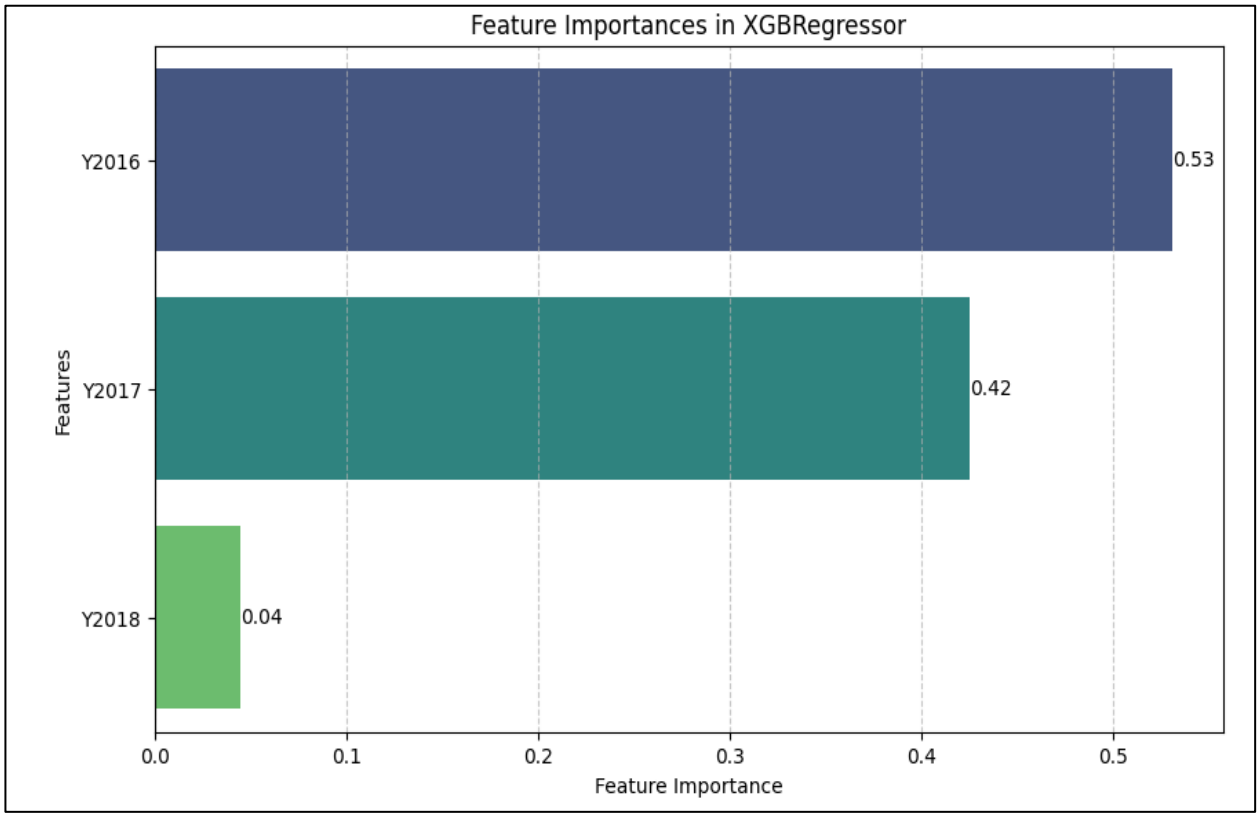


Рисунок Г.9 – Графік важливості ознак XGBRegressor

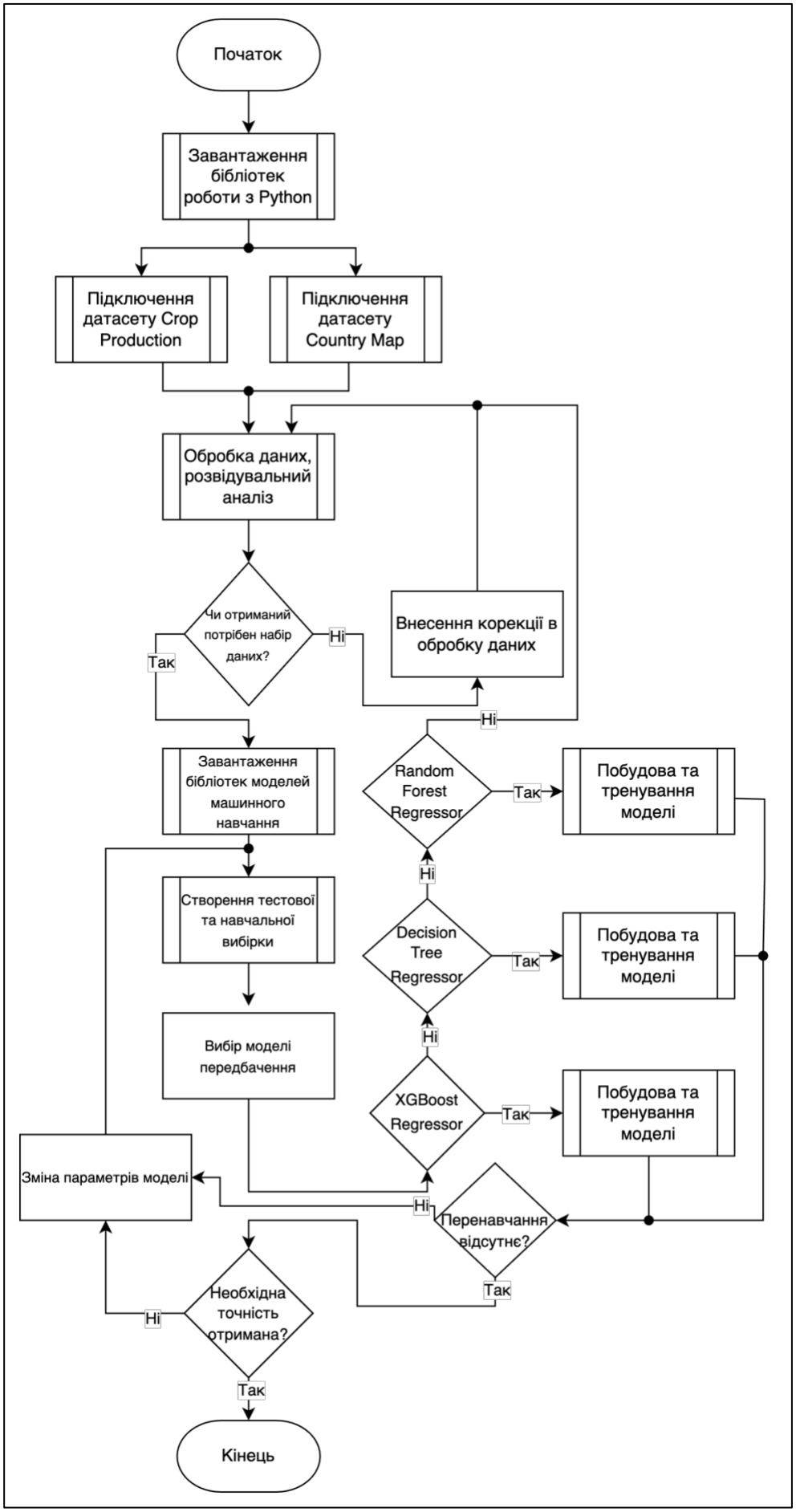


Рисунок Г.10 – Блок-схема алгоритму роботи інформаційної технології