

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАОЧНОЇ ТА ПІСЛЯДИПЛОМНОЇ
ОСВІТИ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему:

**«ОБҐРУНТУВАННЯ МОДЕЛІ МАШИННОГО
НАВЧАННЯ ДЛЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ
СИСТЕМИ ПЛАНУВАННЯ ПРОЦЕСІВ У СІЛЬСЬКОМУ
ГОСПОДАРСТВІ»**

Виконала: студентка групи Іт-71з

спеціальності 126 «Інформаційні системи

та технології» _____ Железняк А.М.

(прізвище та ініціали)

Керівник:

Тригуба А.М.

(прізвище та ініціали)

ДУБЛЯНИ 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАОЧНОЇ ТА ПІСЛЯДИПЛОМНОЇ
ОСВІТИ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Рівень вищої освіти другий (магістерський)
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

д.т.н., професор, Тригуба А. М.

(вч. звання, прізвище, ініціали)

“ ” _____ 202__ року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Желізняк Алли Михайлівни

(прізвище, ім'я, по батькові)

1. Тема роботи «Обґрунтування моделі машинного навчання для інтелектуальної інформаційної системи планування процесів у сільському господарстві»

керівник роботи д.т.н., професор.Тригуба А.М.

(наук.ступінь, вч. звання, прізвище, ініціали)

затверджені наказом Львівського НУП № 33 /к - с від 17.02.2023 р

2. Строк подання студентом роботи 13.02.2024 р.

3. Вихідні дані: вихідні дані та вимоги до інформаційної інтелектуальної системи, опис інструментів та вихідних даних для реалізації моделі машинного навчання, архітектура інформаційної системи планування процесів у сільському господарстві на основі штучного інтелекту; науково-технічна і довідкова література.

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)
Вступ

1. Аналіз стану питання в теорії та практиці та постановка завдання

2. Обґрунтування, вибір та реалізація інструментарію вирішення задачі

3. Результати вирішення задачі

4. Охорона праці та безпека в надзвичайних ситуаціях

5. Визначення ефективності інтелектуальної інформаційної системи

Висновки

Список використаних джерел

5. Перелік графічного матеріалу

Графічний матеріал подається у вигляді презентації

6. Консультанти розділів

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3, 5	<i>Тригуба А.М., завідувач кафедри інформаційних технологій</i>			
4	<i>Городецький І.М., доцент кафедри фізики, інженерної механіки та безпеки виробництва</i>			

7. Дата видачі завдання 01.03.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Відмітка про виконання
1	<i>Отримання завдання. Вивчення рекомендованої літератури по темі роботи. Написання першого розділу</i>	<i>01.03.2023 – 30.04.2023</i>	
2	<i>Проектування та опис технічного завдання, обґрунтування та вибір інструментарію реалізації проекту (написання другого розділу).</i>	<i>01.05.2023 – 30.06.2023</i>	
3	<i>Програмна реалізація поставленого завдання (написання третього розділу)</i>	<i>01.07.2023 – 30.09.2023</i>	
4	<i>Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»</i>	<i>01.10.2023 – 31.10.2023</i>	
5	<i>Оцінка ефективності поставленого завдання (виконання п'ятого розділу)</i>	<i>01.11.2023 – 31.12.2023</i>	
6	<i>Завершення оформлення основної частини, написання висновків та підготовка презентаційного матеріалу</i>	<i>01.01.2024 – 28.01.2024</i>	
7	<i>Завершення роботи в цілому. Підготовка до захисту кваліфікаційної роботи</i>	<i>29.01.2024 – 13.02.2024</i>	

Студент

_____ (підпис)

Желєзняк А.М.
(прізвище та ініціали)

Керівник роботи

_____ (підпис)

Тригуба А.М.
(прізвище та ініціали)

УДК 004.85:004.056]:631.1

Обґрунтування моделі машинного навчання для інтелектуальної інформаційної системи планування процесів у сільському господарстві.

Желізняк А.М. Кафедра інформаційних технологій - Дубляни, Львівський НУП, 2024.

Кваліфікаційна робота: 78 с. текст. част., 41 рис., 15 табл., 63 джерела, 5 додатків

Наведено теоретичні основи використання інтелектуальних інформаційних систем для прийняття управлінських рішень в сільському господарстві. Проведено огляд прикладних інтелектуальних інформаційних систем в аграрній сфері. Обґрунтовано перспективи використання штучного інтелекту та моделей машинного навчання для вдосконалення процесу та результатів прийняття управлінських рішень в аграрній сфері.

Здійснено обґрунтування та вибір моделі машинного навчання для інтелектуальної інформаційної системи планування процесів в сільському господарстві. Проаналізовано перспективи використання цифрових асистентів на базі генеративного штучного інтелекту для прийняття управлінських рішень в аграрній сфері. Подано особливості планування та розробки архітектури інтелектуальної системи для прийняття управлінських рішень в сільському господарстві, що передбачає застосування штучного інтелекту та моделей машинного навчання. Виконано підбір параметрів та описано навчання моделей. Проаналізовано та визначено показники ефективності запропонованих заходів щодо вдосконалення процесу прийняття управлінських рішень в аграрній сфері на основі обраної моделі машинного навчання. Здійснено аналіз охорони праці та ризиків появи травматичних ситуацій в аграрній сфері та їх мінімізацію на основі впровадження інтелектуальної інформаційної системи при виконанні різних типів робіт в рослинництві та тваринництві.

Ключові слова: машинне навчання, інтелектуальна інформаційна система, планування, сільське господарство.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ СТАНУ ПИТАННЯ В ТЕОРІЇ ТА ПРАКТИЦІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	7
1.1 Теоретичні основи функціонування інтелектуальних інформаційних систем планування.....	7
2. Особливості застосування машинного навчання для інтелектуальних інформаційних систем планування.....	10
1.3 Огляд існуючих практик застосування інтелектуальних інформаційних систем та штучного інтелекту в сільському господарстві	22
РОЗДІЛ 2. ОБҐРУНТУВАННЯ, ВИБІР ТА РЕАЛІЗАЦІЯ ІНСТРУМЕНТАРІЮ ВИРІШЕННЯ ЗАДАЧІ	28
2.1 Обґрунтування методу машинного навчання інтелектуальної інформаційної системи планування.....	28
2.2 Вибір інструментарію для вирішення задачі	39
2.3. Визначення перспектив застосування моделей генеративного штучного інтелекту у формі цифрових помічників	45
РОЗДІЛ 3. РЕЗУЛЬТАТИ ВИРІШЕННЯ ЗАДАЧІ.....	47
3.1. Підготовка та аналіз даних для прогнозної моделі машинного навчання інтелектуальної інформаційної системи.....	47
3.2. Побудова та вибір прогнозної моделі машинного навчання інтелектуальної інформаційної системи	57
3.3. Розробка цифрових асистентів для управління процесами прийняття рішень	64
3.4. Проектування архітектури інтелектуальної інформаційної системи планування.....	66
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ..	69
4.1.Обґрунтування можливих чинників травмонебезпечних ситуацій.....	69
4.2. Умови та обставини виникнення небезпечних ситуацій та їх наслідки	71
4.3. Безпека в надзвичайних ситуаціях	72
РОЗДІЛ 5 ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	75
ВИСНОВКИ	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	79
ДОДАТКИ	86

ВСТУП

Сільське господарство сьогодні є досить технологічною галуззю, яка передбачає застосування інноваційних технологій як в вирішенні поточних операційних та технологічних задач, так і в прийнятті управлінських рішень. Сучасна сільськогосподарська техніка та смарт-технології дозволяють оптимізувати управлінські процеси та заощаджувати обмежені ресурси в аграрній сфері. Застосування інтелектуальних інформаційних систем в аграрній сфері дозволить сільськогосподарським виробникам обробляти великі обсяги даних, прогнозувати та оптимізувати виробничі процеси.

Використання машинного навчання для обробки великих даних (метеоумов, даних, що надходять від вбудованих систем та пристроїв, ринкові показники тощо) може в загальному покращити результативність процесу прийняття рішень.

Виходячи з тенденцій розвитку інформаційних технологій поява інтелектуальних інформаційних систем була обумовлена ефективним поєднанням штучного інтелекту та інформаційних програмних продуктів, що дозволило забезпечити високий рівень обробки даних для прийняття рішень.

Розробка інтелектуальної інформаційної системи прийняття рішень з використанням моделей машинного навчання може базуватися на поєднанні на одній платформі таких технологій як машинного навчання, штучного інтелекту, технологій для збору, обробки, аналізу та використання даних: систем моніторингу, датчиків та сенсорів; автоматизованих систем управління (наприклад керування поливом); збору та аналізу даних, систем для обліку господарських операцій, прогнозування тощо.

Аналіз останніх досліджень у сфері інформаційних технологій показав, що чимало українських учених займалися обґрунтуванням та розробкою інтелектуальних інформаційних технологій для сільського господарства або окремих його галузей [1, 8], у т.ч. на основі обробки даних на основі моделей машинного навчання. Разом з цим поява генеративного штучного інтелекту на

основи нейронних мереж, можливостей створення цифрових помічників на їх основі обумовили необхідність додаткових досліджень із зазначеної тематики.

Метою виконання кваліфікаційної роботи є теоретичне та практичне обґрунтування моделі машинного навчання для інтелектуальної інформаційної системи планування процесів у сільському господарстві.

Підготовка та виконання кваліфікаційної роботи передбачала роботу над поставленими завданнями:

1. здійснення аналізу стану питання на основі останніх наукових досліджень у розвитку інтелектуальних інформаційних систем на основі штучного інтелекту та машинного навчання;

2. обґрунтувати модель машинного навчання та архітектури інтелектуальної інформаційної системи на основі штучного інтелекту для реалізації поставленої задачі;

3. здійснити вибір інструментів, технологій та мов програмування для вирішення поставленої задачі;

4. проаналізувати стан охорони праці та безпеки в надзвичайних ситуаціях сільськогосподарських підприємств на основі впровадження інформаційних систем;

5. розкрити питання ефективності застосування інтелектуальної інформаційної системи планування на основі моделей машинного навчання.

Наукова новизна кваліфікаційної роботи полягає в дослідженні, обґрунтуванні та виборі моделі машинного навчання для інтелектуальної інформаційної системи планування процесів у сільському господарстві на основі порівняння алгоритмів машинного навчання та розкриття можливостей застосування цифрових помічників для аналізу даних на основі генеративного штучного інтелекту.

Під час практичної підготовки та виконання кваліфікаційної роботи результати дослідження були апробовані на XII Міжнародній науково-практичній конференції «Інформаційні технології в енергетиці та АПК», яка відбулася у Львівському НУП 4-6 жовтня 2023 р.

РОЗДІЛ 1. АНАЛІЗ СТАНУ ПИТАННЯ В ТЕОРІЇ ТА ПРАКТИЦІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Теоретичні основи функціонування інтелектуальних інформаційних систем планування

Інтелектуальні інформаційні системи набули свого поширення в результаті розвитку та інтеграції сучасних технологій, у т.ч. обробки великих даних, та їх поширення в обчислювальних середовищах. Це дає змогу задовольняти потреби різних категорій користувачів інтелектуальних інформаційних систем планування, у т.ч. на основі застосування методів обробки природньої мови, машинного навчання, інтелектуального аналізу даних, їх обміну та зберігання, аналізу зображень, логістики інформації та накопичення знань.

Інтелектуальні інформаційні системи (*Intelligent information system*) можуть забезпечувати різні рівні інтелектуальної підтримки користувачів та прийняття ними рішень у багатьох прикладних сферах, що передбачає обробку даних на основі моделей машинного навчання або інтелектуального аналізу даних з метою отримання знань та зменшення кількості помилок при прийнятті управлінських рішень.

Інтелектуальна інформаційна система (IIS) представляє собою комплексну структуру, яка поєднує в собі різноманітні технології і методи для збору, обробки та аналізу інформації з метою прийняття управлінських рішень. Ця система може використовувати методи штучного інтелекту, обробки даних та інших інноваційних технологій для автоматизації і оптимізації різних сфер діяльності, у т.ч. і в аграрній сфері.

Таблиця 1.1. – Напрямки застосування інтелектуальних інформаційних систем

Функція	Коротка характеристика
Збір та інтеграція даних	Здатність збирати та інтегрувати різні дані в одну інформаційну систему
Інтелектуальний аналіз даних та формування висновків	Використання алгоритмів машинного навчання та інструментів штучного інтелекту для аналізу даних, встановлення залежностей і зв'язків, прийняття обґрунтованих рішень на їх основі. Покращення точності та якості аналізу у порівнянні із традиційними методами.
Автоматизація прийняття рішень	Можливість системи автоматично приймати рішення на основі аналізу даних та врахування визначених параметрів.
Системи підтримки прийняття управлінських рішень	Розробка інтерфейсів та інструментів для активної підтримки процесу прийняття рішень користувачем.
Автоматизація рутинних завдань	Застосування автоматизації для виконання повсякденних рутинних операцій, звільнюючи час для стратегічного аналізу та рішень.

Інтелектуальна інформаційна система планування процесів на думку експертів здатна ефективніше вирішувати поставлені завдання управління та планування, оскільки традиційна інформаційна система в цьому напрямку має два суттєвих недоліки:

1. Слабка адаптованість до інформаційних потреб користувача, його інформаційної підтримки в режимі реального часу;
2. Труднощі із вирішенням вирішувати завдань, які погано формалізуються.

Перераховані недоліки можуть бути усунені завдяки використанню інтелектуальних інформаційних системах, які мають такі характерні ознаки: здатністю до розвитку та навчання, розвиненим комунікативним здібностям та підтримки діалогу із користувачем (наприклад у формі чат-боту); вміння вирішувати складні та погано формалізовані задачі.

В таблиці 1.2. наведенні класи інтелектуальних інформаційних систем, які умовно розділяють на системи з інтелектуальним інтерфейсом, експертні системи для вирішення складних задач та системи, які здатні самонавчатися.

Таблиця 1.2. – Класи інтелектуальних інформаційних систем

I клас: системи з інтелектуальним інтерфейсом (комунікативні здібності):	II клас: експертні системи (рішення складних завдань):	III клас: системи здатні до самонавчання:
1. Інтелектуальні БД; 2. Природно – мовний інтерфейс; 3. Гіпертекстові системи; 4. Контексти і системи; 5. Когнітивна графіка.	1. Системи, що класифікують; 2. Системи, що до визначають; 3. Системи, що трансформують; 4. Багатоагентні системи.	1. Індуктивні системи; 2. Нейронні мережі; 3. Системи, засновані на прецедентах; 4. Інформаційні сховища.

Провівши певні узагальнення, можна виділити наступні властивості інтелектуальних інформаційних систем планування процесами в сільському господарстві:

1. Визначення методів, шляхів та засобів досягнення кінцевого результату комерційної, виходячи із сформованої або заданої мети (наприклад підвищення продуктивності чи урожайності, зменшення втрат урожаю від несприятливих кліматичних умов, шкідників та захворювань).

2. Обробка неповної інформації та формування висновків на її основі, використовуючи як достовірні, так і правдоподібні правила для породження нових знань.

3. Здатність до апроксимації, узагальнення та концептуалізації знань, що є важливим аспектом інтелектуальних систем планування. Вирішення різноманітних завдань.

4. Інтелектуальні інформаційні системи можуть обробляти та розпізнавати образи, процеси та явища у визначеній прикладній області. Разом з цим вони є інформаційно відкритими системами, що розширюють обсяг та вміст моделі світу про оточуюче середовище та сферу діяльності.

5. Інтелектуальні інформаційні системи не лише використовують існуючу інформацію, але й генерують нову – наприклад знання або дані, виступаючи в ролі виробника та джерела інтелектуальних інформаційних ресурсів.

2. Особливості застосування машинного навчання для інтелектуальних інформаційних систем планування

Сьогодні важко уявити розвиток інтелектуальних інформаційних систем та програмного забезпечення, спрямованого на прийняття рішень на основі обробки структурованих та неструктурованих даних без використання в більшій або меншій мірі сучасних інформаційних технологій, у т.ч. і штучного інтелекту та методів машинного навчання.

Загалом під технологіями штучного інтелекту (*Artificial Intelligence, AI*) розуміють широкий спектр інформаційних технологій, спрямованих на розробку інтелектуальних машин, здатних виконувати завдання, які традиційно вимагають людського інтелекту.

Машинне навчання (*Machine Learning, ML*) є сукупністю методів у галузі штучного інтелекту. Ці методи включають різноманітні алгоритми, які використовуються для створення машин, що можуть навчатися, аналізувати величезні обсяги вхідних даних і виявляти у них закономірності.

З метою дослідження впливу машинного навчання та штучного інтелекту на інтелектуальні інформаційні системи планування, проаналізуємо динаміку їх розвитку протягом останніх років.

Огляд наукових досліджень та літературних джерел показав, яким чином змінювалися підходи до розробки інтелектуальних інформаційних систем в історичній ретроспективі. Зокрема, в дослідженнях 2000-х років [2] можна спостерігати наступну класифікацію типів інтелектуальних інформаційних систем (більшість з них мають своє застосування і сьогодні):

1. Системи, керовані даними (*Data-driven systems*). Даний тип систем передбачає обробку фактичних (історичних) даних з метою побудови прогнозів для прийняття рішень. В даному випадку прийняття рішень на основі даних передбачає комплексне використання постійно оптимізованих даних, результатів аналізу та відповідного програмного забезпечення для прийняття рішень. Для прикладу бізнес-рішення у сфері електронної комерції [3] формуються на основі даних, що клієнти купували в минулому, які товари клієнт оцінив або переглянув, які товари додавав в кошик тощо.

2. Системи, керовані моделлю (*Model-driven systems*). Даний тип систем використовує математичні моделі для імітації результатів прийняття різних рішень. В подальшому цей підхід реалізувався у формі моделі для представлення всіх рівнів інформаційної системи, включаючи обмін даними, алгоритмами машинного навчання та штучного інтелекту, інтерфейсом користувача програми [4]. Наприклад використовуючи можливості продукту C3 AI Type System розробники та аналітики даних можуть розгортати виробничі додатки штучного інтелекту, великих даних і прогнозної аналітики. Обробка змін даних при цьому відбуватиметься шляхом відстеження неявних залежностей між об'єктами, функціями, алгоритмами машинного навчання в спрямованому ациклічному графі. Взаємодія із зовнішніми системами в даному випадку може здійснюватися через інтерфейси RESTful.

В цей період розробка керованих моделями систем виконувалася за допомогою структури архітектури RUPt SE (Rational Unified Process for Systems Engineering) і відповідних методів трансформації (Rational, IBM) [5].

3. Системи, керовані знаннями (*Knowledge-driven systems*). Цей тип систем передбачає використання експертних знань для прийняття рішень. Сьогодні системи, основані на знаннях, використовуються для вирішення складних задач і фактично є підмножиною штучного інтелекту [6].

Інтелектуальні інформаційні системи, побудовані на основі випадків, фактично використовують кейс-орієнтоване навчання (*Case-based learning*) як метод машинного навчання, за якого система навчається, розв'язуючи

попередні випадки, схожі на поточне завдання [7]. Застосування даного типу та методу машинного навчання може включати три фази:

1. Фазу пошуку, під час реалізації якої система шукатиме подібні випадки в базі даних та наборах даних.
2. Фазу адаптації, яка передбачає модифікацію рішення попереднього випадку у відповідності до визначеного поточного завдання.
3. Фазу оцінки як етапу оцінювання (аналізу) запропонованого рішення та порівняння його з оптимальним значенням.

Оскільки даних тип систем може бути застосований в різноманітних прикладних програмах, навчання на основі конкретних випадків може бути ефективним і у випадку прийняття рішень та планування завдань для сільського господарства.

На відміну від традиційних та популярних сьогодні систем штучного інтелекту, які використовують глибоке навчання, великі дані, інтелектуальних аналіз даних на основі виявлених закономірностей, системи, керовані знаннями, обробляють великі обсяги неструктурованих даних. Це дає змогу інтегрувати знання на основі цих даних у великих масштабах.

Прикладом таких систем сьогодні можуть бути системи Blackboard, системи на основі випадків (*Case-based systems*), системи класифікації (*Classification systems*), системи аналізу прийнятності (*Eligibility analysis systems*), експертні системи (*Expert systems*), інтелектуальні системи навчання (*Intelligent tutoring systems*), системи діагностики (*Diagnosis systems*), системи на основі правил (*Rule-based systems*), системи обробки гіпертексту (*Hypertext manipulation systems*) (рис.1.1).

Системи, керовані знаннями, фактично є інноваційними інформаційними платформами, спрямованими на збір, організацію та використання знань у підтримку прийняття рішень. Вони дозволяють ефективно управляти та використовувати обширні обсяги інформації, щоб розв'язувати завдання в складних інтелектуальних областях.

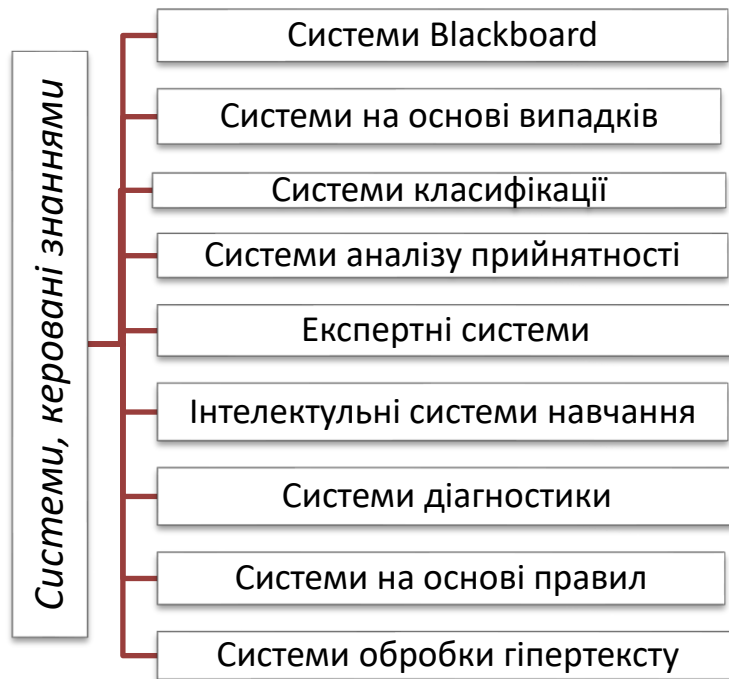


Рисунок 1.1. – Найпоширеніші типи інтелектуальних інформаційних систем, керованих знаннями.

Прийнято вважати, що до архітектури системи, заснованої на знаннях, відноситься механізм логічного висновку та безпосередньо база знань, яка містить колекцію даних. Даний тип систем може працювати в різноманітних програмах, у т.ч. і призначених для планування в аграрній сфері.

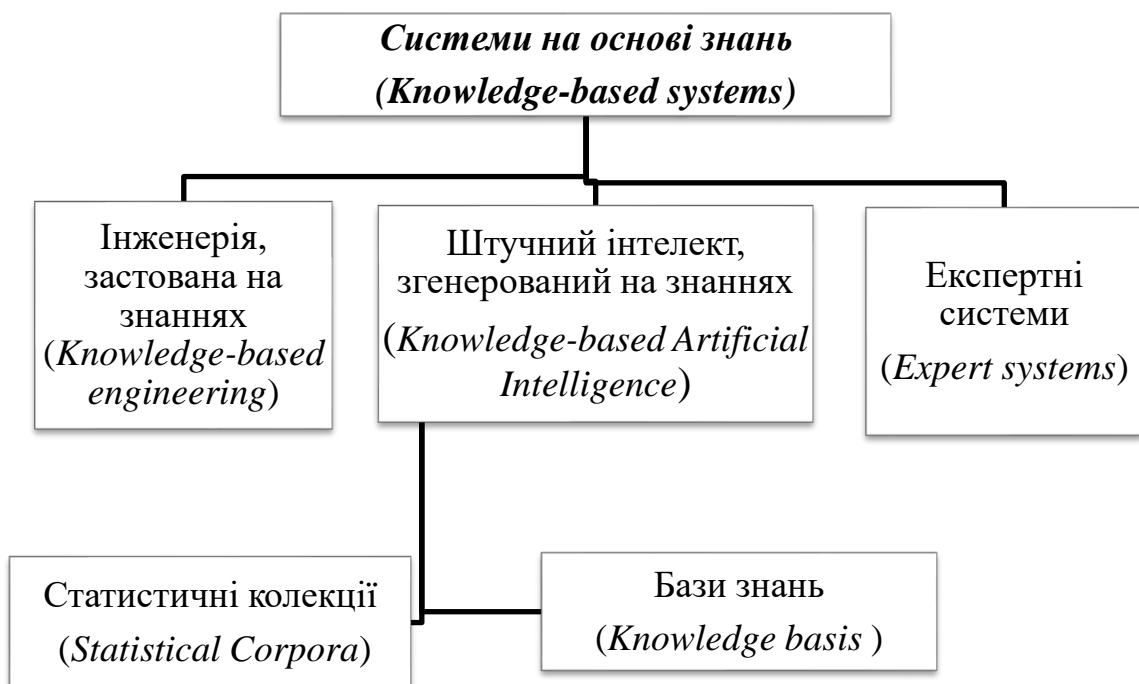


Рисунок 1.2. – Компоненти систем, побудованих на основі знань [9]

Корпус (Corpus) (в множині Corpora) може набувати різного значення у залежності від того, як його використовують [10]. Це може бути структурований і анотований набір даних, які можна імпортувати в програму на Python та проводити дослідження. В прикладній лінгвістиці може бути великою колекцією тексту чи текстових файлів, які можуть використовуватися для тренування і обробки природньої мови (англ. Natural language processing, NLP) з метою навчання та оцінки моделей машинного навчання, мовного моделювання (розуміння природньої мови, машинного перекладу, створення тексту), корпусно-лінгвістичних досліджень (як мова використовується в текстових даних реального світу), навчання моделей генерації тексту (у т.ч. і в формі чат-ботів), аналіз настроїв та емоційного тону тексту.

Обробка природньої мови (NLP) як напрям/різновид штучного інтелекту дозволяє у багатьох випадках зменшити робоче навантаження як в персональних задачах, так і виконанні робочих, часто досить трудомістких завдань. Експерти NLP [14] виділяють кілька напрямків застосування програмного забезпечення із використанням моделей штучного інтелекту на основі обробки природньої мови, які можуть позитивним чином вплинути на продуктивність працівників компаній та ефективність обробки поточних операцій та прийняття рішень (табл.1.3).

Таблиця 1.3. – Застосування програмного забезпечення на основі обробки природньої мови для підвищення ефективності працівників*

Назва	Потенційний вплив
1	2
1. Фільтрування електронної пошти	Фільтрування вхідних електронних листів на «важливі» та «спам». Зменшення часу роботи з кореспонденцією.
2. Мовний переклад	Покращення комунікації та роботи з документами, листуванням з іноземними партнерами.
3. Розумні (цифрові) помічники	Залучення у формі віртуальних асистентів для спілкування, скорочення витрат часу на пошук відповідей на запитання або вирішення рутинних завдань, у т.ч. і з обробки даних.

Продовження таблиці 1.3.

1	2
4. Аналіз документів	Пошук та пришвидшена обробка інформації, класифікація інформації в великих наборах файлів і даних, покращення управління ризиками при прийнятті управлінських рішень.
5. Пошук в інтернеті	Покращення роботи пошукових систем і відповідно якості представлених відповідей до запитів користувачів або персоналу підприємства.
6. Інтелектуальний текст	Зменшення часу на формування текстових повідомлень, введення інформації тощо.
7. Автоматичне підведення підсумків	Зменшення ручної обробки даних та помилок, які при цьому виникають.
8. Аналіз настроїв	Покращення розуміння емоцій користувачів та партнерів на основі опублікованих статей, дописів, відгуків тощо. Покращення крос-культурної комунікації та відносин з іноземними партнерами і споживачами із різних регіонів.
9. Чат-боти	Покращення якості сервісного обслуговування та комунікації з клієнтами, забезпечення належної реакції на реальні потреби клієнта.
10. Моніторинг соціальних мереж	Збір та обробка даних про клієнтів, товари та послуги. Покращення якості обробки неструктурованих даних соціальних мереж і здійснення на їх основі аналізу та прийняття рішень.

Побудовано на основі даних [14]

З позиції перспектив застосування інтелектуального аналізу тексту (*text mining*) [11] як функції інтелектуальної інформаційної системи для планування, на нашу думку заслуговують уваги такі можливі операції, реалізовані на сьогодні і в популярних моделях генеративного штучного інтелекту (у т.ч. і цифрових помічниках), як:

1. класифікація документів;
2. кластеризація/організація документів;
3. резюмування документу;
4. візуалізація простору документу;
5. створення прогнозів (цін, урожайності тощо);

б. впровадження системи рекомендацій на основі вмісту (книг, статей, відео у ютубі тощо).

На рисунку 1.3. показано можливі сфери застосування інтелектуального аналізу тексту.

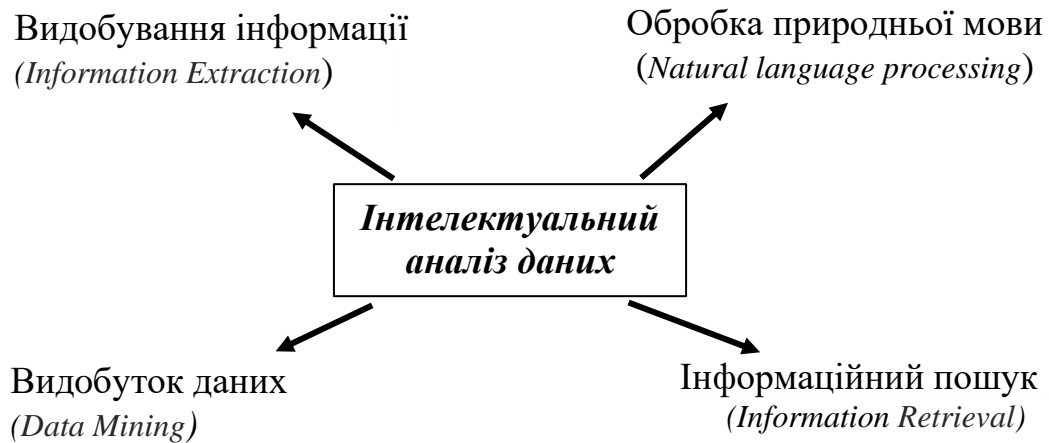


Рисунок 1.3. – Сфери застосування інтелектуального аналізу тексту в інтелектуальних інформаційних системах

Під видобуванням (вилученням) інформації експерти-практики розуміють процес аналізу неструктурованих даних і вилучення важливої інформації у тих форматах даних, які можна редагувати та структуровані [12].



Рисунок 1.4. – Робочий процес вилучення (видобування) інформації з рахунку-фактури [12]

Вилучення інформації з неструктурованих даних може відбуватися як за допомогою розглянутого вище механізму NLP-алгоритмів, так і методів оптичного розпізнавання символів OCR (*Optical character recognition*) та методів глибокого навчання (*Deep Learning*).

Інтелектуальна інформаційна система та програмне забезпечення, яка міститиме OCR з використанням можливостей штучного інтелекту та машинного навчання [13], даватиме можливість користувачу оцифровувати файли і документи, робити їх доступнішими для пошуку (наприклад перетворювати pdf-файли і зображення в редаговані формати), автоматизувати збір даних із сканованих документів/зображень. На нашу думку такі можливості будуть корисними для перевірки накладних, договорів, квитанцій тощо, що дозволить зекономити час, кошти, ресурси, зменшити кількість помилок при прийнятті рішень.

В сільському господарстві ряд операцій можуть заповнюватися в ручному форматі, паперових документах, що обумовлено особливістю даної галузі та можливими перспективами застосування програмного забезпечення на основі OCR. Наприклад завдяки API програми Nanonets можна оцифровувати документи, формувати поля та самі дані і в подальшому інтегрувати їх із повсякденними програмами (наприклад для обліку господарських операцій та планування).

В даному випадку логіка реалізації поставленої задачі буде здійснюватися наступним чином: необхідно створити нову модель за допомогою кінцевої точки (рис.1.5), вказаної на офіційному ресурсі (<https://nanonets.com/documentation/>).

Успішний виклик API поверне структуру json новоствореної моделі. Після цього можна використовувати ідентифікатор моделі, щоб завантажити зображення для кожної категорії, а потім перенавчити модель.

Після цього кінцеву точку можна використовувати для завантаження навчальних зображень для вказаної моделі, навчання та покращення моделі, отримання результатів за ідентифікатором сторінки.

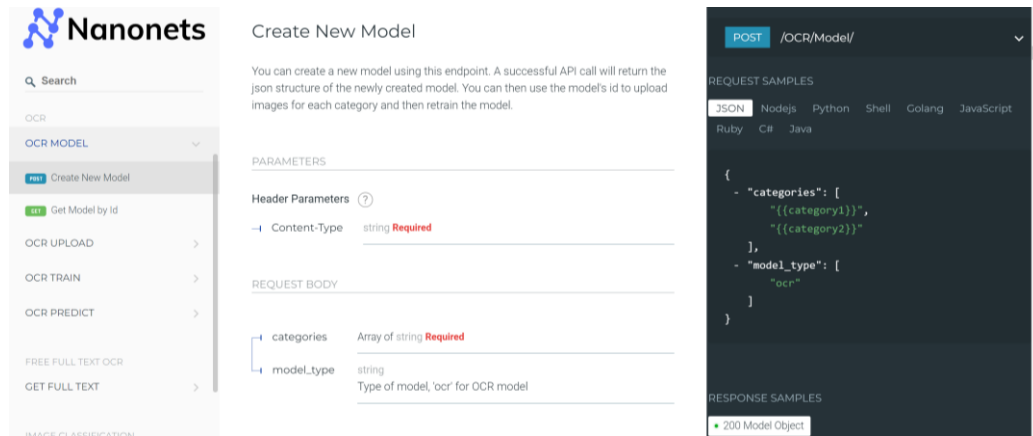


Рисунок 1.5. – Створення нової моделі на основі Nanonets

В додатку А представлена порівняльна таблиця існуючого програмного забезпечення, розробленого на основі методів оптичного розпізнавання символів OCR, аналіз якого може допомогти сформуванню користувачькі вимоги та обрати модель для інтелектуальної інформаційної системи прийняття рішень в сільському господарстві.

Робота з неструктурованими даними на основі методів глибокого навчання (*Deep Learning*) передбачає застосування нейронної мережі з трьома або більше рівнями, дозволяючи навчатися на великих обсягах даних. Глибоке навчання є фактично підмножиною машинного навчання.

На відміну від традиційних підходів у машинному навчанні, алгоритми глибокого навчання усувають частину попередньої обробки даних (що притаманно саме для машинного навчання), тому можуть приймати і обробляти неструктуровані дані (текст, зображення), автоматизувати вилучення певних функцій та усувати залежність від людини-експерта [15].

В алгоритмах глибокого навчання використовуються глибокі нейронні мережі, які маючи доступ до величезних наборів даних та інформації, фактично вивчають та аналізують їх. Як підсумок, застосування глибоких нейронних мереж в інтелектуальних інформаційних системах спрямовані на прийняття рішень на підставі переваг конкретного користувача.

Використання глибокого навчання в інтелектуальних інформаційних системах планування в аграрній сфері, побудованого на основі процесів

градієнтного спуску та зворотнього поширення, покращуватиме точність та якість обробки даних та якості прогнозів наприклад встановлення на основі зображень порід тварин, сортів сільськогосподарських культур, стану посівів, шкідників в рослинництві тощо.

Оскільки алгоритми глибокого навчання є доволі складними, це призвело до появи різних типів нейронних мереж для вирішення конкретних проблем або наборів даних. Для прикладу це можуть бути згорткові нейронні мережі (*Convolutional neural networks (CNN)*) та повторювані нейронні мережі (*Recurrent neural network (RNNs)*). Перший тип мережі використовують в основному в програмах комп'ютерного зору та класифікації зображень, оскільки вона може виявляти особливості та шаблони в зображенні, в результаті чого виконуються завдання із виявлення або розпізнавання об'єктів. Другий тип мережі використовується в програмах природної мови та розпізнавання мовлення, застосовуючи при цьому послідовні або часові ряди даних.

Обґрунтування перспектив застосування методів глибокого навчання (*Deep Learning*) розкрито в багатьох наукових дослідженнях, зокрема дослідженні Андреаса Каміларіса та Франсеска Х. Пренафета-Болду [16], результати експериментів яких наведено на рис. 1.6.

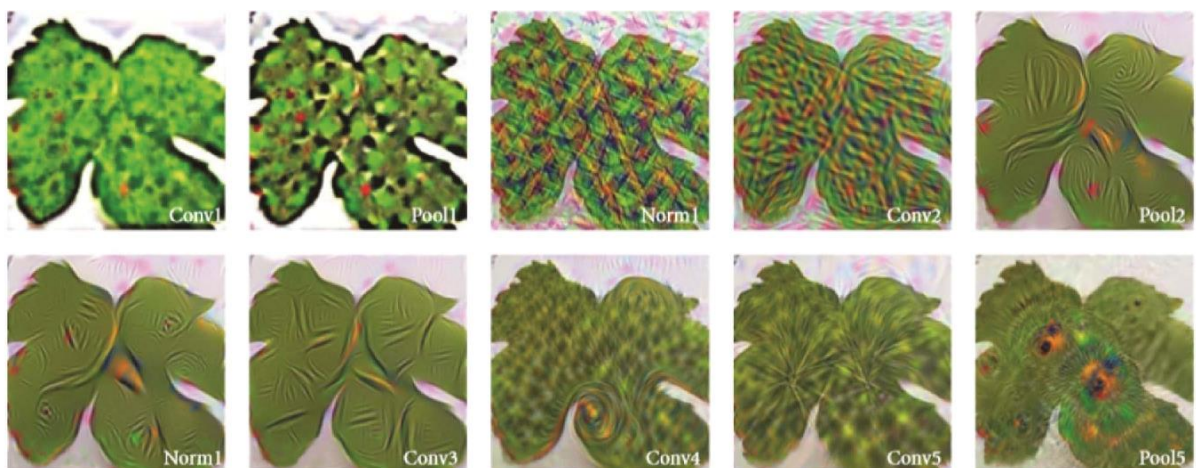


Рисунок 1.6. – Приклад виконання етапу візуалізації зображень вхідних шарів у задачі ідентифікації хвороб рослин на основі зображень листя [16]

Моделі машинного та глибокого навчання фактично можуть використовувати різні типи навчання: які зазвичай класифікуються як контрольоване навчання, неконтрольоване навчання та навчання з підкріпленням.

Контрольоване навчання застосовується у випадку обробки великих обсягів даних для класифікації або прогнозування, які можуть бути марковані (відмічені) людиною. Навпаки, неконтрольоване навчання не потребує мічених наборів даних, адже останніх в реальних ситуаціях є набагато більше. Тому в даній ситуації під час навчання виявляються шаблони в даних, здійснюється їх кластеризація їх за будь-якими відмінними характеристиками. Навчання з підкріпленням становить собою процес, у якому модель вчиться ставати точнішою для виконання дії в середовищі на основі зворотного зв'язку, що дозволяє покращити або збільшити винагороду.

У найзагальнішому випадку розрізняють два типу машинного навчання: навчання по прецедентах, або індуктивне навчання, і дедуктивне навчання. Оскільки останнє прийнято відносити до області експертних систем, то терміни «машинне навчання» і «навчання по прецедентах» можна вважати синонімами.

Навчання по прецедентах, в свою чергу, поділяють на три основних типи: контрольоване навчання, або навчання з учителем (*Supervised learning*), неконтрольоване навчання (*Unsupervised learning*), або навчання без учителя, і навчання з підкріпленням (*Reinforcement learning*).

Як і будь-яка технологія, успіх застосування машинного навчання та штучного інтелекту в прикладній сфері напряду залежить від ефективності в комерційній діяльності та вирішення поставлених бізнес-задач.

В таблиці 1.4. наведено вплив застосування машинного навчання та глибоких нейронних мереж для вирішення бізнес-цілей та бізнес-задач інтелектуальних інформаційних систем планування.

Таблиця 1.4. – Очікуваний вплив застосування машинного навчання та глибинних нейронних мереж для комерційної діяльності

<i>Чинники, які обумовлюють потребу в використанні машинного навчання / нейронних мереж</i>	<i>Бізнес-завдання, які можуть вирішитися</i>
Відсутність програм для обробки та систематизація великих даних, які збираються та накопичуються	<p>1. <i>Прогнозна аналітика</i>: попиту, цін, обсягу продаж, продуктивності, завантаження потужностей, розвитку підприємства тощо.</p> <p>2. <i>Виявлення</i>: тенденцій розвитку ринку, залежностей, прихованих взаємозв'язків, аномалій, повторюваних елементів.</p>
Дані про господарську діяльність не систематизовані, спотворені або неповні	<p>3. <i>Розпізнавання</i>: фото-, відео-, аудіоконтенту (фото клієнтів, працівників, зображення шкідників, стану земельних угідь), спроб шахрайства, брехні, внутрішніх загроз, зовнішніх атак на систему безпеки.</p> <p>4. <i>Автоматизація</i>: сервісного обслуговування, роботи операторів в онлайн-чатах, операторів служб доставки.</p>
Неструктуровані дані, які є настільки різними, що між ними важко встановити зв'язок або закономірності	<p>5. <i>Класифікація</i>: аналіз цільових груп клієнтів, маркетингова діяльність.</p> <p>6. <i>Кластеризація</i>: класифікація за параметрами, які з самого початку не були відомі (наприклад земельних угідь, які перебувають в оренді).</p> <p>7. <i>Розробка</i>: чат-ботів.</p>

Джерело: побудовано на основі даних [21]

Ефективно обрана модель машинного навчання може позитивним чином вплинути на інтелектуальну інформаційну систему планування.

На рисунку 1.7 представлено інтелектуальну інформаційну систему, засновану на використанні формалізму LPR, яка об'єднує такі базові функції, як доступ до знань, міркування, пошук та експертні поради.

На думку авторів розробки одним з можливих модулів інтелектуальної інформаційної системи може бути модуль машинного навчання.

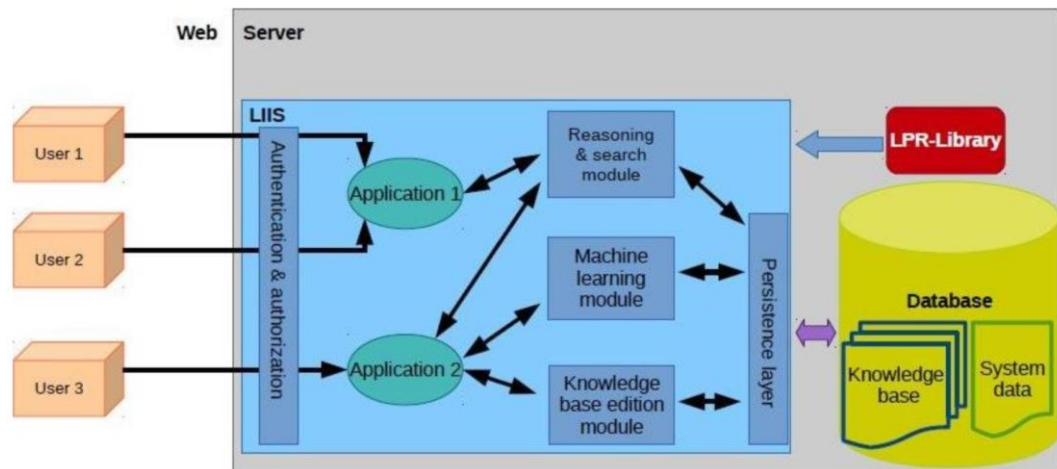


Рисунок 1.7 - Архітектура інтелектуальної інформаційної системи із модулем машинного навчання [20]




Реалізація машинного навчання на прикладі наведеної вище архітектури може ґрунтуватися на трьох ключових елементах: дані, ознаки та алгоритми. Основою даних виступатиме комерційна інформація, завдяки якій може прийматися управлінське рішення. Ознаки передбачають визначення ключових-бізнес потреб компанії, а також характеристик і властивостей системи, які мають бути враховані в ході реалізації навчання. Алгоритми передбачають фактично вибір методу для вирішення бізнес-задачі з поточного, операційного та стратегічного планування.

1.3 Огляд існуючих практик застосування інтелектуальних інформаційних систем та штучного інтелекту в сільському господарстві

Розвиток та впровадження інформаційних технологій в сільському господарстві, яке є доволі технологічною галуззю, призводить по появи нових завдань та бізнес-рішень, пов'язаних із впровадженням смарт-технологій та технологій розумного (точного) землеробства, технологічних способів обробки в рослинництві та управління полями, накопиченням великих обсягів даних в аграрній сфері, які надходять від різноманітних датчиків.

Аналіз наукової літератури та публікацій експертів показує, що в загальному під терміном «розумне сільське господарство» (*Smart agriculture*) означає використання різноманітних розумних технологій, таких як Інтернет речей, датчиків, системи визначення місцезнаходження GPS, роботів, машинного навчання та штучного інтелекту [29]. У підсумку очікуваним ефектом впровадження інтелектуальних інформаційних систем та смарт-технологій може бути покращення якості управління процесами в рослинництві та тваринництві, допоміжних галузях сільського господарства, продуктивності та економічних показників, показників якості продукції.

Таблиця 1.5. - Порівняльна характеристика програмного забезпечення для аграрної сфери з елементами машинного навчання

Назва програмного забезпечення	Опис призначення
	Використовує машинне навчання для аналізу даних про господарство, щоб допомогти фермерам приймати кращі рішення. Можна здійснювати прогнозування урожайності, відстеження стану посівів, встановлення оптимальних режимів зрошення тощо.
	Використовує машинне навчання для аналізу даних про погоду, ґрунт, історичні дані про урожайність та інші чинники, аналіз яких допомагає фермерам управляти технологічними процесами в рослинництві. Платформа може прогнозувати ризиками появи шкідників та хвороб, обирати оптимальні строки посіву та збирання врожаю тощо.
	Використання системи машинного навчання для роботизованого видалення бур'янів. Система використовує камери та датчики для виявлення бур'янів, а потім використовує роботизовані руки для їх видалення.
	Використання системи машинного навчання для аналізу даних та управління процесами в тваринництві. Система може прогнозувати ризик захворювань, рекомендувати оптимальні раціони годівлі тощо.

Інтелектуальні технології докорінно змінюють саму концепцію ведення господарської діяльності в аграрній сфері, впроваджуючи різноманітні новітні технології, такі як ГІС технології та GPS в сільському господарстві, супутникових знімків, дронів, інформаційні технології та машинне навчання

для обробки даних, платформи аналітики та оптимізації, колекції та набори даних.

В додатку Б наведено порівняльний аналіз застосування окремих типів смарт-технологій в аграрній сфері та їх вплив на інтелектуальну інформаційну систему для планування та управління процесами.

Галузь рослинництва характеризується тим, що фермер або агроном змушений приймати рішення в умовах неповної та неточної вхідної інформації (наприклад щодо підживлення та удобрення рослин на різних ділянках поля). Вирішити це складне завдання можна на основі інтелектуального аналізу даних з використанням математичних моделей та інтелектуальних систем. Практичним результатом застосування інтелектуальних інформаційних технологій з використанням технологій машинного навчання планування процесів можуть бути розроблені технологічні карти, різноманітні додатки та користувацькі інтерфейси, які передбачатимуть також їх використання в польових умовах. Застосування автоматизованій інтелектуальній процес планування в точному землеробстві призводить до високої точності та цілодобового контролю за станом посівів та полів.

В тваринництві інтелектуальні інформаційні системи планування процесами можуть допомагати фермерам краще контролювати потреби окремих тварин, коригувати раціони годівлі, запобігати захворюванням та зміцнювати здоров'я стада. Використовуючи бездротові додатки IoT фермери можуть здійснювати для моніторинг місцезнаходження, самопочуття та здоров'я тварин. Використовуючи результати обробки даних та зображень на основі методів машинного та глибинного навчання, можна здійснювати ідентифікацію хворих тварин, їх відокремлення від стада з метою запобігання поширення хвороби.

Управління процесами в розумні теплицях на основі інтелектуальних інформаційних технологій та Інтернету речей може бути зосереджене в першу чергу на контролі мікроклімату теплиці та автоматизацію управління нею.

Прикладом практичного застосування в сільському господарстві системи розпізнавання символів OCR, яка заснована на методах глибинного навчання, є оцифрування відсканованих сільськогосподарських нормативних документів, що містять сертифікати та етикетки. На думку науковців та практиків розпізнавання сертифікатів і етикеток відноситься до складних завдань, адже скановані зображення друкованої форми можуть мати різні макети та розміри тексту, локальні мови і нормативні вимоги щодо оформлення. У дослідженні на задану проблематику [30] науковцями було здійснено оцінку та порівняння найсучасніших моделей виявлення та розпізнавання тексту на основі глибокого навчання, а також особливості застосування бібліотеки OCR Tesseract для вирішення поставленого практичного завдання.

На сьогодні Tesseract (Google Tesseract OCR) є одним з найкращих безкоштовних рішень OCR з відкритим кодом для більшості організацій, добре відоме, має багато варіантів використання, тому може використовуватися при розробці програмного забезпечення для вирішення цих складних задач. Основною функцією даного механізму є витягування друкованого або написаного тексту із зображень, що важливо для розпізнавання етикеток або інших сканованих (сфотографованих) документів (рис. 1.8).

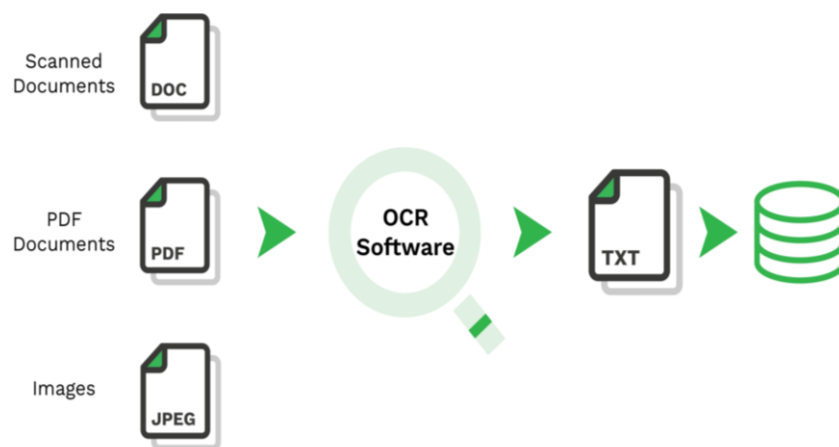


Рисунок 1.8 – Механізм розпізнавання тексту із зображень на основі OCR Tesseract

Використання інтелектуальних інформаційних систем може мати вплив і в пов'язаних галузях до сільського господарства, наприклад харчової промисловості, адже при переробці сільськогосподарської продукції якість сировини, характеристика, вміст корисних і шкідливих елементів мають важливе значення, особливо при реалізації простежуваності продуктів харчування, міжнародних стандартів якості стандартів HACCP, ISO та інших.

В додатку В наведено приклад структури та функціональні можливості інтелектуальної системи, розробленої експертами в ході виконання досліджень застосування штучного інтелекту для покращення харчування клієнтів та вибору страв у відповідності до їх калорійності та наявних інгредієнтів.

В якості прикладу можливого практичного застосування інтелектуальних інформаційних систем для планування процесів в сільському господарстві може бути експертна система Quantum (Quantum Expert System) на основі проекту НВК «Квадрат» (ТМ Quantum), описана у фахових інтернет-ресурсах для аграріїв Полянчиковим С. та Капітанською О. [28]. Автори зазначають, що даний тип інтелектуальної експертної системи може допомагати моделювати знання експертів з питань діагностики, прогнозування та планування в агрономії. Сама ж система, побудована на основі аналізу вхідних значень, використовує накопичений експертний досвід з питань ґрунтознавства, агрохімії та фізіології рослин, а її потенційними користувачами можуть бути фермери, агрономи, менеджери та інші фахівці.

Загальна ідея експертної системи полягає у створенні багатофункціонального сервісу для фахівців з аграрної сфери, який узагальнить знання, реалізуватиме прогнозну аналітику, мінімізуватиме ризики та надаватиме відповіді на складні питання, виходячи з мінливих умов ведення сільськогосподарської діяльності (рис.1.9).



Рисунок 1.9 - Алгоритм дії експертної системи Quantum [28]

Як початкові дані модель приймає два параметри: культуру та поле, де планується посів. На першому етапі, використовуючи цю інформацію разом із попередніми наборами вхідних даних, Quantum Expert System (QES) аналізує погодні дані, тип і наявні результати аналізу ґрунту в регіоні, співставляє з даними історії полів та картами врожайності на конкретному полі. Вже на цьому етапі система може надати попередні рішення щодо строків посіву, підбору гібридів і норм внесення добрив.

Розробка та використання інтелектуальних інформаційних систем на з використанням розглянутих вище типів систем та моделей машинного навчання може допомагати сільськогосподарським виробникам покращувати якість управління процесами, підвищувати продуктивність, ефективність та рентабельність господарської діяльності, економити ресурси та та якісно впливати на екологічні показники стану природних ресурсів, які використовуються в сільському господарстві.

РОЗДІЛ 2.

ОБҐРУНТУВАННЯ, ВИБІР ТА РЕАЛІЗАЦІЯ ІНСТРУМЕНТАРІЮ ВИРІШЕННЯ ЗАДАЧІ

2.1 Обґрунтування методу машинного навчання інтелектуальної інформаційної системи планування

Застосування інтелектуальних інформаційних технологій в сільському господарстві на основі штучного інтелекту для планування передбачає вирішення ряду складних задач, які виникають під час здійснення господарської та комерційної діяльності. Попередній аналіз показав, що все більше сільськогосподарських виробників використовують смарт-технології в рослинництві, тваринництві та допоміжних виробництвах. Це дає змогу збирати великі обсяги структурованих та неструктурованих даних, на основі обробки яких з використанням методів машинного навчання можна будувати та навчати моделі для прогнозувальної аналітики.

Побудова прогнозувальної моделі для сільського господарства на основі методів машинного навчання відкриває нові перспективи управління ресурсами, планування урожайності, подолання виробничих ризиків тощо. Цей інноваційний підхід як функція інтелектуальної інформаційної системи дозволяє більш точно прогнозувати різноманітні чинники та показники на перспективу.

З метою обґрунтування методу машинного навчання інтелектуальної інформаційної системи планування в ході виконання завдань, поставлених в даній кваліфікаційній роботі, було обрано задачу планування вирощування озимої пшениці на основі даних одного із сільськогосподарських підприємств регіону із врахуванням вибірки конкретних параметрів, на основі яких буде побудована модель.

Загалом під прогнозним моделюванням (*Predictive modeling*) розуміють статистичний підхід, який аналізує моделі даних для визначення майбутніх подій або результатів на основі підходів машинного навчання, інтелектуального аналізу даних для прогнозування діяльності, поведінки та тенденцій на основі поточних і минулих даних [31]. Прогнозне моделювання як і машинне навчання в загальному розумінні передбачає використання алгоритмів та побудову прогнозних моделей. Найпоширеніші прогностичні моделі включають дерева рішень, регресії (лінійні та логістичні) та нейронні мережі, які є відносно новою областю методів і технологій глибокого навчання [32]. Незважаючи на те, що досить часто ототожнюють прогнозне моделювання з машинним навчанням, між ними є певні відмінності, які слід враховувати при роботі з інтелектуальними інформаційними технологіями планування процесів у сільському господарстві (табл. 2.1).

Таблиця 2.1. – Порівняльні характеристики прогнозного моделювання та машинного навчання

Характеристика	Прогнозне моделювання (<i>Predictive modeling</i>)	Машинне навчання (<i>Machine learning</i>)
Загальний опис	процес створення моделі, яка може прогнозувати майбутні значення показників на основі історичних даних	сфера штучного інтелекту, пов'язана з розробкою та дослідженням статистичних алгоритмів, які вивчають та узагальнюють дані та виконують завдання без чітких інструкцій.
Тип алгоритму	використовує статистичні методи, такі як регресія, логістична регресія, дерева рішень та інші.	використовує більш складні алгоритми, такі як нейронні мережі, опорні вектори, алгоритми на основі градієнтного бустінгу та інші.
Навчання	моделі зазвичай навчаються на фіксованому наборі даних	моделі можуть навчатися та вдосконалюватися з часом, коли з'являються нові дані
Інтерпретація	моделі зазвичай простіші для інтерпретації та розуміння	моделі можуть бути складними та важкими для інтерпретації

На основі наведеного вище порівняння можна зробити висновок, що прогнозне моделювання та машинне навчання – це два схожих, однак відмінних підходи, які можуть бути реалізовані в інтелектуальних інформаційних системах для прогнозування. На практиці вибір кращого підходу для реалізації може відбуватися конкретної задачі та набору даних, адже жоден алгоритм машинного навчання не працює найкраще для кожної з існуючих проблем в управлінні та плануванні, а особливо це актуально для навчання під наглядом (тобто прогнозного моделювання).

Дослідники в цілому виділяють такі найпоширеніші алгоритми машинного навчання: лінійна регресія, логістична регресія, лінійний дискримінантний аналіз, дерева класифікації та регресії, K-найближчі сусіди (KNN), навчання векторного квантування (LVQ), випадковий ліс, машини опорних векторів, наївний басів класифікатор (Naive Bayes), підсилення, метаалгоритм статистичної класифікації (AdaBoost) [33].

Для вибору алгоритму машинного навчання у випадку вирішення конкретної задачі необхідно врахувати наступні чинники:

- 1) розмір, якість та характеру даних;
- 2) доступність часу для обчислення;
- 3) терміновість поставленого завдання;
- 4) що ви хочете зробити з даними та цілі дослідження.

Вибір алгоритму машинного навчання для прогнозування залежить від характеристик набору даних. Наприклад, для обробки структурованих даних та прогнозу часових рядів може бути ефективним використання таких як алгоритм випадкового лісу (*Random Forest*) або довгої короткочасної пам'яті (*Long Short-Term Memory (LSTM)*). При роботі з великим обсягом даних та надзвичайно складними взаємозв'язками потужним інструментом для точних прогнозів можуть виявитися алгоритми глибинного навчання (*Deep Learning*), зокрема нейронні мережі (*Neural Networks*).

В таблиці 2.2. описано процес вибору алгоритму машинного навчання на основі рекомендацій практиків у сфері штучного інтелекту та інтелектуальних інформаційних технологій.

Таблиця 2.2. – Процес вибору алгоритму машинного навчання

Номер етапу	Опис
Крок 1. Визначення проблеми	Необхідно розпочати із чіткого визначення суті проблеми. Окреслити задачу: яку саме проблему необхідно вирішити або який прогноз побудувати.
Крок 2. Збір та підготовка даних	Переконатися в якості та релевантності даних. Чи вони відповідають заявленій задачі.
Крок 3. Вибір категорії алгоритму	Алгоритми машинного навчання фактично розділити на три основні типи: <i>Контрольоване навчання</i> : використовується для даних із мітками, де потрібно передбачити результат (класифікація чи регресія). <i>Неконтрольоване навчання</i> : використовується для немаркованих даних, де потрібно знайти шаблони або кластери. <i>Навчання з посиленням</i> : використовується для прийняття рішень у послідовних даних.
Крок 4. Вибір конкретного алгоритма	На цьому етапі у вибраній категорії необхідно обрати буде вибрати певний алгоритм. Наприклад: <i>Для алгоритмів контрольованого навчання</i> : логістична регресія, випадковий ліс. <i>Для алгоритмів неконтрольованого навчання</i> : кластеризація K-Means (для групування схожих точок даних), аналіз основних компонентів PCA (для зменшення розмірів даних із збереженням їх основних функцій). <i>Для алгоритмів навчання з посиленням</i> : Q-навчання (застосовується в іграх ШІ та робототехніки, навчаючи агентів приймати рішення з часом).
Крок 5. Оцінка та порівняння	Необхідно оцінити та порівняти вибрані алгоритми за допомогою таких показників, як точність, точність, запам'ятовуваність або оцінка F1. Доцільно обрати той алгоритм, який найкраще працює з даними.
Крок 6. Точне налаштування та оптимізація	Необхідно точно налаштувати обраний алгоритм, задавши гіперпараметри. Цей процес може вимагати експериментів.
Крок 7. Впровадження та моніторинг	Вибравши алгоритм і налаштувавши його, можна застосувати його у програмі. Необхідно постійно відстежувати ефективність і за потреби перенавчайте нові дані.

Побудовано на основі даних [34].

Одним з важливих етапів вибору алгоритму та моделі машинного навчання є оцінка якості моделі, яка дозволяє визначити, наскільки точно та надійно модель виконує прогнози на нових даних, а також може виявляти можливі перекоси чи недоліки. Цей етап фактично є підтвердженням ефективності та адаптації моделі до конкретних завдань та допомагає в уникненні перенавчання чи недонавчання. Оцінка якості моделі з позиції інтерпретації результатів може допомогти зрозуміти, чому модель робить певні прогнози, що важливо для інтелектуальних інформаційних систем планування та прийняття обґрунтованих рішень.

Показники оцінювання обраного алгоритму та побудованої моделі можуть розглядатися з позиції покращення точності, ефективності, якості та результативності моделей машинного навчання, спираючись на об'єктивні та кількісні показники ефективності. На основі рекомендацій практик у сфері машинного навчання [35] узагальнено завдання оцінювання (рис.2.1) та наведені показники оцінювання моделей у відповідності до різних типів задач (табл.2.3).



Рисунок 2.1. – Перелік завдань щодо застосування метрик оцінки якості моделей машинного навчання

На практиці можуть одночасно використовуватися різні критерії для оцінки ефективності моделі, оскільки модель може демонструвати високі результати за одним показником та працювати неоптимально за іншим. У таких ситуаціях фахівці стараються знайти розумний баланс між різними оцінками.

Таблиця 2.3. – Показники оцінювання різних завдань з моделями машинного навчання*

Типи завдань	Показники оцінювання моделі
Класифікаційні завдання	точність, точність запам'ятовування, оцінка F1, площа під ROC-кривою (AUC-ROC)
Завдання регресії	середня квадратична помилка (MSE), середньоквадратична помилка (RMSE), середня абсолютна помилка (MAE), R-квадрат
Завдання кластеризації	оцінка Silhouette, індекс Данна, індекс Ренда
Завдання ранжування та рекомендації	середня середня точність (mAP), нормалізований дисконтований кумулятивний приріст (NDCG), точність і відкликання на K.

*-узагальнено на основі даних [35]

Застосування машинного навчання в різних інтелектуальних інформаційних системах може передбачати досягнення різноманітних, у т.ч. і унікальних цілей. Відповідно такі моделі матимуть різні параметри та властивості. Тому немає універсального підходу до оцінки моделей ML для різних завдань.

Для вирішення поставленої задачі, а саме планування показників та процесів вирощування озимої пшениці у сільськогосподарському підприємстві на основі зібраних та підготовлених даних здійсимо вибір алгоритму для моделі машинного навчання.

Загалом виділяють дві основні категорії задач машинного навчання в залежності від характеру цільової змінної: задачі класифікації та регресії (табл.2.4).

Таблиця 2.4. – Порівняльний аналіз задач класифікації та задач регресії

Критерій	Порівняльна характеристика	
	<i>Класифікація</i>	<i>Регресія</i>
Цільова змінна	Цільова змінна визначає клас або категорію, до якої належить об'єкт (наприклад, розпізнавання обличчя, класифікація електронного листа на "спам" чи "не спам").	Цільова змінна є числовою та представляє собою безперервне значення (наприклад, прогнозування ціни на нерухомість, температури, часу тощо).
Характер вихідних даних	Вихідні дані у завданнях класифікації складаються з категорій або класів, і модель вирішує, до якого класу відноситься об'єкт.	Результати регресійних задач виражаються числовими значеннями, які представляють собою прогнозовану кількість чи величину.
Тип виходу моделі	Модель класифікації надає ймовірності або визначає конкретний клас для кожного вхідного об'єкта	Модель регресії генерує числове значення, що вказує на очікуване числове відношення чи результат.
Метрика оцінки	Для класифікаційних задач використовуються метрики, такі як точність, чутливість, специфічність, F1-середнє тощо	У регресійних задачах використовуються метрики, такі як середньоквадратична помилка (MSE), середня абсолютна помилка (MAE) та інші.

Лінійна регресія фактично представляє собою групу алгоритмів, які застосовуються під час вирішення задач на основі керованого машинного навчання. Традиційно у сфері керованого машинного навчання використовують дві основні категорії задач: класифікацію та регресію. Алгоритми лінійної регресії представляють другу категорію, характеризуючись властивістю працювати з числовими безперервними значеннями цільової змінної.

У класифікаційних завданнях цільова змінна представляє собою категоричне значення, таке як "так/ні", "чорний/синій/білий", "спам/не спам" і т. д. У випадку лінійної регресії цільовою змінною є числове безперервне

значення, і сам алгоритм призначений для прогнозування цього числа, а не конкретного класу чи категорії. Завдання класифікації вимагають від моделей машинного навчання вивчення закономірностей у наборі даних для правильної категоризації нових прикладів.



Рисунок 2.2. – Простий графік лінійної регресії [36]

Обравши групу алгоритмів, які відносяться до типу задач лінійної регресії, розглянемо детальніше дане питання. Оскільки даний тип задач можна розв'язувати на основі методів контрольованого навчання, для порівняння було обрано наступні алгоритми: найменших квадратів (OLS), градієнтного бустингу (XGBoots), лінійної регресії з поліноміальними ознаками (таблиця 2.5).

Стандартним методом підгонки як лінійної, так і поліноміальної регресії наприклад в мові програмування R є метод найменших квадратів. Цей метод передбачає мінімізацію суми квадратів залишків у моделі шляхом коригування значень перетину та коефіцієнтів.

Поліноміальна лінійна регресія є розширенням простої лінійної регресії, яке дозволяє адаптувати поліноміальні функції до вхідних даних. У цьому виді регресії взаємозв'язок між незалежною змінною (x) та залежною змінною (y) моделюється поліноміальним рівнянням ступеня « n ». Ступінь визначає максимальний порядок поліноміальної функції, що використовується в рівнянні [39].

Таблиця 2.5. – Порівняльна характеристика обраних алгоритмів машинного навчання

Назва алгоритму	Визначення	Опис
Звичайна регресія найменших квадратів (OLS)	Стратегія оптимізації, яка дозволяє знайти пряму лінію, максимально наближену до точок даних у моделі лінійної регресії.	Зв'язок між двома змінними можна пояснити такою формулою: $y_i = \alpha + \beta x_i + \varepsilon_i$ <p>Метою простої лінійної регресії є знаходження тих параметрів α і β, для яких похибка мінімізована. Ця процедура називається звичайною помилкою найменших квадратів (OLS). Звичайне рівняння найменших квадратів:</p> $\hat{\alpha} = \min_{\alpha} \sum_{i=1}^n (y_i - \alpha - \beta x_i)^2 = \min_{\alpha} \sum_{i=1}^n \varepsilon_i^2$ $\hat{\beta} = \min_{\beta} \sum_{i=1}^n (y_i - \alpha - \beta x_i)^2 = \min_{\beta} \sum_{i=1}^n \varepsilon_i^2$
Лінійна регресія з поліноміальними ознаками	Поліноміальна регресія є розширенням стандартної моделі лінійної регресії.	Загальна форма поліноміального рівняння регресії така: $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n$ <p>де y представляє залежну змінну, x є незалежною змінною, а $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ є коефіцієнтами.</p>
Гرادієнтний бустинг (XGBoots)	Масштабований інструментарій для навчання на основі узагальненого дерева рішень з градієнтним прискоренням (GBDT)	Модель у контрольованому навчанні зазвичай відноситься до математичної структури, за допомогою якої здійснюється прогноз y_i складається з вхідних даних x_i . Типовим прикладом є лінійна модель, де передбачення подається як лінійна комбінація зважених вхідних характеристик: $\hat{y}_i = \sum_j \theta_j x_{ij}$ <p>Завдання навчання моделі зводиться до пошуку найкращих параметрів θ, які найкраще відповідають навчальним даним x_i і мітки y_i.</p>

Побудовано на основі [36-40]

До переваг застосування поліноміальної лінійної регресії можна віднести те, що вона дозволяє моделювати складні зв'язки (які складно охопити звичайною лінійною регресією), фіксувати нелінійні моделі та кривизну даних. Її застосування може забезпечити кращу відповідність даним, а відповідно і якості прогнозування. Оскільки вона схожа до лінійної регресії, її можна інтерпретувати, досліджуючи вплив незалежних змінних на залежну змінну.

До недоліків відносять те, що із збільшенням ступеня полінома модель стає більш гнучкою, може точніше відповідати навчальним даним, а її складність при цьому зростає. Модель має підвищені вимоги до обчислень, і відповідно ресурсів та часу в порівнянні із простою лінійною регресією та є чутливою до викидів у даних.

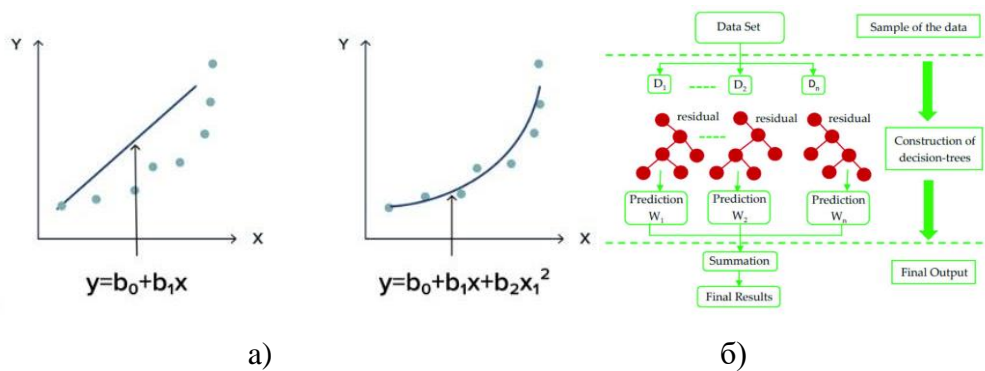


Рисунок 2.3. – Порівняння (а) лінійної регресії (звичайної та з поліноміальними ознаками) та (б) XGBoost [39, 40]

Гradientний бустинг або Extreme Gradient Boost (XGBoost) на сьогодні є найкращою бібліотекою машинного навчання для задач регресії, класифікації та ранжування, яка включає в себе паралельний бустинг дерев [36]. Контрольоване машинне навчання спочатку використовує алгоритми для навчання моделі шляхом виявлення закономірностей у наборі даних з мітками та об'єктами, а потім здійснює прогнозування міток на нових об'єктах набору даних за допомогою уже навченої моделі.

За допомогою дерева рішень створюється модель на основі аналізу запитань у формі "якщо-тоді-інше" та визначається мінімальна кількість

запитань, необхідних для оцінки ймовірності прийняття правильного рішення. Ці дерева можуть передбачати як категоріальні мітки, так і безперервні числові значення. Тому можна зауважити, що градієнтний бустинг дерев рішень (GBDT) фактично представляє собою метод ансамблевого навчання, який базується на деревах рішень і використовується для класифікації та регресії.

Підсумовуючи вище сказане, можна зазначити, що XGBoost використовується для вирішення проблем із контрольованим навчанням, де використовуються навчальні дані (з кількома функціями) x_i , щоб передбачити цільову змінну y_i . Цю особливість зазначено і в офіційній документації бібліотеки [38].

Одним із завершальних кроків вибору алгоритму машинного навчання є їх налаштування та оптимізація. Переважно цей процес включає в себе підбір оптимальних гіперпараметрів, використання ефективних методів нормалізації даних та вдосконалення моделей для досягнення кращої узагальнюючої здатності та точності прогнозів.

Наступним етапом вибору алгоритмів машинного навчання є їх впровадження та моніторинг. Під впровадженням алгоритмів машинного навчання розуміють процес, який передбачає інтеграцію розроблених моделей у реальне виробниче середовище. Важливим етапом у відповідності до цього є розгортання інфраструктури для моніторингу та валідації продуктивності моделей у реальному часі.

Здійснюючи постійний моніторинг можна своєчасно виявляти зміни в даних та адаптувати моделі, що дасть змогу забезпечити їх ефективність та актуальність відповідно до мінливих умов. Враховуючи динаміку даних, оновлення алгоритмів та підтримка відповідності стандартам є важливими аспектами успішного використання систем машинного навчання в інтелектуальних інформаційних системах.

2.2 Вибір інструментарію для вирішення задачі

Практичне впровадження моделей машинного навчання як складової інтелектуальної інформаційної системи планування є важливою складовою процесу і має базуватися на виборі інструментарію для реалізації моделей.

Це завдання вимагає ретельного вивчення різноманітних інструментів, врахування особливостей завдання та вибір оптимальних рішень для досягнення мети і досягнення успішних результатів у відповідності із завданнями та цілями проекту.

Для вибору оптимального інструменту вирішення поставленої задачі в кваліфікаційній роботі та обраного завдання для практичної реалізації моделі машинного навчання було проаналізовано найбільш популярні інструменти та подано їх узагальнену характеристику (додаток Г).

Більш детальніший аналіз окремих популярних інструментів для роботи з моделями машинного навчання наведено в таблиці 2.6.

Таблиця 2.6. – Порівняльна характеристика інструментів для реалізації моделей машинного навчання

Назва інструментарію	Короткий опис застосування
1	2
<i>TensorFlow</i>	Платформа з відкритим кодом для машинного навчання на основі бібліотеки Python, розроблена Google. Використовується для широкого спектру задач, включаючи дослідження, розробку та розгортання моделей машинного навчання та пропонує ряд інструментів.
<i>PyTorch</i>	Платформа з відкритим кодом для машинного навчання, розроблена Facebook на основі бібліотеки Torch. Застосовується для задач комп'ютерного бачення та обробки природної мови.
<i>scikit-learn</i>	Бібліотека з відкритим кодом для машинного навчання на Python, яка використовується для широкого спектру задач, включаючи класифікацію, регресію, кластеризацію та обробку природної мови. Надає функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації, таких як лінійна регресія, випадковий ліс, градієнтний бустинг і працює разом із бібліотеками NumPy та SciPy.

Продовження таблиці 2.6

1	2
<i>Keras</i>	Відкрита нейромережна бібліотека, написана на мові Python. API високого рівня для TensorFlow та PyTorch. Використовується для спрощення процесу створення та тренування моделей машинного навчання і пропонує простий у використанні інтерфейс для їх створення.
<i>Apache Spark</i>	Платформа для розподіленої обробки даних. Використовується для тренування моделей машинного навчання на великих наборах даних. Пропонує широкий спектр інструментів для розподіленої обробки даних.
<i>Jupyter Notebook</i>	Інтерактивне середовище для програмування, яке використовується для аналізу даних, візуалізації даних та машинного навчання. Становить собою проект і спільноту, метою яких є «розробка програмного забезпечення з відкритим кодом, відкритих стандартів та служб для інтерактивних обчислень на десятках мов програмування».

Для реалізації поставленого завдання з обґрунтування інструментарію для реалізації моделі машинного навчання інтелектуальної інформаційної системи планування було обрано інтерактивне середовище Jupyter Notebook (<https://jupyter.org/>).

Можна виділити наступні переваги використання інструменту Jupyter Notebook:

1. Інтерактивність у роботі з кодом, оскільки Jupyter Notebook дозволяє запускати код та бачити результати одразу, що робить його зручним для експериментів та дослідження.
2. Jupyter Notebook пропонує широкий спектр інструментів для візуалізації даних, що робить його зручним для аналізу даних та розуміння результатів машинного навчання.
3. Jupyter Notebook дозволяє вставляти текст, зображення та відео у нотатки, що робить його зручним для документування роботи та пояснення результатів дослідження.
4. Сприятливі умови для командної та проектної роботи, оскільки Jupyter Notebook можна легко ділитися з іншими.

Інтерактивне середовище Jupyter Notebook працює з так званою двопроесною моделлю, заснованою на інфраструктурі ядро-клієнт. Дана модель застосовує подібну концепцію до середовища програмування REPL (Read-Evaluate-Print Loop), яке приймає вхідні дані одного користувача, оцінює їх і повертає результат користувачеві [42].

Фактично даний інструмент є веб-програмою з відкритим вихідним кодом, яка дозволяє створювати та ділитися документами, що містять живий код, рівняння, візуалізації та описовий текст. Результати у формі блокнотів автоматично зберігаються на диску у форматі JavaScript Object Notation (JSON) з відкритим кодом і з розширенням .ipynb.

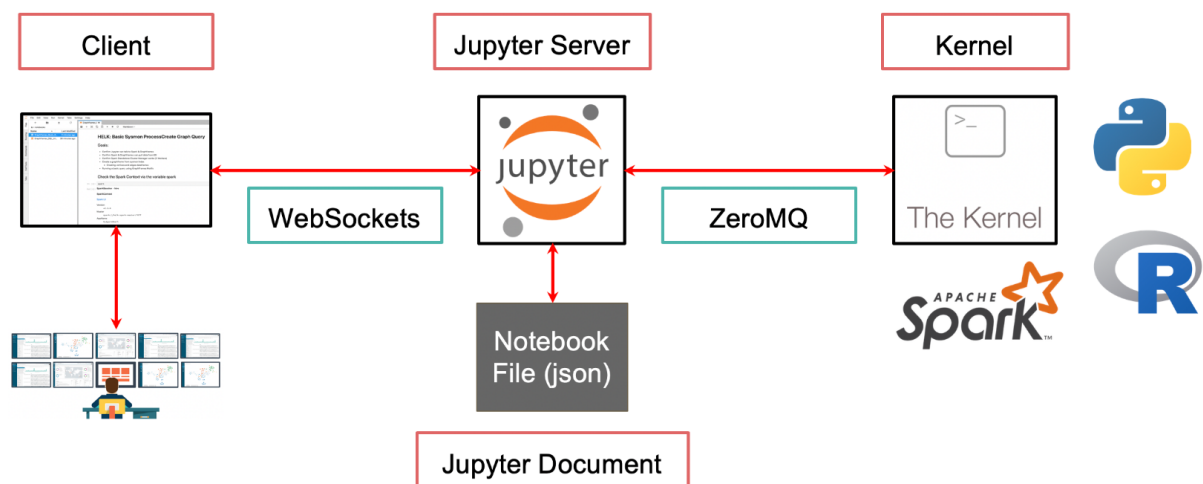


Рисунок 2.4 – Основні компоненти Jupyter Notebook [42]

Інтерактивне середовище Jupyter Notebook може використовуватися для широкого спектру задач машинного навчання, включаючи: очищення даних та підготовки їх до машинного навчання; аналізу даних та розуміння їх характеристик; розробки та тренування моделей машинного навчання; оцінки якості моделей машинного навчання; розгортання моделей машинного навчання в реальних системах.

Наведемо приклад застосування блокноту Jupyter для виконання поліноміальної регресії в Python [39].

1. *Імпорт бібліотек Python.* На цьому етапі можна використовувати такі бібліотеки Python:

- NumPy (для виконання певних математичних операцій);
- Pandas (для зберігання даних у фреймах даних pandas);
- matplotlib.pyplot (для побудови даних).

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

Рисунок 2.5. – Імпорт бібліотек Python

2. *Завантаження набору даних.* Дані імпортуються в DataFrame та зберігаються в табличному форматі.

```
df = pd.read_csv( 'Data[1].csv' )
df.head()
X = df.iloc[:, :- 1 ].values
y = df.iloc[:, - 1 ].values
```

Рисунок 2.6. – Завантаження набору даних

3. *Розділення набору даних на набір для навчання та тестування.* На цьому етапі необхідно імпортувати функцію `train_test_split` з модуля `sklearn.model_selection`. Він надає різні методи поділу даних на підмножини для навчання моделі, оцінки та перевірки.

```
from sklearn.model_selection import train_test_split X_train, X_test,
y_train, y_test = train_test_split(X,y,test_size= 0.30 ,random_state= 42 )
```

Рисунок 2.7. – Розділення набору даних

В даному прикладі `X` та `y` представляють вхідні функції та відповідні цільові значення. Параметр `test_size` визначає пропорцію даних, які мають бути виділені для тестування. У випадку, представленою на рисунку 2.7., 30%

даних буде використано для тестування (`test_size=0,3`), а решта 70% – для навчання.

Функція `train_test_split` повертає чотири окремі масиви: `X_train`, `X_test`, `y_train` і `y_test`. Масиви даних `X_train` і `y_train` представляють безпосередньо дані навчання, а `X_test` і `y_test` – відповідно набір даних для тестування.

4. Навчання моделі поліноміальної лінійної регресії на навчальному наборі.

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
poly_reg = PolynomialFeatures(degree= 4 )
X_poly = poly_reg.fit_transform(X_train)
regressor = LinearRegression()
regressor.fit(X_poly,y_train)
```

Рисунок 2.8. – Навчання моделі на навчальному наборі

На цьому етапі необхідно імпортувати модулі `PolynomialFeatures` та `LinearRegression` та створити екземпляр класу `PolynomialFeatures`, призначивши його `poly_reg`. Значення параметра ступеня (4) вказує, до якого ступеня будуть генерувати поліноміальні об'єкти. Метод `fit_transform()` адаптує `PolynomialFeatures` до навчальних даних і перетворює вихідні функції на поліноміальні. Пізніше відбувається тренування моделі лінійної регресії на поліноміальних ознаках.

5. Прогнозування результатів тестового набору.

```
y_pred = regressor.predict(poly_reg.transform(X_test))
np.set_printoptions(precision= 2 )
print (y_pred)
```

Рисунок 2.9. – Рядок коду для прогнозування

В даному випадку використано метод `predict()` навченої моделі для створення прогнозів для тестових даних `X_test`, який приймає вхідні функції (`X_test`) як аргумент і повертає прогнозовані значення для цільової змінної.

б. *Оцінка продуктивності моделі.* Із модуля метрик `scikit-learn` імпортовано функцію `r2_score` для оцінки регресійних моделей (зокрема частку дисперсії цільової змінної). Якщо показник R-квадрат буде наближатися до 1, це означатиме ідеальну відповідність.

```
from sklearn.metrics import r2_score
r2_score(y_pred, y_test)
```

Рисунок 2.10 – Оцінка продуктивності моделі

Після вибору та навчання моделі машинного навчання доцільно обирати технологічні рішення для розгортання та обслуговування моделі.

Для прикладу використання BentoML, який є уніфікованою структурою додатків штучного інтелекту, дає змогу полегшити та пришвидшити доставку програм машинного навчання, які характеризуватимуться надійністю, маштабованістю та економічною ефективністю.

```
from __future__ import annotations
import bentoml
from transformers import pipeline

@bentoml.service(
    resources={"cpu": "2"},
    traffic={"timeout": 10},
)
class Summarization:
    def __init__(self) -> None:
        # Load model into pipeline
        self.pipeline = pipeline('summarization')

    @bentoml.api
    def summarize(self, text: str) -> str:
        result = self.pipeline(text)
        return result[0]['summary_text']
```

Рисунок 2.11 – Визначення логіки обслуговування моделі у `service.py` файлі сервісу BentoML

BentoML дозволяє імпортувати або створювати власні моделі на основі фреймворків PyTorch, TensorFlow, Keras, Scikit-Learn, XGBoost та багатьох інших. Він також містить підтримує паралельну обробку висновків моделі для підвищення швидкості та ефективності, дозволяє застосовувати адаптивне

пакування для оптимізації обробки та вбудовану оптимізація для конкретних архітектур моделей (наприклад, OpenLLM для LLM) [46].

BentoML фактично є першим інструментом Python для розгортання та підтримки API у виробництві, який масштабується за допомогою потужних оптимізацій за допомогою паралельного висновку та адаптивного пакування та дозволяє забезпечити апаратне прискорення.

2.3. Визначення перспектив застосування моделей генеративного штучного інтелекту у формі цифрових помічників

В ході виконання кваліфікаційної роботи було з'ясовано, що одне з бізнес-завдань, яке можуть бути вирішені на основі використання моделей машинного навчання та глибинних нейронних мереж у комерційній діяльності, є створення та використанням чат-ботів та інтелектуальних (цифрових) помічників у комерційній діяльності.

Застосування цифрових помічників не є новою практикою в управлінні та автоматизації бізнес-процесів. Підтвердження цього знаходимо в різноманітних наукових дослідженнях та практичних кейсах. Наприклад у 2021 році як інноваційні рішення уже використовувалися в нафтогазовій, енергетичній, хімічній промисловості та будівництві [43].

Наприкінці 2021 р. телекомунікаційний гігант KDDI уклав партнерство з фірмою Mawaгі на створення віртуального помічника типу «Цифрова людина як послуга» (*Digital human as a service (DHaaS)*), розуміючи під цим інтерфейс технологій штучного інтелекту, який фізично виглядає, звучить і поводить себе як люди в реальному житті [44]. Технологія DHaaS може використовуватися не лише як сервісне програмне забезпечення і персональний посібник в смартфонах, але через ефект візуалізації та віртуальної присутності може бути досить ефективною у сфері обслуговування клієнтів та підтримки бренду в маркетингових процесах.

В основі DHaas закладені можливості обробки природної мови (NLP), які були розкриті в даній кваліфікаційній роботі, та алгоритмів і моделей машинного навчання (ML). Поєднання стеку технологій разом із використанням хмарних інтелектуальних 3D-аватар дозволило розробникам цифрового асистента покращити взаємодією штучного інтелекту та людини, забезпечити більш природній (людський) досвід обслуговування на основі цих технологічних рішень.

Проаналізувавши найбільш поширені світові практики застосування генеративного штучного інтелекту в сільському господарстві [45], можна відмітити різноманіття їх форм та напрямків застосування для аграрних виробників (таблиця 2.7).

Таблиця 2.7. – Форми цифрових помічників для аграрної сфери

Тип	Приклад реалізації
Штучний інтелект для навчання	Віртуальний помічник у формі консультаційної службою для фермерів на основі штучного інтелекту Farmer.CHAT. Може використовуватися для оптимізації процесів у рослинництві, управляти ресурсами та показниками виробничої ефективності (наприклад урожайністю культур).
Штучний інтелект для підтримки рішень та досліджень	Semantic Scholar на основі штучного інтелекту для забезпечення швидкого доступу до знань в аграрній сфері.
Штучний інтелект для власників аграрного бізнесу	Автоматизоване рішення зі штучним інтелектом Yooz для інтелектуальної підтримки управління бізнес-процесами. Дозволяє керувати рахунками-фактурами та іншою документацією.

З лютого 2023 року для широкого використання стали доступні технологічні рішення на основі потужних мовних моделей генеративного штучного інтелекту (ChatGPT, Bard та ін), ознаменували нову епоху використання нейронних мереж та штучного інтелекту в різноманітних сферах.

РОЗДІЛ 3. РЕЗУЛЬТАТИ ВИРІШЕННЯ ЗАДАЧІ

3.1. Підготовка та аналіз даних для прогнозної моделі машинного навчання інтелектуальної інформаційної системи

Вибір алгоритму машинного навчання та побудова прогнозної моделі на його основі, як було встановлено на попередніх етапах дослідження, залежить від кількох чинників:

- 1) розміру, якості та характеру даних;
- 2) доступності часу обчислення та терміновості завдання;
- 3) мети та цілей дослідження і подальшого використання прогнозних даних.

Підготовка та аналіз даних безумовно відіграють одну з найважливіших ролей у розробці моделей машинного навчання. Підготовка даних часто називається попередньою обробкою даних, оскільки цей етап передбачає роботу із зібраними необробленими даними та підготовка їх до передачі алгоритмам машинного навчання (чистка, масштабування та вибір релевантних ознак). Фактично цей процес дозволяє забезпечити якісні вхідні дані для навчання моделі та позитивно вплинути на ефективність моделі машинного навчання у подальших прогнозах чи класифікаціях. Адекватна обробка даних допомагає зменшити шум, підвищує точність та надійність моделі, забезпечуючи більш точні та значущі результати.

У загальному підготовку та аналізу даних для моделей машинного навчання рекомендують здійснювати в кілька етапів [48]: формулювання проблеми; збір даних (витяг, завантаження та перетворення); категоризацію даних (класифікація, кластеризація, регресія, рейтинг); попередню обробку даних (перевірка якості даних, форматування даних, зменшення та очищення

якості даних); перетворення даних (масштабування, розкладання, агрегація); розробка та вибір функції; розподілення даних в машинному навчанні.

На етапі збору даних може бути реалізований один з двох варіантів:

1) Витяг, завантаження та перетворення (*Extract, Load, Transform (ELT)*). В цьому випадку неструктуровані дані перетворюються всередині моделі перед обробкою.

2) Витяг, перетворення та завантаження (*Extract, Transform, Load (ETL)*). Дані витягуються, а потім готуються для доставки в модель перед тим, як перейти до завантаження (часто з використанням проміжної області).

Джерелами даних для моделі машинного навчання інтелектуальної інформаційної системи планування можуть бути бази даних, електронні таблиці, офіційні статистичні показники, веб-ресурси або API.

Таблиця 3.1 – Автоматизовані методи збору даних

Методи збору даних автоматизації на основі ШІ	Методи збору даних низькорівневої автоматизації	Методи збору даних автоматизації на основі IoT
<ul style="list-style-type: none"> • Оптичне розпізнавання символів (OCR) • Інтелектуальне розпізнавання символів (ICR) • Оптичне розпізнавання позначок (OMR) • Інтелектуальна обробка документів (IDP) • Обробка природньої мови (NLP) • Розпізнавання мови • Видобуток даних 	<ul style="list-style-type: none"> • Запит до бази даних • QA-код і розпізнавання штрих-коду • Веб-збирання • API 	<ul style="list-style-type: none"> • Датчики

Побудовано на основі даних [50]

На фоні активного використання технологій Інтернету речей (IoT), зокрема в аграрній сфері, за допомогою датчиків можна автоматично отримувати різні типи даних.

Отримані дані обробляються на невідповідності або виправлення відсутніх значень, нормалізуються та кодуються, зменшується їх складність та наприклад розмірність.

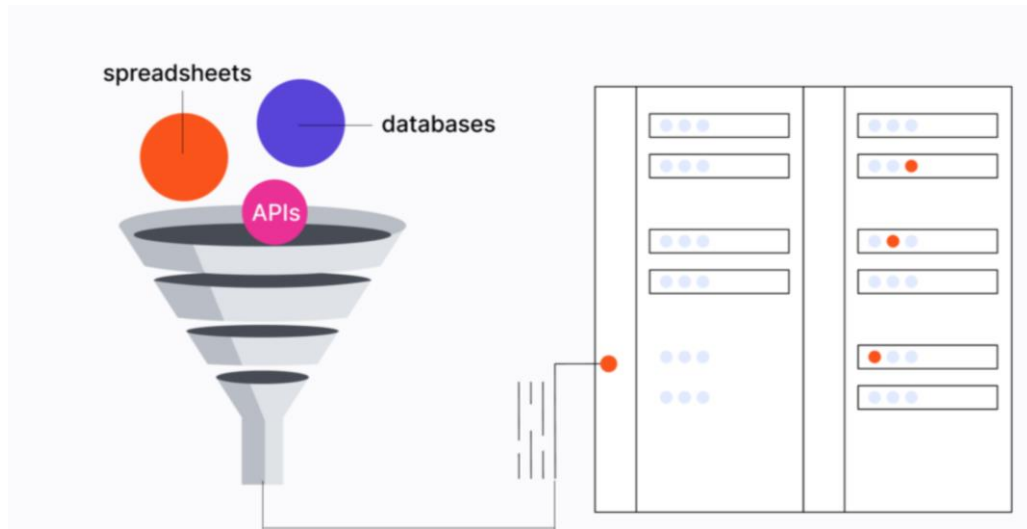


Рисунок 3.1. – Етап збору даних для машинного навчання [49]

Для моделей машинного навчання загалом використовують три типи даних [51]:

- 1) Структуровані дані, які організовані переважно у форматі електронної таблиці. Часто структуровані дані формуються на основі інформації, зібраної з баз даних або транзакційних систем.
- 2) Неструктуровані дані, які невідповідають звичайним моделям даних (зображення, відео, аудіозаписи тощо).
- 3) Напівструктуровані дані місять деякі структурні елементи, проте в загальному не відповідають формату табличної моделі даних (теги або метадані, дані у форматах XML або JSON).

Оскільки не у всіх випадках дані зібрані повністю або належної якості, брак точок даних може бути компенсований деякими техніками. По-перше, це розширення даних з метою генерації більших даних з існуючих зразків (на основі методик обертання, переміщення або масштабування). По-друге, активним навчанням, у т.ч. із залученням фахівця-експерта. По-третє, трансферним навчанням на основі використання попередньо навчених

алгоритмів ML. По-четверте, спільним обміном даними з іншими експертами, дослідниками чи установами (компаніями), які спеціалізуються або ж використовують моделі машинного навчання.

Для вирішення поставленої задачі у ході виконання кваліфікаційної роботи, а саме обґрунтування та побудова моделі машинного навчання інтелектуальної інформаційної системи планування процесами в сільському господарстві на прикладі планування вирощування озимої пшениці був проведений аналіз та збір даних (таблиця 3.2).

Таблиця 3.2. – Основні характеристики набору даних для побудови моделі

Назва даних	Позначення	Опис або обґрунтування вибору
Азот	N	Азот значною мірою відповідає за ріст листя на рослині
Фосфор	P	Фосфор в значній мірі відповідає за ріст коренів та розвиток квітів і плодів
Калій	K	Калій є поживною речовиною, яка сприяє правильному виконанню загальних функцій рослини
Температура	temperature	Температура в градусах Цельсія
Вологість	humidity	Відносна вологість у відсотках
Кислотність	ph	Значення ph ґрунту
Кількість опадів	rainfall	Кількість опадів у мм
Урожайність	harvest	Урожайність пшениц, т/га (визначена)

Зазначений набір даних передбачає використання 7 вхідних змінних ($X_1...X_7$) та відповідно 1 вихідну змінну (Y_1).

Для виконання поставленого завдання у кваліфікаційній роботі було використано описаний в другому розділі інтерактивне середовище Jupyter

Notebook, що дозволило легко візуалізувати дані, проводити експерименти та налагоджувати моделі машинного навчання.

Початковим етапом роботи є підключення бібліотек Python в середовище блокнота Jupyter, з яким ми будемо працювати (рис.3.1).

Importing Packages

```
In [1]: import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
sns.set_style('darkgrid')
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
import statsmodels.api as sm
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
from statsmodels.stats.outliers_influence import variance_inflation_factor
import xgboost as xgb
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
```

Рисунок 3.1. - Імпорт бібліотек Python в середовище блокнота Jupyter.

Після імпорту необхідних бібліотек було реалізовано наступний етап – завантаження та виведення початкового масиву даних (рис.3.2). Дані імпортуються із файлу з CSV-файлу в DataFrame та зберігаються в табличному форматі. Використання DataFrame пояснюється тим, що це основна структура даних у бібліотеці Pandas, яка використовується для зручної обробки та аналізу даних.

В результаті виконання очікувалося виведення на екран перших п'яти рядків даних для огляду, що фактично і було реалізовано під час виконання задачі.

Read Dataset

```
In [2]: df = pd.read_csv("F:\Dipl_2023\Dipl_mag_126\ЖАМ\Data_1.csv")
df.head(5)
```

Out[2]:

	N	P	K	temperature	humidity	ph	rainfall	harvest
0	112.454037	42.213853	125.919733	20.3	82.0	6.8	673.3	51.6
1	96.527968	30.913770	124.239416	19.6	80.3	7.3	609.5	24.6
2	89.529466	43.770578	172.147264	20.0	82.3	6.5	663.8	43.2
3	86.088393	43.762988	135.907373	16.5	80.2	6.8	600.3	19.4
4	107.637741	41.808115	149.138370	17.5	83.4	7.2	414.3	27.5

Рисунок 3.2 – Завантаження набору даних у середовище Jupyter

Як видно із рисунку 3.2, на цьому етапі реалізації дослідження відбулося дві операції:

1) Використано бібліотеку Pandas для читання даних з CSV-файлу за допомогою коду `pd.read_csv("F:\Dipl_2023\Dipl_mag_126\ЖАМ\Data_1.csv")`. Файл із даними ("F:\Dipl_2023\Dipl_mag_126\ЖАМ\Data_1.csv") був завантажений у вигляді DataFrame в змінну `df`.

2) Застосований метод `df.head(5)`, який вивів перші п'ять рядків з DataFrame `df`. Даний метод використовується з метою швидкого перегляду структури та вмісту даних у завантаженому наборі.

```

Data Analysis

In [5]: df.harvest.value_counts()

Out[5]: harvest
18.0    11
28.3    11
26.1    10
30.6    10
20.1     9
..
32.9     1
53.5     1
56.7     1
42.5     1
43.1     1
Name: count, Length: 445, dtype: int64

In [6]: df.shape

Out[6]: (1516, 8)

In [7]: # To check whether there null values
df.apply(lambda x: len(x.isnull()))

Out[7]: N          1516
P          1516
K          1516
temperature 1516
humidity    1516
ph          1516
rainfall    1516
harvest     1516
dtype: int64

```

Рисунок 3.3.- Реалізація операцій із даними DataFrame `df`

Наступний етап роботи із завантаженими давними DataFrame `df` у середовищі Jupyter передбачав виконання таких трьох послідовних операцій:

1) підрахунок унікальних значень у стовпці "harvest" та виведення їх загальної кількості на основі виконання команди `df.harvest.value_counts()`. Дана операція може бути корисною для отримання розподілу значень за даним показником та допомогти в розумінні характеристик даних.

2) `df.shape` – виклик фактично повертає кортеж, що містить розмірність DataFrame, яка відповідає кількості рядків та стовпців у `df`.

3) виклик функції *apply* для кожного стовпця в DataFrame (при цьому використовується анонімна лямбда-функцію) для підрахунку кількості пропущених значень у кожному стовпці за допомогою *df.apply(lambda x: len(x.isnull()))*. В результаті виконання коду визначено кількість пропущених значень у кожному стовпці.

```
In [13]: df.describe()
```

```
Out[13]:
```

	N	P	K	temperature	humidity	ph	rainfall	harvest
count	1516.000000	1516.000000	1516.000000	1516.000000	1516.000000	1516.000000	1516.000000	1516.000000
mean	110.994377	33.067944	174.229833	18.051385	77.085026	7.037797	556.217876	32.589842
std	16.431427	9.521133	31.317429	2.227738	16.243847	0.434551	84.583246	12.817795
min	80.105438	15.133802	120.026406	14.000000	14.700000	6.300000	400.100000	15.000000
25%	97.043035	25.342854	146.813145	16.200000	65.175000	6.700000	483.450000	22.300000
50%	111.133760	32.946560	173.921220	18.100000	82.100000	7.000000	560.250000	29.700000
75%	124.247634	41.059207	200.382101	19.900000	90.800000	7.400000	627.775000	40.800000
max	139.938526	49.989670	229.773893	22.000000	100.000000	7.800000	699.800000	64.700000

Рисунок 3.4. -Опис даних, сформований методом *df.describe()*

На рисунку 3.4 показано виведений опис даних, який містить таку інформацію:

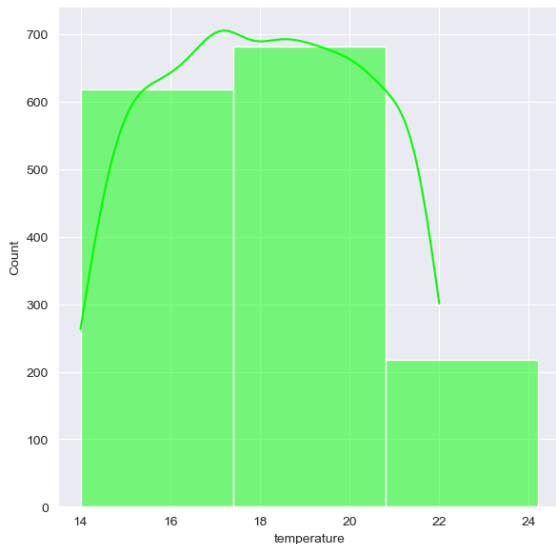
- *count* - кількість спостережень у кожній колонці.
- *mean* - середнє значення в кожній колонці.
- *std* - стандартне відхилення в кожній колонці.
- *min*- мінімальне значення в кожній колонці.
- *25%* - 25-й перцентиль в кожній колонці.
- *50%* - 50-й перцентиль в кожній колонці (медальйон).
- *75%* - 75-й перцентиль в кожній колонці.
- *max* - максимальне значення в кожній колонці.

```
In [14]: df.info()
```

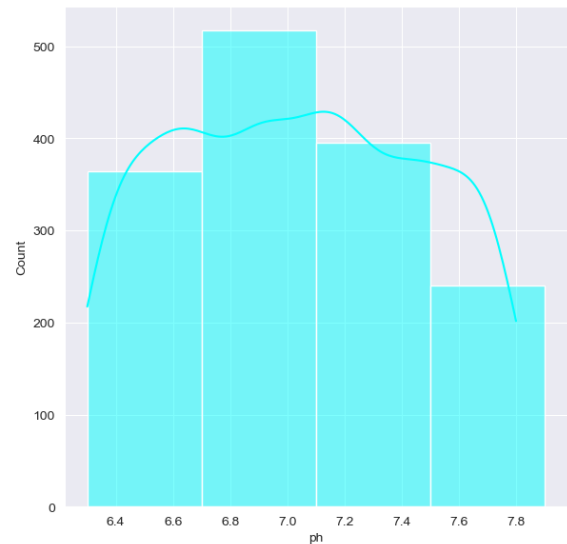
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1516 entries, 0 to 1515
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   N                1516 non-null   float64
1   P                1516 non-null   float64
2   K                1516 non-null   float64
3   temperature      1516 non-null   float64
4   humidity         1516 non-null   float64
5   ph               1516 non-null   float64
6   rainfall         1516 non-null   float64
7   harvest          1516 non-null   float64
dtypes: float64(8)
memory usage: 94.9 KB
```

Рисунок 3.5. – Узагальнена інформація по набору даних на основі *df.info()*

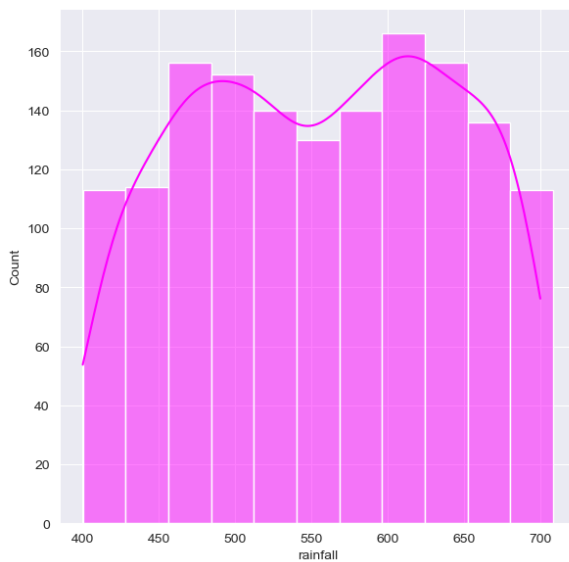
Представлена інформація може бути корисною для розуміння характеристик набору даних, їх якості та повноти. Проаналізовані таким чином вхідні дані для навчання моделі (наприклад на предмет наявності аномалій або пропущених даних) можуть давати розуміння, наскільки ефективною може бути модель машинного навчання у подальших прогнозах чи класифікаціях.



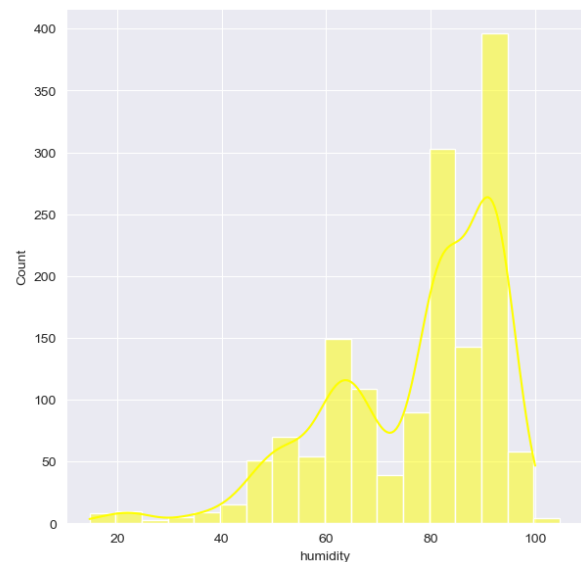
а)



б)



в)



г)

Рисунок 3.6. - Розподіл значень даних по таких показниках як температура (а), кислотність ґрунту (б), кількість опадів (в), вологість (г).

```
In [17]: #Checking the Statistics for all the df
print("Average Ratio of nitrogen in the soil : {0: .2f}".format(df['N'].mean()))
print("Average Ratio of Phosphorous in the soil : {0: .2f}".format(df['P'].mean()))
print("Average Ratio of Potassium in the soil : {0: .2f}".format(df['K'].mean()))
print("Average temperature in Celsius : {0: .2f}".format(df['temperature'].mean()))
print("Average Relative Humidity in % is : {0: .2f}".format(df['humidity'].mean()))
print("Average pH value of the soil : {0: .2f}".format(df['ph'].mean()))
print("Average Rain fall in mm : {0: .2f}".format(df['rainfall'].mean()))

Average Ratio of nitrogen in the soil : 110.99
Average Ratio of Phosphorous in the soil : 33.07
Average Ratio of Potassium in the soil : 174.23
Average temperature in Celsius : 18.05
Average Relative Humidity in % is : 77.09
Average pH value of the soil : 7.04
Average Rain fall in mm : 556.22
```

Рисунок 3.7 – Оцінка статистичних характеристик для змінних набору даних DataFrame df

На рисунку 3.7. представлено результати оцінки статистичних характеристик на основі DataFrame df. Зокрема, виконані такі операції:

1) `print("Average Ratio of nitrogen in the soil: {0: .2f}".format(df['N'].mean()))` - визначено та виведено середнє значення концентрації азоту у ґрунті.

2) `print("Average Ratio of Phosphorous in the soil: {0: .2f}".format(df['P'].mean()))` – визначено та виведено середнє значення концентрації фосфору у ґрунті.

3) `print("Average Ratio of Potassium in the soil: {0: .2f}".format(df['K'].mean()))` - визначено та виведено середнє значення концентрації калію у ґрунті.

4) `print("Average temperature in Celsius: {0: .2f}".format(df['temperature'].mean()))` - визначено та виведено середнє значення температури у градусах Цельсія.

5) `print("Average Relative Humidity in % is: {0: .2f}".format(df['humidity'].mean()))` - визначено та виведено середнє значення відносної вологості у відсотках.

6) `print("Average pH value of the soil: {0: .2f}".format(df['ph'].mean()))` - визначено та виведено середнє значення рН у ґрунті.

7) `print("Average Rain fall in mm: {0: .2f}".format(df['rainfall'].mean()))` - визначено та виведено середнє значення кількості опадів у міліметрах.

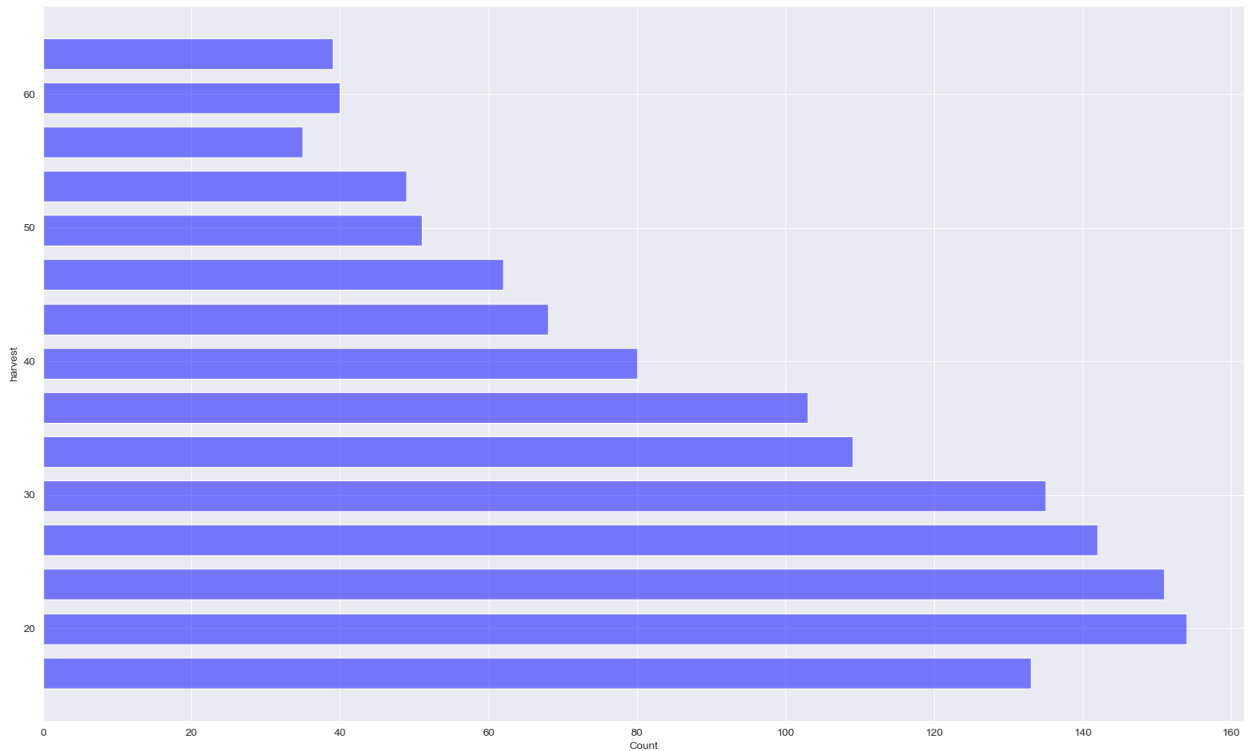


Рисунок 3.8 – Розподіл урожайності по дням з температурним режимом в діапазоні від 15 до 28 градусів Цельсія

Оскільки для досягання урожаю пшениці велике значення має температурний режим влітку, було здійснено аналіз розподілу показника урожайності озимої пшениці для тих днів, коли температура знаходилась в діапазоні від 15 до 28 градусів Цельсія. Результати дослідження представлені у формі гістограми на рисунку 3.8, де по вертикалі відображено значення параметра "harvest" (урожайність).

В додатку Д у формі рисунків наведені інші результати оцінки даних на етапі їх підготовки та аналізу щодо якісних характеристик.

Виконання описаного в даному підпитанні коду загалом дозволяє отримати уявлення та загальну інформацію щодо статистики даних у наборі DataFrame. Отримані результати дослідження свідчать про те, що набір даних є якісним та може використовуватися для побудови моделі машинного навчання на наступному етапі виконання кваліфікаційної роботи.

3.2. Побудова та вибір прогновної моделі машинного навчання інтелектуальної інформаційної системи

Побудова та вибір прогновної моделі в машинному навчанні для інтелектуальної інформаційної систем основним етапом практичної реалізації завдань, поставлених для виконання у кваліфікаційній роботі.

На основі попередньо завантажених в інтерактивне середовище Jupyter Notebook, оброблених та проаналізованих характеристик даних, можна визначити ключові параметри та обирати оптимальні алгоритми навчання для досягнення найвищої точності та ефективності моделі. Подальший відбір та налаштування гіперпараметрів дозволить оптимізувати модель для поставленого завдання, забезпечуючи найкращі результати прогнозування в рамках інтелектуальної інформаційної системи. На попередніх етапах дослідження в ході виконання кваліфікаційної роботи було встановлено доцільність вибору наступних алгоритмів для навчання моделі прогнозування урожайності озимої пшениці:

- ✓ найменших квадратів (OLS);
- ✓ градієнтного бустингу (XGBoost);
- ✓ лінійної регресії з поліноміальними ознаками.

```
In [34]: #RFE
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

```
In [35]: from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression

lm = LinearRegression()
lm.fit(X_train, y_train)

# Specify the number of features to select (e.g., 10 in this case)
num_features_to_select = 10

rfe = RFE(lm, n_features_to_select=num_features_to_select)
rfe = rfe.fit(X_train, y_train)
```

```
In [36]: list(zip(X_train.columns, rfe.support_, rfe.ranking_))
```

```
Out[36]: [('N', True, 1),
          ('P', True, 1),
          ('K', True, 1),
          ('temperature', True, 1),
          ('humidity', True, 1),
          ('rainfall', True, 1)]
```

```
In [37]: X_train.columns[rfe.support_]
```

```
Out[37]: Index(['N', 'P', 'K', 'temperature', 'humidity', 'rainfall'], dtype='object')
```

Рисунок 3.9. – Внесення параметрів для прогновної моделі

```
In [39]: def build_model(X,y):
        X = sm.add_constant(X) #Adding the constant
        lm = sm.OLS(y,X).fit() # fitting the model
        print(lm.summary()) # model summary
        return X

def checkVIF(X):
    vif = pd.DataFrame()
    vif['Features'] = X.columns
    vif['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
    vif['VIF'] = round(vif['VIF'], 2)
    vif = vif.sort_values(by = "VIF", ascending = False)
    return(vif)
```

Рисунок 3.10 - Побудова прогновної моделі в середовищі Jupyter Notebook

Наступним етапом дослідження є навчання прогновної моделі з використанням різних типів алгоритмів машинного навчання.

Для початку побудуємо модель на основі алгоритму найменших квадратів. Побудова моделі на основі цього алгоритму є досить популярним підходом у задачах регресії, оскільки дозволяє знайти лінійну функцію, яка найкращим чином відображає залежність між незалежними та залежними змінними, мінімізуючи суму квадратів відхилень. Після визначення коефіцієнтів моделі, вона може бути використана для прогнозування значень залежної змінної на основі нових вхідних даних. Використання алгоритму найменших квадратів забезпечує ефективний і точний спосіб побудови регресійних моделей.

```
In [40]: X_train_new = build_model(X_train_rfe,y_train)
```

OLS Regression Results						
Dep. Variable:	harvest	R-squared:	0.905			
Model:	OLS	Adj. R-squared:	0.904			
Method:	Least Squares	F-statistic:	1669.			
Date:	Fri, 05 Jan 2024	Prob (F-statistic):	0.00			
Time:	19:11:51	Log-Likelihood:	1185.9			
No. Observations:	1061	AIC:	-2358.			
Df Residuals:	1054	BIC:	-2323.			
Df Model:	6					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	-1.6650	0.021	-79.315	0.000	-1.706	-1.624
N	0.6718	0.010	68.584	0.000	0.653	0.691
P	0.7083	0.010	71.759	0.000	0.689	0.728
K	0.4067	0.009	45.954	0.000	0.389	0.424
temperature	0.5614	0.009	59.696	0.000	0.543	0.580
humidity	0.8719	0.014	61.752	0.000	0.844	0.900
rainfall	0.3492	0.009	39.037	0.000	0.332	0.367
Omnibus:	46.067	Durbin-Watson:	2.067			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	103.425			
Skew:	-0.239	Prob(JB):	3.48e-23			
Kurtosis:	4.453	Cond. No.	18.2			

Рисунок 3.11 – Модель звичайних найменших квадратів (OLS)

Структура коду моделі на основі алгоритму найменших квадратів подано на рисунку 3.12.

```

In [41]: import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
from statsmodels.stats.outliers_influence import variance_inflation_factor

# ... (your existing code for feature selection and model building)

def plot_results(y_true, y_pred, title='Model Performance'):
    """
    Display a plot of true values vs. predicted values and print accuracy metrics.

    Parameters:
    - y_true: array-like, true values
    - y_pred: array-like, predicted values
    - title: str, title of the plot (default='Model Performance')
    """
    # Plotting true values vs. predicted values
    plt.scatter(y_true, y_pred)
    plt.title(title)
    plt.xlabel('True Values')
    plt.ylabel('Predictions')
    plt.show()

    # Calculate accuracy metrics
    mse = np.mean((y_true - y_pred) ** 2)
    rmse = np.sqrt(mse)
    mae = np.mean(np.abs(y_true - y_pred))
    r2 = 1 - (np.sum((y_true - y_pred) ** 2) / np.sum((y_true - np.mean(y_true)) ** 2))

    # Print accuracy metrics
    print(f'Mean Squared Error (MSE): {mse:.4f}')
    print(f'Root Mean Squared Error (RMSE): {rmse:.4f}')
    print(f'Mean Absolute Error (MAE): {mae:.4f}')
    print(f'R-squared (R2): {r2:.4f}')

    # Fit the model and make predictions
    lm = LinearRegression()
    lm.fit(X_train_rfe, y_train)
    y_pred_train = lm.predict(X_train_rfe)

    # Plot and print results
    plot_results(y_train, y_pred_train, title='Training Performance')

```

Рисунок 3.12 – Код побудови та навчання моделі на основі алгоритму найменших квадратів

Побудова моделі на основі алгоритму градієнтного бустингу, зокрема XGBoost, включає кілька ключових етапів. На основі проведеного вище аналізу та підготовки даних, визначаються параметри моделі, такі як глибина дерев, швидкість навчання та кількість дерев.

Після цього розпочинається сам процес навчання моделі, де використовується градієнтний спуск для підвищення якості прогнозів. За допомогою ансамблю дерев рішень, XGBoost вдосконалюється за кілька ітерацій, надаючи вагу помилкам та виправляючи їх на кожному кроці.

Остаточний етап включає налаштування гіперпараметрів, щоб забезпечити оптимальну продуктивність моделі.

```
In [42]: import xgboost as xgb

# Fit XGBoost model
model_xgboost = xgb.XGBRegressor()
model_xgboost.fit(X_train_rfe, y_train)

# Зберегти навчену модель у файл
model_path = r'F:\Dipl_2023\Dipl_mag_126\ЖАМ\trained_xgboost_model.model'
model_xgboost.save_model(model_path)

# Make predictions
y_pred_xgboost = model_xgboost.predict(X_train_rfe)

# Plot and print results
plot_results(y_train, y_pred_xgboost, title='XGBoost Training Performance')
```

Рисунок 3.13 – Побудова моделі на основі алгоритму градієнтного бустингу (XGBoost)

Навчання прогнозної моделі на основі градієнтного бустингу XGBoost - це ітеративний процес, який використовує ансамбль слабких моделей (зазвичай дерев рішень) для покращення прогнозів.

Основна ідея полягає в тому, щоб на кожному кроці підкоригувати модель так, щоб вона виправляла помилки попередньої ітерації.

Етапи навчання цього типу моделі можна описати таким чином:

1. Спочатку створюється початкова модель, яка може бути дуже простою.
2. Обчислюється градієнт помилок та розраховується градієнту функції втрат відносно прогнозів для кожного прикладу в навчальному наборі.
3. Додається нова модель (зазвичай дерево рішень), яка намагається виправити помилки, обчислені на попередньому кроці. Ця модель додається до ансамблю, враховуючи коефіцієнт навчання, який контролює внесок нової моделі.
4. Прогнози моделі оновлюються шляхом додавання до них прогнозів нової моделі з урахуванням коефіцієнта навчання.
5. Цей процес повторюється, додаючи нові моделі, кожна з яких намагається виправити залишкові помилки ансамблю.

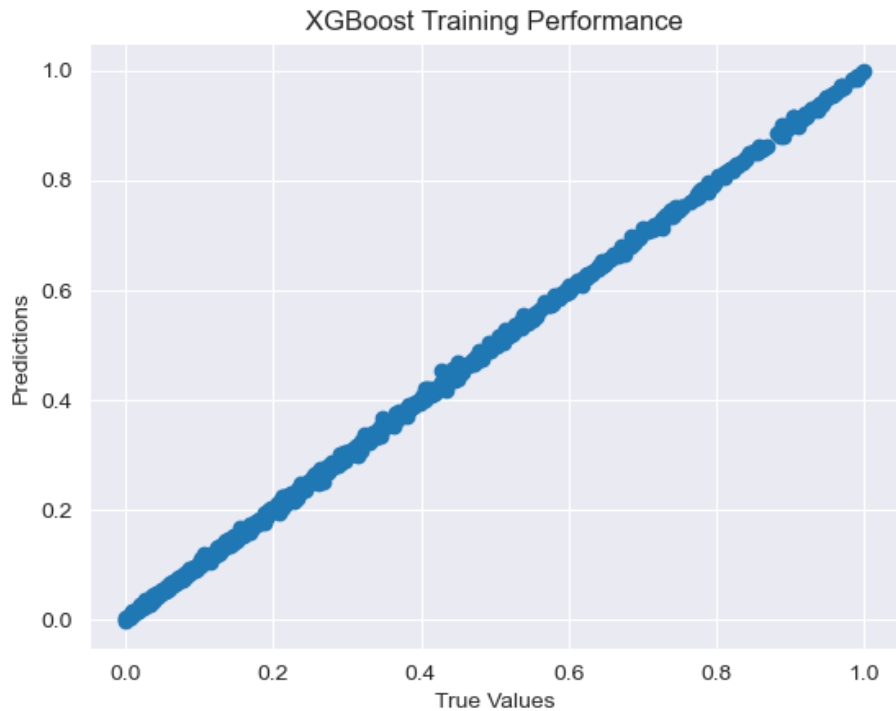


Рисунок 3.14 – Візуалізація результатів навчання на основі градієнтного бустингу XGBoost

Було також проведено побудову та навчання прогнозувальної моделі на основі алгоритму лінійної регресії з поліноміальними ознаками.

```
In [43]: from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline

# Create polynomial features
degree = 2 # You can adjust the degree as needed
polyreg = make_pipeline(PolynomialFeatures(degree), LinearRegression())

# Fit the model
polyreg.fit(X_train_rfe, y_train)

# Make predictions
y_pred_polyreg = polyreg.predict(X_train_rfe)

# Plot and print results
plot_results(y_train, y_pred_polyreg, title=f'PolyReg Training Performance (Degree {degree})')
```

Рисунок 3.15 - Побудова моделі на основі алгоритму лінійної регресії з поліноміальними ознаками

На початку розглядаються оригінальні ознаки даних, а потім генеруються поліноміальні ознаки, підняті в ступені для отримання нелінійних залежностей. Дана модель навчається за допомогою тренувального набору, де

вона вирішує оптимальні значення ваг для кожної ознаки, мінімізуючи функцію втрат.

Прогнозна модель на основі алгоритму лінійної регресії з поліноміальними ознаками оцінюється на тестовому наборі для оцінки її прогностичної здатності та уникнення перенавчання. Цей підхід дозволяє лінійній регресії адаптуватися до нелінійних закономірностей у даних, що робить його ефективним для моделювання складних залежностей.

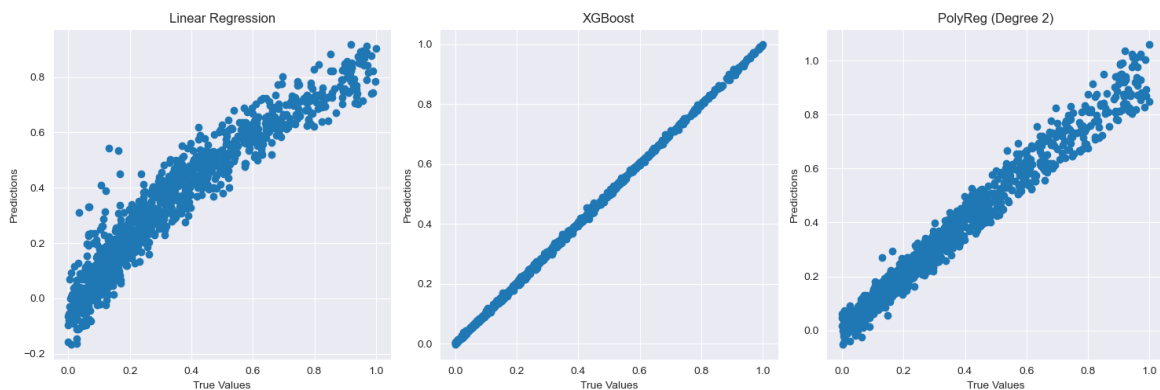


Рисунок 3.16 – Порівняльне візуальне представлення результатів навчання трьох типів прогнозних моделей

З метою вибору оптимальної моделі для побудови прогнозів було здійснено оцінку якості моделей на основі метрик, розглянутих в попередніх розділах кваліфікаційної роботи. Метрика MSE вимірює середнє значення квадратів різниць між спостережуваними та передбаченими значеннями. RMSE є квадратним коренем з MSE і вимірює середньоквадратичну помилку. MAE вимірює середнє абсолютне значення різниць між спостережуваними та передбаченими значеннями. R-squared вказує, яка частина змінності в пояснюваній змінній пояснюється моделлю.

За отриманими даними можна сказати, що найкращою моделлю є модель градієнтного бустингу (XGBoost). Вона має найнижчі значення всіх метрик якості, включаючи MSE, RMSE, MAE та R-squared. Це означає, що вона робить найточніші прогнози значень залежної змінної, а саме показника урожайності озимої пшениці.

Comparison of Model Performance Metrics



Рисунок 3.17 – Результати застосування метрик оцінки якості побудованих прогнозних моделей

На основі обраної прогнозної моделі можна створювати інтелектуальну інформаційну систему планування процесів в сільському господарстві з модулем для планування збору врожаю озимої пшениці за відомої прогнозованої урожайності, що включає в себе:

1.Визначення очікуваного обсягу валового збору сільськогосподарської культури шляхом використання прогнозованої урожайності та планових посівних площ.

2.Планувати потребу у сільськогосподарській техніці та працівниках, залучених до технологічних процесів в рослинництві.

3.Планування маршрутів і графіку збирання врожаю. Це дасть змогу забезпечити ефективно і своєчасне збирання врожаю та знизити його втрати.

4.Планування зберігання врожаю, очищення зібраної пшениці. Це необхідно для того, щоб зберегти урожай в хорошому стані до його реалізації на аграрному ринку.

3.3. Розробка цифрових асистентів для управління процесами прийняття рішень

Розвиток моделей машинного навчання, інформаційних технологій, мов програмування, технік збору даних призвів до появи різних форм та напрямків застосування штучного інтелекту, серед яких широкого застосування останнім часом набули моделі генеративного штучного інтелекту у формі відомих користувачам асистентів ChatGPT, Gemini (Bard), Microsoft Copilot, GitHub Copilot, Claude та інших.

Загалом під генеративного ШІ (genAI) широке позначення для опису будь-якого тип штучного інтелекту (ШІ), який може створювати новий текст, таблиці, програмний код, зображення, відео чи аудіокліпи. З технічної точки зору цей тип штучного інтелекту вивчає шаблони з навчальних даних і генерує нові, унікальні результати з такими ж статистичними властивостями [52].

Асистенти на основі генеративного штучного інтелекту мають різні моделі, що обумовлює певні відмінності у результатах виконання запитів користувачів.

Генеративні моделі штучного інтелекту використовують підказки, щоб спрямовувати процес створення вмісту, і використовують перехідне навчання, щоб стати більш досвідченим [52]. Для прикладу користувач може створити персонального цифрового помічника на основі моделі ChatGPT або Microsoft Copilot та надавати йому інструкції та додаткові матеріали для генерації відповідей.

В основі навчання великих мовних моделей LLM було закладено принципи оперативного навчання, що становить собою стратегію повторного використання тієї самої моделі для різних завдань без повторного навчання. Для прикладу асистенти на базі моделей GPT-4 і (Gemini) BERT, вимагають попереднього навчання моделі з даними без міток, а потім її тонкого налаштування для конкретних завдань із даними з мітками.

Взявши за основу найбільш поширену модель ChatGPT, виділимо її основні можливості: розширений аналіз даних (*Advanced Data Analysis*), персональний ChatGPT (*My GPTs*), API помічника (*Assistant API*), застосування плагінів (наприклад комбінацію ChatGPT+ Wolfram).

За даними офіційного ресурсу (<https://www.wolfram.com/wolfram-plugin-chatgpt/>) Wolfram надає моделі ChatGPT доступ до потужних обчислень, точної математики, підібраних знань, даних у реальному часі та візуалізації через Wolfram|Alpha та Wolfram Language. Ці нові можливості цифрового асистента розширюють його застосування не лише в персональних цілях, але і як складової інтелектуальної системи для прийняття рішень, покращення та пришвидшення обробки даних, які можуть бути завантажені для аналізу у формі наприклад електронних таблиць.

Фактично плагін Wolfram для ChatGPT додає деякі додаткові функції до чат-бота штучного інтелекту: перевірка вмісту інформації та фактів (використовуючи власну базу даних), розв'язання складних задач, пов'язаних із STEM, аналіз даних, генерування інфографіки [55].

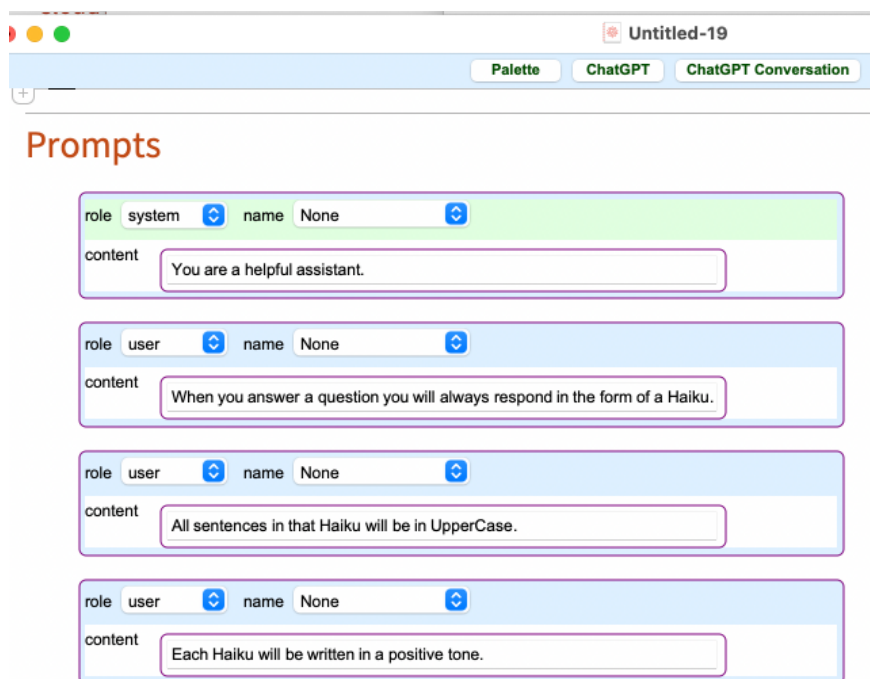


Рисунок 3.18 – Приклад налаштування мовного сеансу ChatGPT на основі пакету Wolfram [54]

Наведений на рисунку 3.18 приклад може допомогти створити персонального цифрового асистента - наприклад клієнта сільськогосподарського підприємства, який здійснюватиме покупку продукції в інтернет-магазині.

Для повноцінного використання можливостей мовних моделей на основі генеративного штучного інтелекту в персональних або комерційних цілях слід використовувати ChatGPT API, яке дозволяє розробникам інтегрувати ChatGPT у власні програми, продукти чи послуги. При цьому власникам таких ресурсів доцільно враховувати, що API ChatGPT не входить у підписку ChatGPT Plus (20 дол. США на місяць) і оплачується окремо.

Схожі опції персонального налаштування цифрового помічника є і в корпоративній версії моделі генеративного штучного інтелекту Microsoft Copilot. Підключення налаштованого цифрового асистента до пакетів MS Office може покращити роботу працівників сільськогосподарського підприємства із спільними документами, інструкціями тощо.

3.4. Проектування архітектури інтелектуальної інформаційної системи планування

Проектування архітектури інтелектуальної інформаційної системи (ІС) планування та управління процесами в сільському господарстві є ключовим етапом в розробці системи, яка здатна вирішувати завдання планування на основі аналізу та обробки великих обсягів даних. Архітектура ІС планування повинна бути гнучкою, масштабованою та ефективною для вирішення різноманітних завдань користувачів. Є ряд етапів, які треба буде реалізувати на практиці.

На початку необхідно ретельно вивчити та проаналізувати вимоги користувачів і бізнес-процесів в аграрній сфері, які система планування повинна підтримувати. Це допоможе визначити функціональні та нефункціональні вимоги до системи.

Розробити оптимальну структуру для зберігання та обробки даних, зокрема визначення форматів даних, баз даних, та механізмів для ефективного доступу до інформації. Обрати відповідні технології, які відповідають вимогам проекту. Сюди відноситься вибір мов програмування, фреймворків, баз даних, та інших інструментів.

Розбити систему на логічні модулі, що дозволяє забезпечити чітку організацію та можливість розвитку. Кожен модуль виконуватиме конкретні завдання, що полегшить розподіл роботи та підтримку коду. Архітектура інтелектуальної інформаційної системи для сільського господарства може включати архітектури обробки даних, модулів підтримки бізнес-процесів та архітектури додатків.

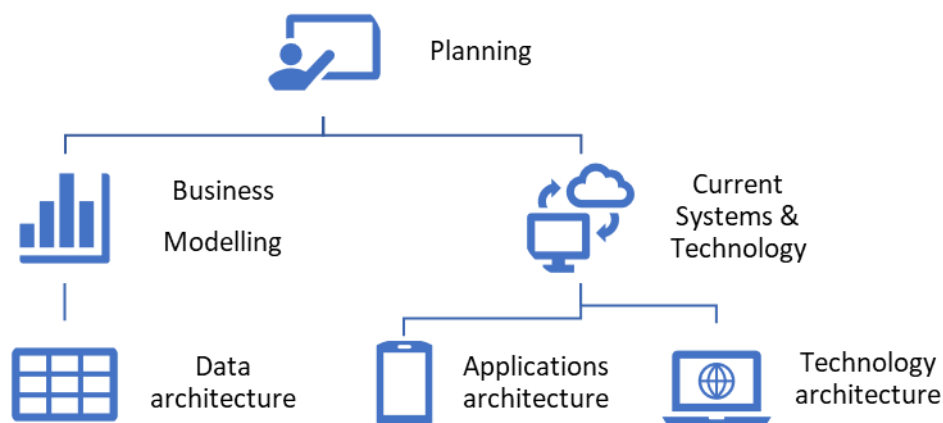


Рисунок 3.19 – Архітектура інформаційної системи планування в сільському господарстві [27]

Ядром інтелектуальної інформаційної системи планування аграрного підприємства є програмне забезпечення, яке містить інформацію про наявні виробничі структури та одиниці (поля, посіви, ферми), ресурси (земельні, людські ресурси, основні засоби, сільськогосподарська техніка) та бізнес-процеси в сільському господарстві. Обробка даних може відбуватися з врахуванням традиційних підходів, так і прогнозової аналітики на основі моделей машинного навчання.

Модулі ІС з планування та управління процесами інтегрується в систему об'єктів аграрного підприємства Farm Enterprise. Застосування загальних програмних компонентів та програмного забезпечення для підтримки виробничих процесів в поєднанні з програмними додатками дозволить забезпечити обмін даними та їх подальший аналіз.

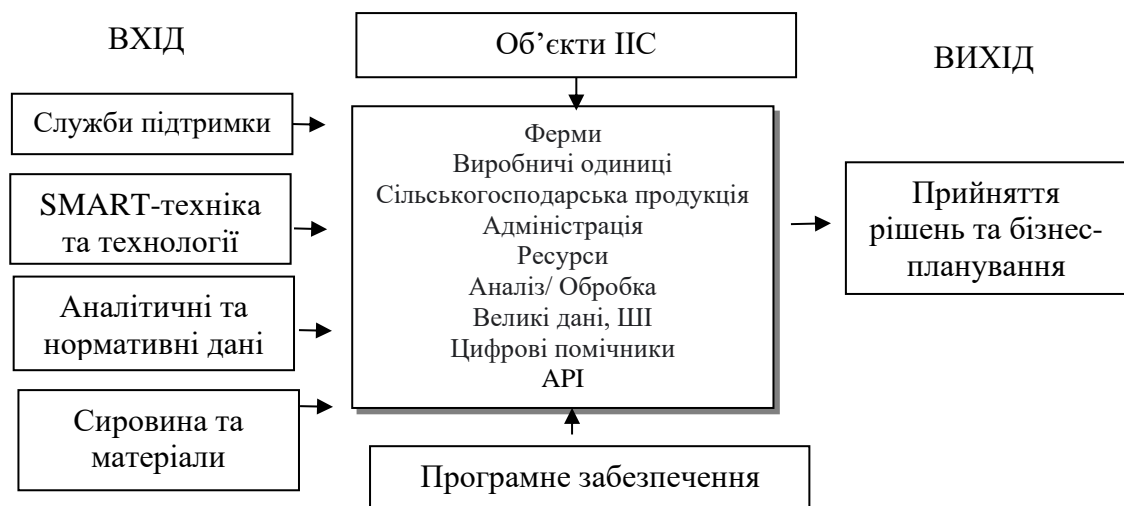


Рисунок 3.20 – Розроблена архітектура інтелектуальної інформаційної системи планування

Важливим етапом є визначення та впровадження заходів безпеки для захисту конфіденційності та цілісності даних інтелектуальної інформаційної системи та механізму її інтеграції з іншими системами. Слід розглянути також забезпечення системи можливістю масштабування для роботи з великим обсягом даних та оптимізації продуктивності у випадку збільшення господарства та диверсифікації господарської діяльності. Важливо також реалізувати ряд заходів та тестових процедур, валідації для перевірки функціональності і відповідності встановленим на попередніх етапах функціональним і нефункціональним вимогам.

Проектування архітектури ІС планування вимагає балансу між потребами користувачів, технічними обмеженнями та ефективністю системи для досягнення оптимальних результатів у плануванні.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Обґрунтування можливих чинників травмонебезпечних ситуацій

Кожного дня люди в побутових та робочих умовах стикаються з різноманітними ситуаціями, які можуть призвести до травм. Обґрунтування можливих чинників травмонебезпечних ситуацій є важливим завданням для розуміння та запобігання можливим негативним наслідкам. Найрізноманітніші фактори, такі як недбалість, необачність, технічні невдачі чи природні катастрофи, можуть стати витоків травматичних ситуацій. Ретельний аналіз цих чинників дозволяє розробити ефективні стратегії та заходи безпеки, спрямовані на зменшення ризику і забезпечення безпеки громадян у різних сферах їхнього життя.

Аграрна сфера, виходячи із специфіки ведення сільського господарства, характеризується підвищеними ризиками та специфічними чинниками травмонебезпечних ситуацій. Сільськогосподарські процеси, використання техніки, обробка ґрунту та обслуговування тварин можуть призвести до небезпеки для працівників та спричинити травматичні події. До чинників, що можуть викликати травмонебезпечні ситуації в аграрній галузі, можна віднести несправність сільськогосподарської техніки, недостатню підготовку персоналу, а також вплив непередбачуваних факторів, таких як погодні умови чи різкі зміни врожаю.

Прикладами травмонебезпечних ситуацій в сільському господарстві можуть бути: падіння з висоти (при роботі на складах, збиральних машинах); защемлення або порізи рук та ніг рухомими частинами машин та обладнання; травмування в тваринництві (укуси, удари ВРХ); отруєння хімічними речовинами (пестициди, добрива); перегрівання або переохолодження організму під час виконання польових робіт тощо.

Аналіз цих аспектів може допомогти розробити ефективні заходи безпеки в аграрному секторі, спрямовані на запобігання та зменшення травматичних ризиків для працівників у сільському господарстві.

Інформаційні технології можуть відігравати ключову роль у зниженні ризиків травмонебезпечних ситуацій в аграрній сфері. Застосування сучасних технологій та інтелектуальної інформаційної системи управління процесами може допомогти вдосконалити процеси аграрній сфері та забезпечити безпеку працівників.

Системи IoT (технології Інтернету речей) можуть використовуватися для постійного моніторингу та діагностики сільськогосподарської техніки. За допомогою сенсорів та вбудованих систем можна заздалегідь виявляти можливі несправності чи відхилення в роботі обладнання, що дозволяє уникнути аварій та травматичних ситуацій.

GPS-технології сприяють ефективному плануванню маршрутів для сільськогосподарської техніки. Це допомагатиме уникненню конфліктів та травматичних ситуацій на полі, а також забезпечує оптимальне використання ресурсів.

Електронні системи навчання та інформування можуть застосовуватися для підвищення обізнаності працівників щодо безпеки в аграрній сфері. Сюди можна віднести віртуальні тренажери, онлайн-курси та доступ до інформації про найкращі практики в галузі техніки та безпеки.

Використання автоматизованих систем та робототехніки може допомогти зменшити потребу в ручних операціях, що часто пов'язані з ризиком травмування. Роботи, які виконуються автономно чи за допомогою дистанційного управління, можуть забезпечувати вищий рівень безпеки. Наприклад під час обробітку земельних угідь, які були забруднені пестицидами або є ризик їх замінування в умовах військових дій.

Загалом, інтеграція інформаційних технологій в аграрний сектор може покращити якість роботи, знизити ризики травм і негативних подій, а також сприяти створенню більш безпечного та продуктивного середовища для працівників.

4.2. Умови та обставини виникнення небезпечних ситуацій та їх наслідки

Умови та обставини, що призводять до виникнення небезпечних ситуацій, є об'єктом уваги в безпеці та управлінні ризиками. Велика кількість чинників може сприяти виникненню небезпечних ситуацій у різних сферах життєдіяльності. Однією з головних умов є людський фактор, включаючи недбалість, недостатню підготовку, або неправильне використання техніки та обладнання.

Технічні невдачі та несправності обладнання також можуть бути вирішальними умовами виникнення небезпечних ситуацій. Погодні умови, природні катастрофи та інші природні явища також можуть впливати на створення небезпечних ситуацій та загроз.

У таблиці 4.1. наведено процес формування травмонебезпечних ситуацій при роботі з електрообладнанням в офісі ІТ-компанії.

Таблиця 4.1. - Процес формування травмонебезпечних ситуацій

Вид роботи, виробничий підрозділ	Виробнича безпека			Можливі наслідки	Заходи запобігання
	Небезпечна умова НУ	Небезпечна дія НД	Небезпечна ситуація НС		
Робота з електрообладнанням, використання розеток Блок-схема	НУ ₁ – обладнання зламалось; НУ ₂ – несправна електромережа НУ ₁ → НУ ₂ →	НД – перебування у небезпечній зоні працівника НД →	НС ₁ – пошкодження обладнання; НС ₂ – дотикання до корпусу; НС ₃ – ураження струмом НС ₁ → НС ₂ → НС ₃	Т – Травма → Т	Обладнати засобами захисту, стабілізатор струму та відсікач. Перевірити справність заземлення і електрообладнання

Травмонебезпечні ситуації в офісі та ІТ-відділі сільськогосподарського підприємства можуть виникати з різних причин, і їх важливо ідентифікувати та управляти для забезпечення безпеки працівників.

Довготривала робота за комп'ютером може призвести до фізичних проблем працівників, які працюють за ПК, таких як біль у спині, шії чи

зап'ястях. Неправильна організація робочого місця можуть призвести до погіршення здоров'я.

Несправності та непередбачувані технічні проблеми з комп'ютерною та офісною технікою можуть стати джерелом небезпеки та спричинити травматичні ситуації.

Наслідки небезпечних ситуацій можуть бути різноманітними і включати травми, матеріальні збитки, екологічні проблеми та навіть загрозу життю. Управління цими умовами та обставинами вимагає комплексного підходу, включаючи розробку ефективних стратегій безпеки, вдосконалення технічних засобів та впровадження сучасних технологій для прогнозування та управління ризиками.

Запобігання цим ситуаціям включає в себе впровадження ефективних політик охорони праці, навчання працівників техніці безпеки, виявлення ризиків та відповідний технічний облік.

4.3. Безпека в надзвичайних ситуаціях

Безпека в надзвичайних ситуаціях – це важливий аспект громадянської безпеки, спрямований на зменшення ризиків та мінімізацію можливих наслідків при аваріях, катастрофах чи інших небезпечних подіях. Важливим є розвиток систем та стратегій, що дозволяють ефективно впоратися з надзвичайними ситуаціями та захистити життя та майно населення.

Відповідно до діючого законодавства, під надзвичайною ситуацією слід розуміти ситуацію на окремій території, в межах організації, на водотоці чи водоймі, за якої порушуються нормальні умови життя населення.

Ефективні системи попередження та прогнозування надзвичайних ситуацій грають ключову роль у запобіганні та підготовці. Забезпечення доступу до точної та своєчасної інформації дозволяє людям уникнути небезпеки та прийняти необхідні заходи.

Надзвичайна ситуація може бути державною, регіональною, місцевою, об'єктовою. Відповідно до Закону України «Про правовий режим надзвичайного стану», надзвичайний стан є особливим правовим режимом, що обмежує певні конституційні права людей та надає додаткові повноваження органам державної влади.

Проведення навчань та симуляцій допомагає підготувати населення та працівників як аграрного підприємства, так і ІТ-компанії, до дій в надзвичайних ситуаціях. Це включає навички першої допомоги, використання евакуаційних шляхів та роботу з екстреною інструкцією.

Забезпечення безпечної інфраструктури та впровадження заходів захисту, таких як стабільні будівлі, протипожежне обладнання та системи контролю, грають важливу роль у зменшенні ризику. Ефективна комунікація та співпраця між рятувальними службами, органами влади та населенням є ключовими чинниками успіху в управлінні надзвичайними ситуаціями.

Умови військових дій породжують складні завдання для забезпечення безпеки працівників у компаніях, які функціонують в зонах конфліктів. Працівники підприємств стають уразливими перед небезпеками, що виникають як безпосереднім результатом збройних конфліктів.

Важливою метою управління в цих умовах є забезпечення не лише ефективної роботи підприємства, але й фізичного та психологічного благополуччя його персоналу. Застосування стратегій та політик безпеки, адаптованих до особливостей військових конфліктів, стає критичним аспектом управління ризиками та забезпеченням стабільності у важливих галузях економіки. В даному контексті, важливо розглядати інтегровані підходи до безпеки працівників як невід'ємну складову стратегії управління в умовах воєнних дій.

Один із ключових аспектів – це забезпечення фізичного захисту приміщень та працівників. Укриття надає фізичний захист від прямих загроз, таких як обстріли, вибухи чи інші форми нападу. Забезпечення працівників такими захисними спорудами є стратегічно важливим кроком для збереження

їхнього життя та здоров'я в умовах військових конфліктів. В компанії мають бути розроблені та реалізовані ефективні плани евакуації та алгоритм дій під час повітряної тривоги, обстрілів чи інших загроз. Будівлі мають мати сховище або вказівники, де воно знаходиться, а співробітники - інструкції та доступ до аптечок, а за потреби засоби для самозахисту від хімічних речовин тощо. Плани евакуації мають ретельно бути розроблені та доведені до працівників проводяться тренування з персоналом, щоб забезпечити ефективну реакцію у випадку надзвичайних обставин.

Умови війни вимагають оцінку, планування та аналізу ризиків як бізнес-процесів, так і безпеки працівників, підтримки технологічних процесів в критичних галузях (наприклад евакуація тварин або забезпечення їх годівлі в цих умовах). Компанії повинні мати детальні плани дій у випадку загострення конфлікту, включаючи можливості віддаленої роботи, забезпечення зв'язку та збереження даних.

РОЗДІЛ 5

ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Сучасні аграрні виробники отримують значні вигоди від інформаційних технологій у сільському господарстві, адже дозволяють економити ресурси, знижувати витрати, підвищувати продуктивність праці та ефективність технологічних процесів.

В цілому ефективність інтелектуальної інформаційної системи прийняття рішень та управління процесами на основі моделей машинного навчання залежить від напрямку та цілей застосування даного виду програмного забезпечення (рис.5.1), технологічного оснащення виробника, наявності смарт-технологій та ступеня впровадження технологій точного землеробства тощо.

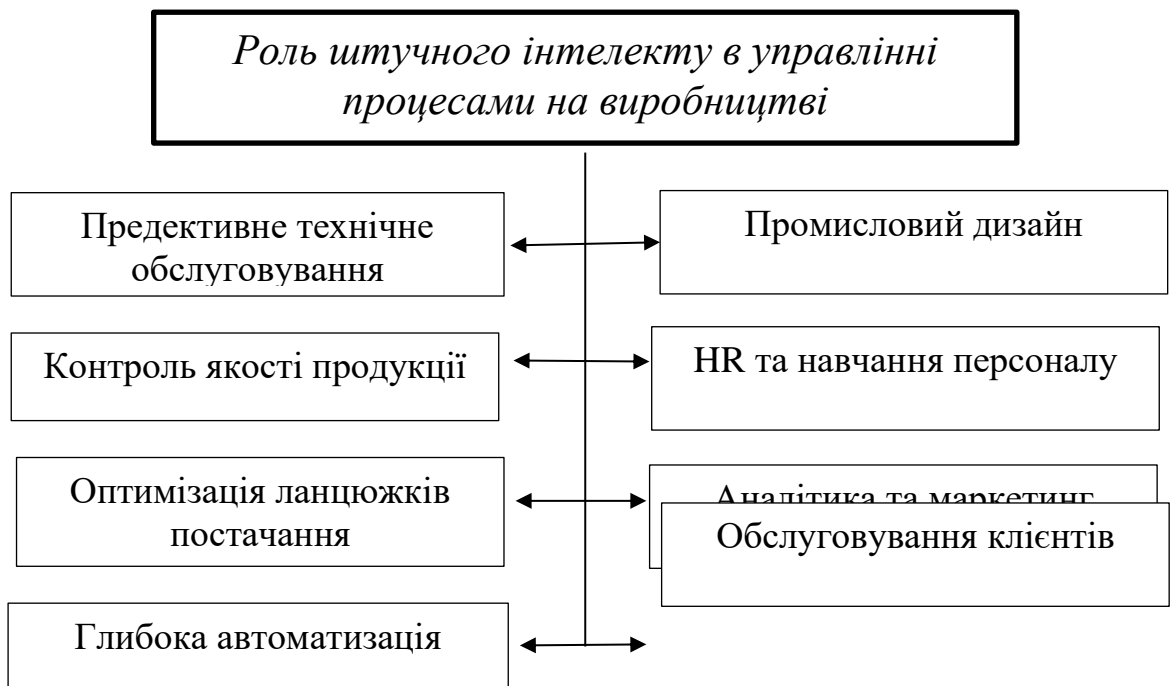


Рисунок 5.1 – Роль штучного інтелекту в управлінні процесами [56]

Використовуючи смарт технології, датчики, технології Інтернету речей в аграрній сфері, сільськогосподарські підприємства останніми роками почали накопичувати і зберігати надзвичайну кількість різноманітних даних, які можна використовувати для вирішення різних проблем. Тому традиційні методи отримання інформації з даних на сьогодні вже не є достатніми, враховуючи обсяг, швидкість і різноманіття даних.

Застосування інтелектуальної інформаційної системи в аграрній сфері безумовно надаватиме ряд переваг, до яких можна віднести зниження рівня споживання води, поживних речовин та добрив, засобів захисту рослин, зниження негативного впливу на навколишню екосистему, зменшення хімічного стоку у місцеві ґрунтові води та річки, підвищення ефективності виробництва та продуктивності праці, зниження собівартості продукції та багато іншого. Це в свою чергу забезпечуватиме належний рівень рентабельності аграрного бізнесу, його стійкості до негативних впливів зовнішнього середовища.

Нові моделі та методи штучного інтелекту і розроблені на їх основі ІС можуть допомогти керівникам аграрних підприємств автоматизувати складні завдання на основі даних про сільське господарство. Для прикладу застосування таких можливостей штучного інтелекту, таких як аналітика тексту та обробка природної мови, може значно зменшити або усунути ручне сортування записів в Інтернеті чи на паперових носіях. Використання моделей розпізнавання символів пришвидшуватиме ряд рутинних операцій, пов'язаних із обробкою документації, даних з етикеток, позитивно впливаючи на ці процеси. Застосування безпілотників у поєднанні з можливостями технологій комп'ютерного зору дадуть швидко оцінити стан та визначити заходи боротьби зі шкідниками.

Програмне забезпечення для аграрного сектору може передбачати використання і інших передових технологій для вирішення ключових завдань в сільському господарстві, що сприятиме збільшенню обсягу даних та підвищенню ефективності сільськогосподарської діяльності.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи проаналізовано основні підходи та практики застосування моделей машинного навчання як складової інтелектуальної інформаційної системи планування та управління процесами у сільському господарстві.

Розглянуто питання впливу смарт-технологій як вихідного джерела надходження нових даних в сільському господарстві на функціонування інформаційної системи планування в цілому.

Інтелектуальні технології докорінно змінюють саму концепцію ведення господарської діяльності в аграрній сфері, впроваджуючи різноманітні новітні технології, такі як ГІС технології та GPS в сільському господарстві, супутникових знімків, дронів, інформаційних технологій та машинне навчання для обробки даних, платформи аналітики та оптимізації, колекції та набори даних.

З метою дослідження впливу машинного навчання та штучного інтелекту на інтелектуальні інформаційні системи планування, було проаналізовано динаміку їх розвитку протягом останніх років та встановлено бурхливий розвиток і застосування різноманітних моделей штучного інтелекту, у тому числі побудованих на основі великих мовних моделей.

Для реалізації поставленого завдання з обґрунтування інструментарію для реалізації моделі машинного навчання інтелектуальної інформаційної системи планування було обрано інтерактивне середовище Jupyter Notebook.

Вибір алгоритму машинного навчання для прогнозування залежить від характеристик набору даних. Для вирішення поставленої задачі, а саме планування показників та процесів вирощування озимої пшениці у сільськогосподарському підприємстві на основі зібраних та підготовлених даних було здійснено вибір алгоритму для моделі машинного навчання. Отримані результати оцінки якості набору даних показали, що набір є

якісним та може використовуватися для побудови моделі машинного навчання.

Одним з важливих етапів вибору алгоритму та моделі машинного навчання є оцінка якості моделі, яка дозволила визначити, наскільки точно та надійно модель виконує прогнози на нових даних, а також може виявляти можливі перекоси чи недоліки. В ході виконання кваліфікаційної роботи, виходячи із поставленого завдання та особливостей задачі, був здійснений вибір наступних алгоритмів для навчання моделі прогнозування урожайності озимої пшениці: найменших квадратів (OLS), градієнтного бустингу (XGBoost), лінійної регресії з поліноміальними ознаками.

За отриманими даними дослідження можна сказати, що найкращою моделлю машинного навчання визначено модель градієнтного бустингу (XGBoost), яка має найкращі значення всіх метрик якості та робить найточніші прогнози значень залежної змінної, а саме показника урожайності озимої пшениці. На основі обраної прогнозної моделі можна створювати інтелектуальну інформаційну систему планування процесів в сільському господарстві з модулем для планування збору врожаю озимої пшениці за відомої прогнозованої урожайності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шаров С.В., Лубко Д.В. Розробка інтелектуальної інформаційної системи для птахівництва. *Системи обробки інформації*, Вип.4 (150), 2017. С.170-174.
2. O, Ying-Lie. (2004). Intelligent Information Systems for Decision Support.. 1. pp.326-332.
3. Three Examples of How Companies Make Data-Driven Decisions [Електронний ресурс]. – Режим доступу: <https://programs.online.utica.edu/resources/article/data-driven-decisions>
4. IT for Enterprise AI: Model-Driven Architecture [Електронний ресурс]. – Режим доступу: <https://c3.ai/what-is-enterprise-ai/it-for-enterprise-ai/model-driven-architecture/>
5. Lycett, Mark & Marcos, Esperanza & Storey, Veda. (2007). Model-driven systems development: An introduction. *European Journal of Information Systems*. 16. 10.1057/palgrave.ejis.3000684.
6. Alexander S. Gillis. Definition knowledge-based systems (KBSes). [Електронний ресурс]. – Режим доступу: <https://www.techtarget.com/searchcio/definition/knowledge-based-systems-KBS>
7. What is Case-based learning (CBL)? [Електронний ресурс]. – Режим доступу: <https://gamco.es/en/glossary/case-based-learning-cbl/>
8. Shiqiang Zh., Ting Y., Tao X., Hongyang Ch., Schahram D., Sylvain G., Deniz G., Ekram H., Yaochu J., Feng L., et al. Intelligent Computing: The Latest Advances, Challenges, and Future. *Intell Comput.* 2023;2:0006.DOI:10.34133/icomputing.0006
9. Knowledge Base Systems [Електронний ресурс]. – Режим доступу: <https://indiafreenotes.com/knowledge-base-systems/>
10. What is corpus/corpora in text mining? [Електронний ресурс]. – Режим доступу: <https://www.quora.com/What-is-corpus-corpora-in-text-mining>

11. Peddineni A. What can you do with text mining? [Електронний ресурс]. – Режим доступу: <https://www.quora.com/What-can-you-do-with-text-mining>
12. Kurama V. Information Extraction. [Електронний ресурс]. – Режим доступу: <https://nanonets.com/blog/information-extraction/>
13. Prithiv S. Top 10 OCR Software in 2024 [Електронний ресурс]. – Режим доступу: <https://nanonets.com/blog/ocr-software-best-ocr-software/>
14. Alam F. Applications of Natural Language Processing [Електронний ресурс]. – Режим доступу: <https://datasciencedojo.com/blog/natural-language-processing-applications/#>
15. What is deep learning? [Електронний ресурс]. – Режим доступу: <https://www.ibm.com/topics/deep-learning>
16. Kamilaris A., Prenafeta-Boldú F., Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, Volume 147, 2018, pp.70-90, <https://doi.org/10.1016/j.compag.2018.02.016>.
17. Копустинський К.В., Шаров С.В. Напрямки використання інтелектуальних інформаційних систем. – Режим доступу: <https://core.ac.uk/download/pdf/233567899.pdf>
18. Здітовецький Ю.С., Бісікало О.В., Іванов Ю.Ю. Інтелектуальна інформаційна система розпізнавання та аналізу складу продуктів харчування. Вісник Вінницького політехнічного інституту. №2(2023). Вип. 2. С. 66–71.
19. Alton L. Intelligent Systems Vs. Artificial Intelligence: How To Help Your Clients. – Режим доступу: <https://www.datasciencecentral.com/intelligent-systems-vs-artificial-intelligence-how-to-help-your/>
20. Nawarecki, E. & S., Kluska-Nawarecka & Wilk-Kołodziejczyk, Dorota & Śnieżyński, Bartłomiej & Legień, Grzegorz. (2018). Integrated Multi-functional LPR Intelligent Information System. 10.1007/978-3-319-62120-3_12.
21. Machine Learning, ML [Електронний ресурс]. – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>

22. Шаров С.В. Лубко Д.В., Осадчий В.В. Інтелектуальні інформаційні системи: навч. посіб. Мелітополь: Вид-во МДПУ ім. Б. Хмельницького, 2015. – 144 с.

23. Запорожець Т.В. Застосування інтелектуальних технологій та систем штучного інтелекту для підтримки прийняття управлінських рішень. Вчені записки ТНУ імені В.І. Вернадського. Том 31 (70) № 2 2020. С.79-85.

24. Alton L. Intelligent Systems Vs. Artificial Intelligence: How To Help Your Clients [Електронний ресурс]. – Режим доступу: <https://www.datasciencecentral.com/intelligent-systems-vs-artificial-intelligence-how-to-help-your/>

25. Машинне навчання – основа штучного інтелекту [Електронний ресурс]. – Режим доступу: <https://artline.ua/uk/news/mashinnoe-obuchenie-osnova-sovremennogo-iskusstvennogo-intellekta>

26. Shiraly K. AI in Nutrition: How Technology is Transforming what We Eat [Електронний ресурс]. – Режим доступу: <https://www.width.ai/post/ai-in-nutrition>.

27. Zhelyeznyak A., Ptashnyk V. Modelling the architecture of a planning system for agricultural enterprises. *Selected Papers from the Xth International Conference «Information technologies in energy and agro-industrial complex»*, ІТЕА 2021, October 6-8, 2021, 2022. pp.32-37.

28. Полянчиков С., Капітанська О. Інтелектуальне сільське господарство. [Електронний ресурс]. – Режим доступу: <https://www.agronom.com.ua/intelektualne-silske-gospodarstvo/>

29. What is smart agriculture and why smart agriculture is the future? [Електронний ресурс]. – Режим доступу: <https://ondo.io/what-is-smart-agriculture/>

30. Hua Leong, F., Farn Haur, C. Deep Learning-Based Text Recognition of Agricultural Regulatory Document. In: Bădică, C., Treur, J., Benslimane, D., Hnatkowska, B., Krótkiewicz, M. (eds) *Advances in Computational Collective*

Intelligence. ICCCI 2022. *Communications in Computer and Information Science*, vol 1653. Springer, Cham. https://doi.org/10.1007/978-3-031-16210-7_18

31. Predictive Modeling Techniques - A Comprehensive Guide. [Электронный ресурс]. – Режим доступа: <https://www.projectpro.io/article/predictive-modelling-techniques/598>

32. Halton C. Predictive Analytics: Definition, Model Types, and Uses. [Электронный ресурс]. – Режим доступа: <https://www.investopedia.com/terms/p/predictive-analytics.asp>

33. Le J. The Top 10 Machine Learning Algorithms Every Beginner Should Know [Электронный ресурс]. – Режим доступа: <https://builtin.com/data-science/tour-top-10-algorithms-machine-learning-newbies>

34. Gangadhar K. A Simple Guide to Choosing the Right Machine Learning Algorithm for Real-World Success [Электронный ресурс]. – Режим доступа: <https://www.linkedin.com/pulse/simple-guide-choosing-right-machine-learning-success-gangadhar/>

35. Buhl N. F1 Score in Machine Learning [Электронный ресурс]. – Режим доступа: <https://encord.com/blog/f1-score-in-machine-learning/>

36. Alto V. Understanding Ordinary Least Squares (OLS) Regression. [Электронный ресурс]. – Режим доступа: <https://builtin.com/data-science/ols-regression>

37. XgBoost [Электронный ресурс]. – Режим доступа: <https://itwiki.dev/data-science/ml-reference/ml-glossary/xgboost>

38. Introduction to Boosted Trees [Электронный ресурс]. – Режим доступа: <https://xgboost.readthedocs.io/en/stable/tutorials/model.html>

39. Visva. Demystifying Polynomial Regression: Understanding and Implementation [Электронный ресурс]. – Режим доступа: <https://medium.com/@vk.viswa/demystifying-polynomial-regression-understanding-and-implementation-5f5635870b0c>

40. Kaffayatullah K., Waqas Ah., Muhammad A., Ayaz A., Sohaib N., Anas A. Compressive Strength Estimation of Steel-Fiber-Reinforced Concrete and

Raw Material Interactions Using Advanced Algorithms. *Polymers*. 14. 2022. 10.3390/polym14153065.

41. Spanton R. Polynomial Regression: An Introduction. [Електронний ресурс]. – Режим доступу: <https://builtin.com/machine-learning/polynomial-regression>

42. Jupyter Notebooks [Електронний ресурс]. – Режим доступу: <https://threathunterplaybook.com/tutorials/jupyter/introduction.html>

43. Підгайна Є. Цифрові помічники: як інновації запускають для нафтогазу, енергетики, хімічної промисловості та будівництва [Електронний ресурс]. – Режим доступу: <https://mind.ua/publications/20225863-cifrovi-pomichniki-yaki-innovaciyi-zapuskayut-dlya-naftogazu-energetiki-himichnoyi-promislovesti-ta-bud>

44. Scheldt A. Digital Human as a Service (DHaaS). [Електронний ресурс]. – Режим доступу: <https://www.webopedia.com/definitions/digital-human-as-a-service-dhaas/>

45. Желізняк А.М., Пташник В.В., Смолінський В.Б., Падюка Р.І. Віртуальні помічники на основі нейронних мереж для прийняття рішень в сільському господарстві. *Інформаційні технології в енергетиці та агропромисловому комплексі*. Матеріали XII Міжнародної науково-практичної конференції, м.Львів, 4-6 жовтня 2023 р. / Львів.нац. ун-т природ. Львів, 2023. С.108-109.

46. BentoML: The Unified AI Application Framework [Електронний ресурс]. – Режим доступу: <https://github.com/bentoml/BentoML>

47. 17 Top MLOps Tools You Need to Know [Електронний ресурс]. – Режим доступу: <http://surl.li/qsobm>

48. Як підготувати набір даних для машинного навчання за 7 кроків [Електронний ресурс]. – Режим доступу: <https://www.zfort.com.ua/blog/yak-pidgotuvati-nabir-danikh-mashinnogo-navchannya-za-7-krokv>

49. Data Preparation for Machine Learning: The Ultimate Guide to Doing It Right [Електронний ресурс]. – Режим доступу: <https://www.pecan.ai/blog/data-preparation-for-machine-learning/>
50. Alkhaldi N. How your company could benefit from automated data collection [Електронний ресурс]. – Режим доступу: <https://itrexgroup.com/blog/how-your-company-could-benefit-from-automated-data-collection/>
51. Shashkina V. Data preparation for machine learning: a step-by-step guide [Електронний ресурс]. – Режим доступу: <https://itrexgroup.com/blog/data-preparation-for-machine-learning/>
52. Rouse M. Generative AI [Електронний ресурс]. – Режим доступу: <https://www.techopedia.com/definition/34633/generative-ai>
53. Rouse M. Prompt-Based Learning [Електронний ресурс]. – Режим доступу: <https://www.techopedia.com/definition/34832/prompt-based-learning>
54. Wolfram languagepaclet repository. Community-contributed installable additions to the wolfram language [Електронний ресурс]. – Режим доступу: <https://resources.wolframcloud.com/PacletRepository/>
55. Maxwell T. 3 Ways to Use ChatGPT'S Wolfram Plugin [Електронний ресурс]. – Режим доступу: <https://www.makeuseof.com/use-chatgpt-wolfram-plugin/>
56. Штучний інтелект у виробництві: переваги та приклади застосування [Електронний ресурс]. – Режим доступу: <https://wezom.com.ua/ua/blog/shtuchniy-intelekt-u-virobnitstvi-perevagi-ta-prikladi-zastosuvannya>
57. Тригуба А., Тригуба І., Чубик Р., Кондисюк І., Коваль Н., Панюра Я. Прогнозування обсягів заготівлі сировини на території громад із використанням штучних нейронних мереж. Вісник Львівського національного аграрного університету: Агроінженерні дослідження. 2020. № 24. С. 143-151.
58. Тригуба А., Коваль Н., Тригуба І., Падюка Р., Боярчук О. Системна модель цифрової трансформації сільських територіальних громад на основі

обчислювального інтелекту. Вісник Львівського національного аграрного університету: агроінженерні дослідження. №26. Львів: Львів НАУ, 2022. С.177-184.

59. Tryhuba, A., Kondysiuk, I., Tryhuba, I., Voiarchuk, O., Tatomyr, A. Intellectual information system for formation of portfolio projects of motor transport enterprises. CEUR Workshop Proceedings, 2022, 3109, pp. 44–52.

60. Tryhuba A., Batyuk B., Dyndyn M. Coordination of Configurations of Complex Organizational and Technical Systems for Development of Agricultural Sector Branches. Journal of Automation and Information Sciences.52(2). pp.63-76

61. Тимочко В., Городецький І.М., Березовецький А.П., Войналович О.В., Вісин О.О. Аналіз нормативної бази безпеки праці для механізованого обприскування сільськогосподарських культур. Науковий журнал «Техніка та енергетика» / «Machinery & Energetics». Київ, 2021. Vol.12, № 2. С.23-31. <http://dx.doi.org/10.31548/machenergy2021.02.023>.

62. Войналович О., Тимочко В., Гнатюк О., Городецький І.М. Визначення ризику травмування під час тракторних робіт на основі дефектоскопічного контролю. Вісник Львівського національного аграрного університету: Агроінженерні дослідження. Львів, 2019. № 23. С.131-136. DOI: <https://doi.org/10.31734/agroengineering2019.23.131>.

63. Городецький І. М., Мазур І.Б., Городецька Н. Г., Березовецький А.П. Вплив обставин на формування небезпечних ситуацій аграрного виробництва. Вісник Львівського НАУ: Агроінженерні дослідження. 2017. № 21. С. 162–166.

ДОДАТКИ

Додаток А

Таблиця 1 – Порівняльний аналіз доступного програмного забезпечення, розробленого на основі методів оптичного розпізнавання символів OCR *

Назва	Опис та призначення	Характеристика
1	2	3
Наномережі (<i>Nanonets</i>)	програмне забезпечення OCR на основі штучного інтелекту, яке автоматизує збір даних для інтелектуальної обробки документів рахунків-фактур, квитанцій, ідентифікаційних карток тощо	<p>Плюси:</p> <ul style="list-style-type: none"> ✓ Сучасний інтерфейс користувача ✓ Обробляє великі обсяги документів ✓ Розумна ціна та простота використання ✓ Вилучення даних з нульовим ударом або без навчання ✓ Не вимагає внутрішньої команди розробників ✓ Алгоритм/моделі можна навчати/перенавчати ✓ Хороша документація та підтримка ✓ Безліч варіантів налаштування ✓ Широкий вибір варіантів інтеграції ✓ Працює не тільки з англійською або кількома мовами ✓ Повна двостороння інтеграція з кількома програмами для бухгалтерського обліку ✓ Хороший OCR API для розробників <p>Мінуси:</p> <ul style="list-style-type: none"> ✓ Інтерфейс захоплення таблиці може бути кращим
ABBYY Flexcapture	стабільне, масштабоване програмне забезпечення для створення зображень документів і вилучення даних, яке автоматично перетворює документи будь-якої структури, мови чи вмісту в придатні для використання та доступні бізнес-дані	<p>Плюси:</p> <ul style="list-style-type: none"> ✓ Дуже добре розпізнає зображення ✓ Легко зберігати друкований результат ✓ Добре інтегрується з ERP-системами ✓ Автоматизує вилучення даних з документів (певною мірою) <p>Мінуси:</p> <ul style="list-style-type: none"> ✓ Початкове налаштування може бути важким і складним ✓ Не налаштована автоматична обробка рахунків ✓ Немає готових шаблонів ✓ Важко налаштувати ✓ Немає доступних ресурсів ✓ Може мати кращу інтеграцію з рішеннями RPA ✓ Низька точність із зображеннями/документами з низькою роздільною здатністю ✓ Пакетні перевірки припиняються, навіть якщо є помилка лише в певному розділі

		<ul style="list-style-type: none"> ✓ Повідомлення про помилки рядків-позицій з'являються навіть для елементів, які слід пропустити ✓ RESTful API недоступний у локальній версії
ABBY FineReader PDF	<p>програмне забезпечення для оптичного розпізнавання тексту з підтримкою редагування PDF-файлів. Програма дозволяє конвертувати графічні документи в редаговані електронні формати.</p>	<ul style="list-style-type: none"> ✓ Зручний для клавіатури OCR-редактор для внесення виправлень вручну ✓ Надзвичайно зрозумілий інтерфейс ✓ Експортує в декілька форматів ✓ Унікальна функція порівняння документів <p>Мінуси:</p> <ul style="list-style-type: none"> ✓ Немає повнотекстового індексування для швидкого пошуку ✓ Потрібна крива навчання ✓ Ціни можуть бути завищеними ✓ Неможливість перегляду історії змін документа ✓ Неможливо об'єднати кілька файлів в один ✓ Може знадобитися додаткова обробка ✓ Інтерфейс користувача спочатку може бути приголомшливим ✓ Повільно обробляє великі файли
IBM Datacap	<p>оптимізує захоплення, розпізнавання та класифікацію бізнес-документів для отримання важливої інформації з них, має потужний механізм OCR. Працює в кількох каналах, включаючи сканери, мобільні пристрої, багатофункціональні периферійні пристрої та факс.</p>	<p>Плюси:</p> <ul style="list-style-type: none"> ✓ Налаштовує складні програми для збору даних ✓ Механізм сканування ✓ Простота використання <p>Мінуси:</p> <ul style="list-style-type: none"> ✓ Дуже мало онлайн-підтримки ✓ Інтерфейс може бути більш інтуїтивно зрозумілим ✓ Налаштування може бути громіздким ✓ Повільно ✓ Створити налаштований потік непросто ✓ Пакетні коміти вимагають часу
Google Document AI	<p>Одне з рішень у пакеті Google Cloud AI, Document AI (DocAI). Представляє собою консоль обробки документів, яка використовує машинне навчання для автоматичної класифікації, вилучення, збагачення даних і розблокування</p>	<p>Плюси:</p> <ul style="list-style-type: none"> ✓ Легко налаштувати ✓ Дуже добре інтегрується з іншими службами Google ✓ Зберігання інформації ✓ Швидкість <p>Мінуси:</p> <ul style="list-style-type: none"> ✓ Модулі AI не мають належної документації ✓ Налаштувати існуючі модулі та бібліотеки важко ✓ Не підходить для Python або інших мов програмування

	аналітичних даних у документах.	<ul style="list-style-type: none"> ✓ Застаріла документація API ✓ Дорого ✓ Не підходить для розгортання гібридної хмари ✓ Не підходить для випадків використання, які потребують спеціальних алгоритмів ШІ
AWS Texttract	автоматично витягує текст та інші дані зі сканованих документів за допомогою машинного навчання та OCR. Ідентифікація, розуміння та вилучення даних із форм і таблиць.	<ul style="list-style-type: none"> ✓ Платіжна модель за використання ✓ Простота використання <p>Мінуси:</p> <ul style="list-style-type: none"> ✓ Не піддається навчанню ✓ Різна точність ✓ Не призначена для рукописних документів
Docparser	хмарне програмне забезпечення для обробки документів і OCR, яке може автоматизувати малоцінні завдання та робочі процеси для компаній	<p>Плюси:</p> <ul style="list-style-type: none"> ✓ Легке налаштування ✓ Інтеграція Zapier <p>Мінуси:</p> <ul style="list-style-type: none"> ✓ Нестабільність використання Web Hook ✓ Потрібна певна підготовка, щоб зрозуміти правила розбору ✓ Недостатня кількість шаблонів ✓ Зональний підхід OCR (не може працювати з невідомими шаблонами) ✓ Інтерфейс може бути кращим ✓ Повільно завантажуються сторінки ✓ Документація могла б бути кращою
Klipa	надає автоматизовані рішення для керування документами, обробки, класифікації та вилучення даних для оцифрування паперових документів	<p>Плюси:</p> <ul style="list-style-type: none"> ✓ Швидке налаштування ✓ Чудова підтримка та API для розробників ✓ Чітка та лаконічна документація API ✓ Добре поєднується з бухгалтерськими програмами ✓ Конкурентоспроможна ціна ✓ Інтеграції <p>Мінуси:</p> <ul style="list-style-type: none"> ✓ Розпізнавання OCR може бути кращим ✓ Обмежені налаштування шаблону ✓ Обмежені налаштування білої мітки ✓ Групові коригування не підтримуються ✓ Неможливо навчити модель OCR ✓ Процес вибору непростий, оскільки є багато варіантів

*-побудовано на основі [13]

Додаток Б

Таблиця 1 – Застосування смарт-технологій для планування процесів на основі інтелектуальних інформаційних систем у рослинництві

Назва технології	Застосування для реалізації планування процесами
<i>ГІС технології</i>	Складання карти поточних і майбутніх змін кількості опадів, температури, врожайності, здоров'я рослин, оптимізація внесення добрив та пестицидів на певних ділянках, економія ресурсів, часу. Основа для збору даних структурованих та неструктурованих даних про стан рослин, ґрунту, рельєф місцевості для виконання технологічних процесів (оранка, передпосівний обробіток ґрунту тощо).
<i>Супутникові дані</i>	Прогнозування врожайності, моніторинг полів, виявлення загроз за допомогою супутникових даних, отримання зображень, застосування спектральних індексів (наприклад вегетаційного індексу NDVI, індексу вмісту хлорофілу в покривах CCCI, нормалізованого індексу на вміст азоту RedEdge (NDRE), модифікованого ґрунтово-корегованого вегетаційного індексу MSAVI).
<i>Дрони</i>	Визначення біомаси врожаю, фази росту та висоти рослин, забур'яненості посівів, наявності води або засухи на окремих ділянках поля. Отримання більш якісних та точних даних та зображень із кращою роздільною здатністю. Визначення наявності шкідників, пошкоджених посівів внаслідок обробітку ґрунту та рослин ЗЗР (у т.ч. неякісними), недостатньої кількості внесених органічних та мінеральних добрив.
<i>Аналіз даних про погоду (метеоумов)</i>	Використання численних наборів даних про метеоумови. Встановлення загроз, пов'язаних із вимерзанням рослин (низькі температури), загрози заморозків (при зниженні температури нижче -6°C), загрози від посухи (дні з підвищенням температури вище +30°C).
<i>Технології штучного інтелекту та машинного навчання</i>	Виявлення шкідників і хвороб і боротьби з ними до оптимізації росту культур і прогнозування врожайності. Застосування в комплексі із роботами та дронами для перевірки посівів, стану ґрунту, оцінки стану посівів, планування процесів поливу, удобрення та збирання урожаю. Оптимізація використання ресурсів (води, добрив, ЗЗР, енергії, пального), часу та підвищення стійкості сільськогосподарського підприємства. Відстеження та обробка даних про рівень освітленості, забрудненості та якості води. Виявлення потенційних проблем, попередження про можливі захворювання рослин, дефіцит поживних речовин або появу шкідників. Оптимізація роботи зрошувальних систем. Підвищення ефективності збору врожаю на основі використання технологій комп'ютерного зору, машинного навчання та ШІ для ідентифікації дозрілого урожаю та визначення початку його збору.

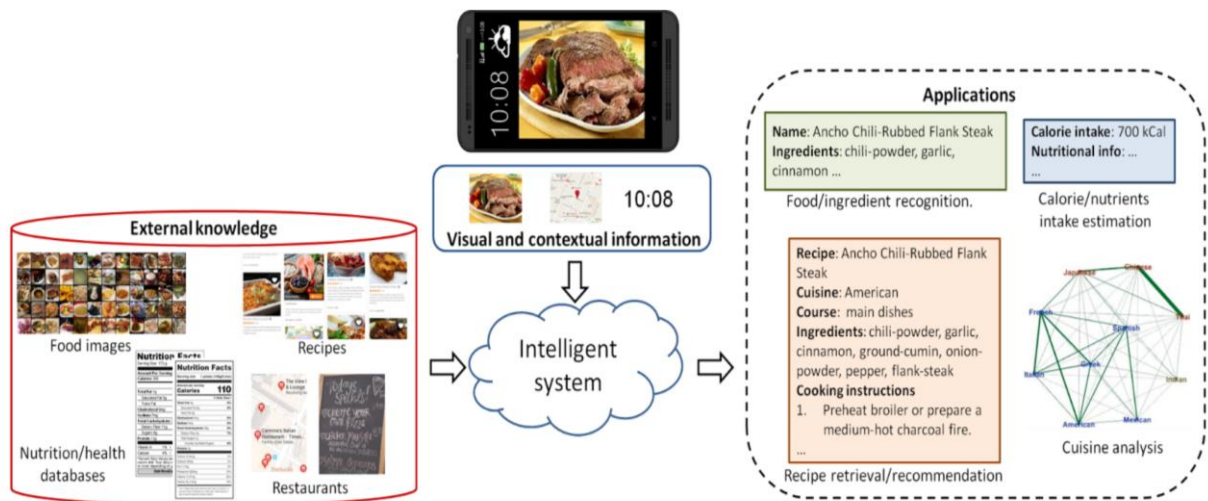


Рисунок 1 – Структура інтелектуальної інформаційної системи для ідентифікації їжі та планування харчування [26]

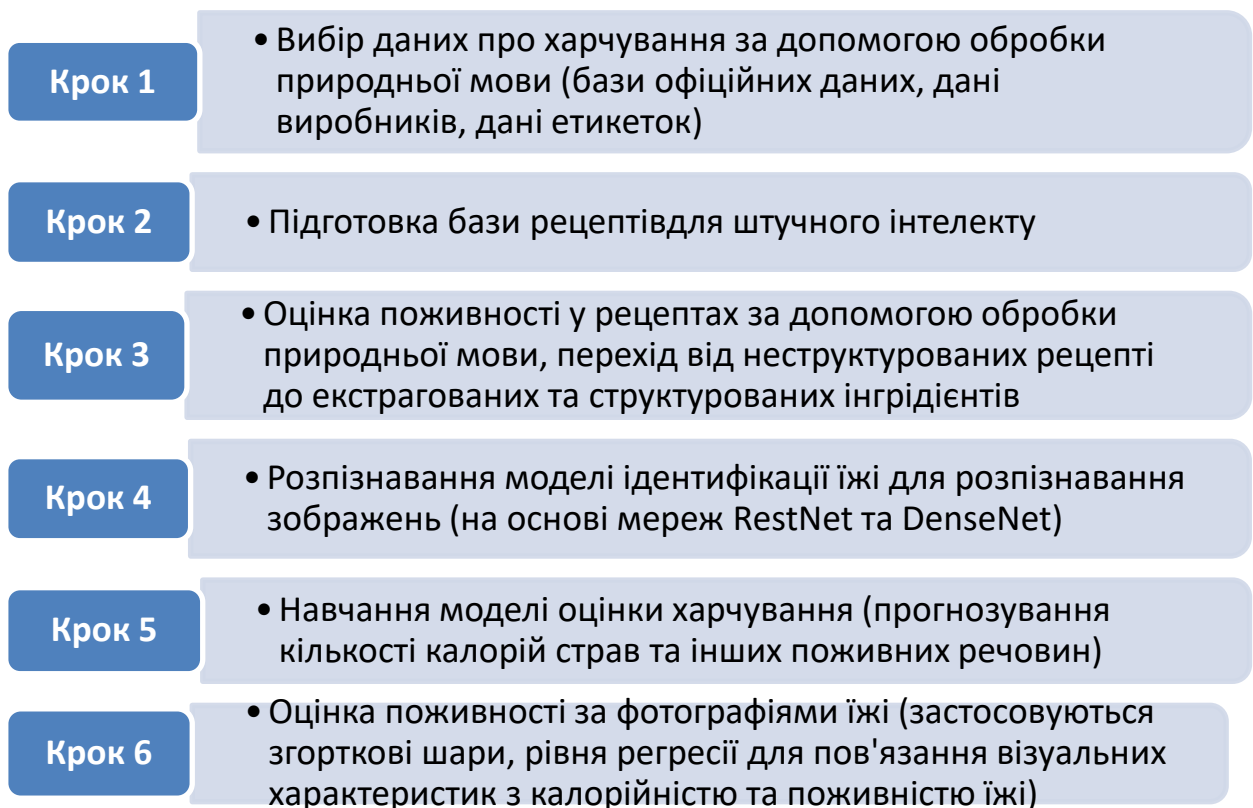


Рисунок 2 – Механізм дослідження оцінки калорійності та поживності порції їжі з використанням штучного інтелекту [26]

Додаток В

Таблиця 1 – Застосування смарт-технологій для планування процесів на основі інтелектуальних інформаційних систем у рослинництві

Назва технології	Застосування для реалізації планування процесами
<i>ГІС технології</i>	Складання карти поточних і майбутніх змін кількості опадів, температури, врожайності, здоров'я рослин, оптимізація внесення добрив та пестицидів на певних ділянках, економія ресурсів, часу. Основа для збору даних структурованих та неструктурованих даних про стан рослин, ґрунту, рельєф місцевості для виконання технологічних процесів (оранка, передпосівний обробіток ґрунту тощо).
<i>Супутникові дані</i>	Прогнозування врожайності, моніторинг полів, виявлення загроз за допомогою супутникових даних, отримання зображень, застосування спектральних індексів (наприклад вегетаційного індексу NDVI, індексу вмісту хлорофілу в покривах CCCI, нормалізованого індексу на вміст азоту RedEdge (NDRE), модифікованого ґрунтово-корегованого вегетаційного індексу MSAVI).
<i>Дрони</i>	Визначення біомаси врожаю, фази росту та висоти рослин, забур'яненості посівів, наявності води або засухи на окремих ділянках поля. Отримання більш якісних та точних даних та зображень із кращою роздільною здатністю. Визначення наявності шкідників, пошкоджених посівів внаслідок обробітку ґрунту та рослин ЗЗР (у т.ч. неякісними), недостатньої кількості внесених органічних та мінеральних добрив.
<i>Аналіз даних про погоду (метеоумов)</i>	Використання численних наборів даних про метеоумови. Встановлення загроз, пов'язаних із вимерзанням рослин (низькі температури), загрози заморозків (при зниженні температури нижче -6°C), загрози від посухи (дні з підвищенням температури вище +30°C).
<i>Технології штучного інтелекту та машинного навчання</i>	Виявлення шкідників і хвороб і боротьби з ними до оптимізації росту культур і прогнозування врожайності. Застосування в комплексі із роботами та дронами для перевірки посівів, стану ґрунту, оцінки стану посівів, планування процесів поливу, удобрення та збирання урожаю. Оптимізація використання ресурсів (води, добрив, ЗЗР, енергії, пального), часу та підвищення стійкості сільськогосподарського підприємства. Відстеження та обробка даних про рівень освітленості, забрудненості та якості води. Виявлення потенційних проблем, попередження про можливі захворювання рослин, дефіцит поживних речовин або появу шкідників. Оптимізація роботи зрошувальних систем. Підвищення ефективності збору врожаю на основі використання технологій комп'ютерного зору, машинного навчання та ШІ для ідентифікації дозрілого урожаю та визначення початку його збору.

Додаток Г

Таблиця 1 – Узагальнена характеристика популярних інструментів для роботи з моделями машинного навчання

Назва інструментарію	Короткий опис застосування
<i>Мова програмування Python та бібліотеки для машинного навчання</i>	Python є однією з найпоширеніших мов програмування для розробки моделей машинного навчання, а його бібліотеки (<i>scikit-learn</i> , <i>TensorFlow</i> , <i>PyTorch</i>), забезпечують широкі можливості для створення та оптимізації різноманітних моделей.
<i>Мова програмування R</i>	Мова програмування R разом з пакетами (<i>caret</i> і <i>ggplot2</i>) може бути ефективним вибором для роботи з даними та реалізації моделей, використовуючи статистичний підхід.
<i>AutoML (Automated Machine Learning) інструменти</i>	Використання інструментів AutoML (<i>Google AutoML</i> , <i>H2O.ai</i> , <i>AutoKeras</i>), може спростити процес створення моделей для тих, хто не має глибоких знань у сфері машинного навчання.
<i>Хмарні платформи</i>	Хмарні сервіси (<i>Google Cloud ML Engine</i> , <i>Microsoft Azure ML</i> , <i>Amazon SageMaker</i>), надають інтегровані середовища для розробки, тренування та розгортання моделей без значних витрат на обладнання.
<i>Компоненти відкритого програмного забезпечення</i>	Використання відкритих програмних засобів, наприклад <i>Apache Spark</i> для обробки великих обсягів даних або <i>KNIME</i> для візуального моделювання.

Таблиця 2 – Опис інструментів для роботи з моделями машинного навчання

Назва інструменту	Коротка характеристика
<i>Інструменти відстеження експерименту та керування метаданими моделі</i>	
<i>MLFlow</i>	Інструмент із відкритим кодом, який допомагає керувати основними частинами життєвого циклу машинного навчання. Можна керувати експериментами з машинним навчанням і моделювати метадані за допомогою CLI, Python, R, Java і REST API. MLflow має чотири основні функції: 1. Відстеження MLflow: зберігання та доступ до коду, даних, конфігурації та результатів. 2. Проекти MLflow: наукове джерело пакетних даних для відтворюваності. 3. Моделі MLflow: розгортання та керування моделями машинного навчання в різних середовищах обслуговування. 4. Реєстр моделей MLflow: центральне сховище моделей, яке забезпечує версії, переходи етапів, анотації та керування моделями машинного навчання.
<i>Comet ML</i>	Платформа для відстеження, порівняння, пояснення та оптимізації моделей і експериментів машинного навчання. Можна використовувати з будь-якою бібліотекою машинного навчання, такою як Scikit-learn, PyTorch, TensorFlow і HuggingFace.
<i>Weights & Biases</i>	Платформа ML для відстеження експериментів, версії даних і моделей, оптимізації гіперпараметрів і керування моделлю. Можна використовувати для реєстрації артефактів (наборів даних, моделей, залежностей, конвеєрів і результатів) і візуалізації наборів даних (аудіо, візуальних, текстових і табличних). Інтегрується з іншими бібліотеками машинного навчання, такими як Fastai, Keras, PyTorch, Hugging face.
<i>Інструменти для створення проектів даних і керування робочими процесами машинного навчання</i>	
<i>Prefect</i>	Стек даних для моніторингу, координації та організації робочих процесів між додатками. Легкий інструмент із відкритим кодом, створений для наскрізних конвеєрів машинного навчання.
<i>Metaflow</i>	Автоматично відстежує та версії експериментів і даних машинного навчання. Можна візуалізувати результати в блокноті. Metaflow працює з кількома хмарами (включаючи AWS, GCP і Azure) і різними пакетами Python для машинного навчання (наприклад, Scikit-learn і Tensorflow), а API також доступний для мови R.
<i>Інструменти керування версіями даних і конвеєрів</i>	
<i>Pachyderm</i>	Автоматизує перетворення даних за допомогою версії даних, походження та наскрізних конвеєрів у Kubernetes. Можна інтегрувати будь-які дані (зображення, журнали, відео, файли CSV), будь-яку мову (Python, R, SQL, C/C++) і в будь-якому масштабі.
<i>Data Version Control (DVC)</i>	Бездоганно працює з Git, щоб надати код, дані, модель, метадані та версії конвеєра. Відстеження експерименту (метрики моделі, параметри, версії). Інтеграція та розгортання для машинного навчання.

Узагальнено на основі [47]

Результати оцінки якості даних моделі

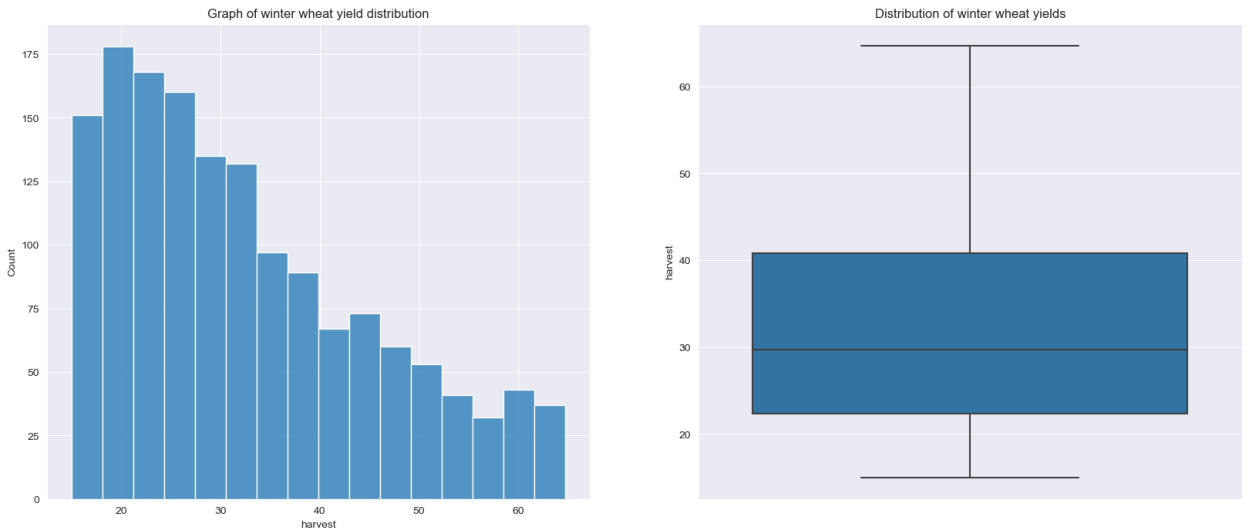


Рисунок 1 – Дослідження якості даних фактичної урожайності

Correlation between different features

```
In [25]: fig, ax = plt.subplots(1, 1, figsize=(17, 8))
sns.heatmap(df.corr(), annot=True)
plt.title('Correlation between different features', fontsize = 15, c='black')
plt.show()
```

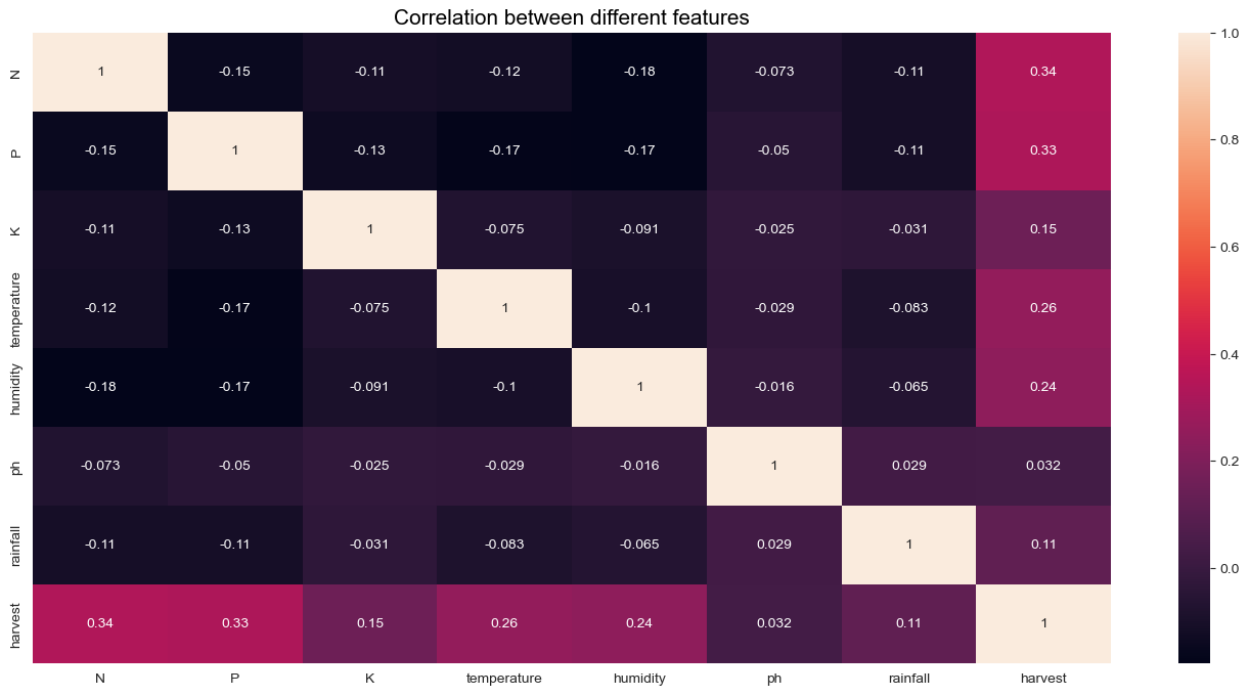


Рисунок 2 – Дослідження зв'язку між ознаками


```
In [28]: from sklearn.model_selection import train_test_split
np.random.seed(0)
df_train, df_test = train_test_split(harvest_winter_wheat, train_size = 0.7, test_size = 0.3, random_state = 100)
```

```
In [29]: from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
num_vars = ['N', 'P', 'K', 'temperature', 'humidity', 'rainfall', 'harvest']
df_train[num_vars] = scaler.fit_transform(df_train[num_vars])
```

```
In [30]: df_train.head()
```

```
Out[30]:
```

	harvest	N	P	K	temperature	humidity	rainfall
345	0.110664	0.037724	0.365975	0.855876	0.6875	0.769050	0.318318
976	0.674044	0.690595	0.070655	0.708812	0.9250	0.902696	0.833500
306	0.899396	0.977458	0.244632	0.399779	0.8875	0.838218	0.550551
355	0.002012	0.115058	0.468292	0.018464	0.7250	0.872216	0.068402
474	0.452716	0.697651	0.708794	0.448553	0.4750	0.595545	0.493827

```
In [31]: df_train.describe()
```

```
Out[31]:
```

	harvest	N	P	K	temperature	humidity	rainfall
count	1061.000000	1061.000000	1061.000000	1061.000000	1061.000000	1061.000000	1061.000000
mean	0.354929	0.516858	0.515232	0.501488	0.502651	0.733339	0.521824
std	0.256537	0.274471	0.277516	0.288678	0.276461	0.188429	0.283484
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.150905	0.280557	0.281302	0.253937	0.275000	0.599062	0.278946
50%	0.295775	0.522392	0.507277	0.502011	0.500000	0.790152	0.534201
75%	0.519115	0.737736	0.749939	0.742485	0.737500	0.892145	0.763764
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

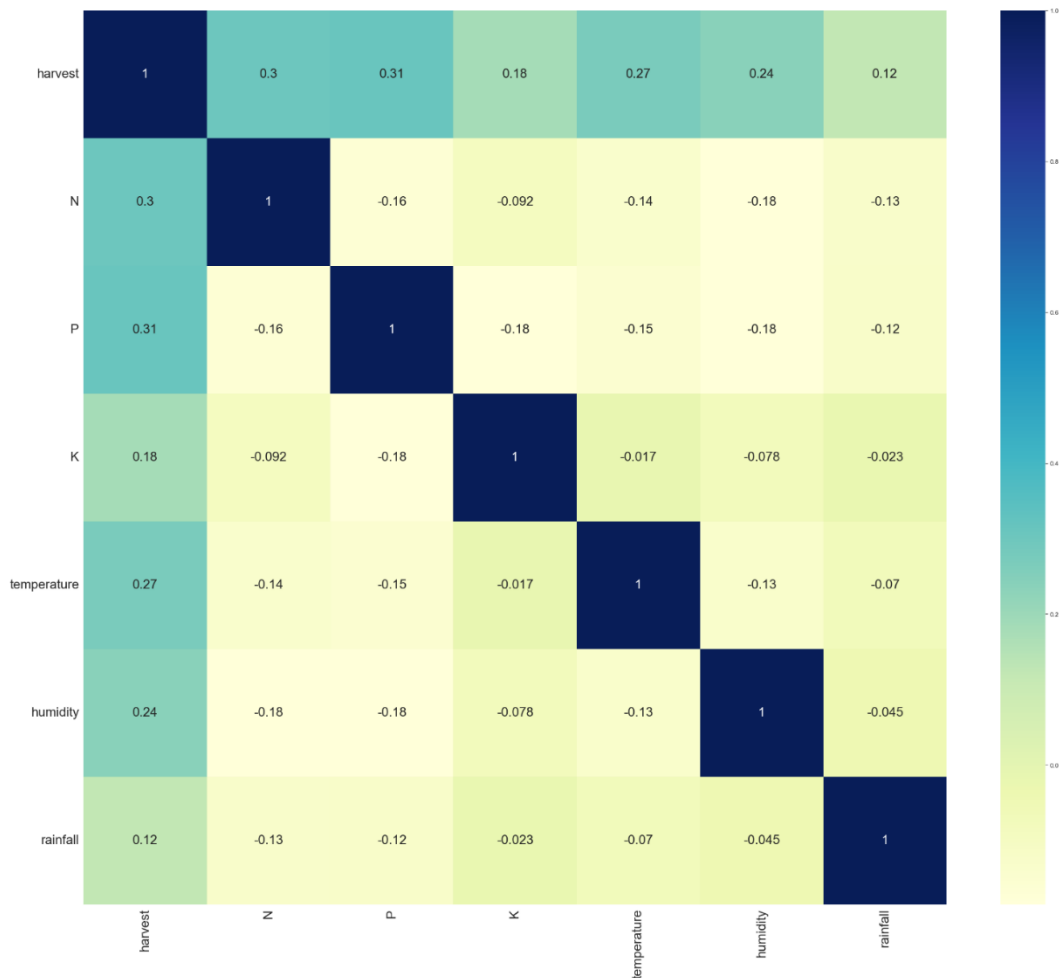


Рисунок 3 – Візуальне представлення виконання досліджень

