

ПОТАПЧУК Артур Сергійович

**Модуль рекомендацій для агрокультур на основі
даних про ґрунт та кліматичні умови з
використанням алгоритмів машинного навчання /
Recommendations module for agriculture based on
soil and climatic data using machine learning
algorithms**

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-41
А. С. Потапчук

Науковий керівник:
к.т.н., доцент, П. Є. Биковий

Кваліфікаційну роботу
допущено до захисту:

" ____ " _____ 20 ____ р.

Завідувач кафедри
_____ **М. П. Комар**

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
« ____ » _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ПОТАПЧУКУ Артуру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Модуль рекомендацій для агрокультур на основі даних про ґрунт та кліматичні умови з використанням алгоритмів машинного навчання / Recommendations module for agriculture based on soil and climatic data using machine learning algorithms

керівник роботи Биковий Павло Євгеновичу к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 12 грудня 2023 р. № 753.

2. Строк подання студентом закінченої кваліфікаційної роботи 15 травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- Провести детальний аналіз агрономічних параметрів.
- Здійснити огляд існуючих рішень у сфері машинного навчання.
- Розробити архітектуру модуля рекомендацій.
- Впровадити ансамблевий підхід з використанням бустингу.
- Інтегрувати і оптимізувати алгоритм Light Gradient Boosting Machine (LightGBM) для використання у модулі рекомендацій.
- Розробити програмно-технологічне забезпечення.
- Провести візуалізацію та аналіз набору даних.
- Оцінити ефективність і точність моделі.

5. Перелік графічного матеріалу в роботі:

- Архітектура модуля рекомендацій на основі машинного навчання
- Алгоритм для аналізу даних про рекомендації щодо вирощування рослин
- графіки з результатами експериментальних досліджень.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2024 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2024 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2024 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2024 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2024 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 20.05.2024 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту у системі «Unichек».	до 10.06.2024 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 14.06.2024 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії.	до 14.06. 2024 р.	

Студент _____ А. С. Потапчук
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ Биковий П.Є.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Модуль рекомендацій для агрокультур на основі даних про ґрунт та кліматичні умови з використанням алгоритмів машинного навчання» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 43 сторінки і містить 19 ілюстрацій, 2 таблиці та 13 використаних джерел.

Метою роботи є розробка модуля рекомендацій для агрокультур, заснованого на аналізі даних про ґрунт та кліматичні умови з використанням алгоритмів машинного навчання.

Методами розроблення обрано метод аналізу (для дослідження існуючих підходів до рекомендаційних систем), метод синтезу (для поєднання переваг існуючих методів), методи моделювання (для представлення та дослідження процесів рекомендацій), метод порівняльного аналізу (для оцінювання адекватності моделі рекомендацій).

Внаслідок виконання роботи обґрунтовано раціональний підхід до розроблення моделей рекомендацій та розроблено програмний засіб, який дозволяє створювати і досліджувати моделі рекомендацій.

Результати дослідження можуть бути використані в науково-дослідних установах і підрозділах підприємств, що займаються розробленням моделей рекомендацій.

Ключові слова: РЕКОМЕНДАЦІЙНІ СИСТЕМИ, АГРОКУЛЬТУРИ, МАШИННЕ НАВЧАННЯ, ҐРУНТОВІ УМОВИ, КЛІМАТИЧНІ ДАНІ.

ANNOTATION

Qualification work on the topic «Recommendation module for agriculture based on soil and climatic data using machine learning algorithms» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 43 pages and it contains 19 figures, 2 tables, and 13 sources.

The purpose of the work is to develop a recommendation module for agriculture based on the analysis of soil and climatic data using machine learning algorithms.

Research methods include analysis (to study existing approaches to recommendation systems), synthesis (to combine the advantages of existing methods), modeling (to represent and study recommendation processes), and comparative analysis (to evaluate the adequacy of the recommendation model).

As a result of the work, a rational approach to the development of recommendation models was substantiated, and a software tool was developed that allows creating and researching recommendation models.

The research results can be used in research institutions and enterprise departments involved in the development of recommendation models.

Keywords: RECOMMENDATION SYSTEMS, CROPS, MACHINE LEARNING, SOIL CONDITIONS, CLIMATIC DATA.

ЗМІСТ

Вступ	7
1 Аналіз предметної області і постановка задачі дослідження	11
1.1 Агрономічні параметри та їх вплив на врожайність	11
1.2 Огляд існуючих рішень	14
1.3 Вибір перспективного шляху і постановка задачі дослідження	18
2 Алгоритмічне забезпечення рекомендацій для агрокультур на основі даних про ґрунт та кліматичні умови з використанням алгоритмів машинного навчання	21
2.1 Архітектура модуля рекомендацій на основі машинного навчання	21
2.2 Бустинговий підхід ансамблевого машинного навчання	23
2.3 Алгоритм Light Gradient Boosting Machine	25
2.4 Алгоритм рекомендацій для агрокультур на основі даних про ґрунт та кліматичні умови з використанням алгоритмів машинного навчання	27
3 Програмно-технологічне забезпечення рекомендацій для агрокультур на основі даних про ґрунт та кліматичні умови з використанням алгоритмів машинного навчання	30
3.1 Опис набору даних	30
3.2 Візуалізація та аналіз даних	32
3.3 Аналіз ефективності моделі LightGBM у класифікації сільськогосподарських культур	40
Висновки	47
Список використаних джерел	49
Додаток А Псевдокод алгоритму	52
Додаток Б Код для реалізації	53
Додаток В Апробація отриманих результатів	62

ВСТУП

Актуальність розробки модуля рекомендацій для агрокультур, заснованого на аналізі даних про ґрунт і кліматичні умови з використанням алгоритмів машинного навчання, обумовлена рядом важливих факторів. В умовах глобальних змін клімату та зростання населення планети забезпечення стабільного і ефективного агропромислового виробництва стає критично важливим. Виклики, пов'язані зі зміною кліматичних умов, зниженням кількості придатних для вирощування земель і підвищенням потреби у продовольстві, вимагають новітніх технологічних рішень.

Застосування машинного навчання у агрономії відкриває нові можливості для оптимізації агрокультурних технологій. Алгоритми, зокрема LightGBM, дозволяють з високою точністю аналізувати великі обсяги даних про ґрунти, кліматичні умови та їх вплив на ріст та розвиток рослин. Це допомагає аграріям приймати обґрунтовані рішення щодо вибору культур, часу посіву, використання добрив та захисту рослин, тим самим підвищуючи ефективність агропромислового виробництва.

Оперативне впровадження інноваційних рішень в агросекторі також сприяє сталому розвитку галузі, мінімізації впливу на довкілля та підвищенню адаптаційних можливостей до змін клімату. Модуль рекомендацій, розроблений на основі машинного навчання, забезпечує реалізацію цих завдань, адже він дозволяє точно прогнозувати потреби культур в залежності від змінних умов і реагувати на них оперативно. Такі системи сприяють зменшенню втрат урожаю, збільшенню продуктивності та зниженню витрат на виробництво.

Крім того, розробка і впровадження такого модуля має велике значення для підвищення конкурентоспроможності аграрного сектору на національному та міжнародному рівнях. Використання передових технологій, які базуються на аналітиці великих даних і машинному навчанні, стає

необхідною умовою для ефективного ведення агробізнесу в сучасних умовах. Такі інновації допомагають не тільки вирішувати практичні завдання оптимізації виробництва, але й сприяють розвитку наукових досліджень у галузі агротехнологій, відкриваючи нові напрями для подальших інновацій.

Метою цієї бакалаврської роботи є розробка та валідація модуля рекомендацій для агрокультур, який використовує алгоритми машинного навчання для аналізу даних про ґрунт та кліматичні умови, з метою оптимізації вирощування різних сільськогосподарських культур. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Провести детальний аналіз агрономічних параметрів і їх впливу на врожайність, визначивши ключові фактори для модуля рекомендацій.
2. Здійснити огляд існуючих рішень у сфері машинного навчання для агрокультур, аналізуючи їхні переваги та недоліки.
3. Розробити архітектуру модуля рекомендацій, що використовує машинне навчання для аналізу даних про ґрунт і клімат.
4. Впровадити ансамблевий підхід з використанням бустингу для підвищення точності прогнозування.
5. Інтегрувати і оптимізувати алгоритм Light Gradient Boosting Machine (LightGBM) для використання у модулі рекомендацій.
6. Розробити програмно-технологічне забезпечення для реалізації алгоритму рекомендацій, включаючи збір і обробку даних.
7. Провести візуалізацію та аналіз набору даних для оцінювання агрономічних умов і адаптації рекомендацій.
8. Оцінити ефективність і точність моделі LightGBM у класифікації сільськогосподарських культур на основі зібраних даних і виконати коригування моделі залежно від отриманих результатів.

Об'єктом дослідження є процес визначення оптимальних умов для вирощування агрокультур на основі аналізу даних про ґрунт та кліматичні умови.

Предметом дослідження є алгоритми машинного навчання, зокрема LightGBM, що застосовуються для розробки модуля рекомендацій для агрокультур.

Методи дослідження у цій бакалаврській роботі включають застосування алгоритмів машинного навчання, аналітичні та статистичні методи для обробки та аналізу великих наборів даних. Основу методології складає використання бустингових технік, зокрема алгоритму Light Gradient Boosting Machine (LightGBM), який дозволяє моделювати взаємозв'язки між агрономічними параметрами і врожайністю культур. Крім технічних методів, використовуються методи візуалізації даних для кращого розуміння тенденцій і шаблонів, а також методи оцінювання якості моделі, такі як перехресна перевірка (cross-validation) та аналіз залишків, що забезпечують достовірність і надійність результатів прогнозування.

Практичне значення розробленого модуля рекомендацій для агрокультур важко переоцінити, оскільки воно спрямоване на підвищення продуктивності і сталості агропромислового виробництва. Інструмент дозволить аграріям ефективніше планувати посівні площі, оптимально використовувати добрива та захисні засоби, адаптуватися до змін клімату та мінімізувати ризики неурожаю. Впровадження таких технологій сприятиме зниженню екологічного навантаження від агроіндустрії, збільшенню економічної ефективності фермерських господарств та забезпеченню продовольчої безпеки регіону. Таким чином, модуль стане значущим внеском у розвиток сучасних агротехнологій, спрямованих на створення розумного сільського господарства.

Структура і розмір дослідження. Кваліфікаційна робота складається із вступу, трьох розділів, висновків та списку використаних джерел. Загальний обсяг роботи складає 43 сторінки тексту, що включають 19 рисунків і 2 таблиці. У списку використаних джерел зазначено 13 джерел, які займають 2 сторінки.

Апробація результатів дослідження. Основні теоретичні положення роботи й практичні результати дослідження доповідалися й обговорювалися на VI Міжнародної мультидисциплінарної студентської наукової конференції «Розвиток суспільства та науки в умовах цифрової трансформації», яка відбулася 31 травня 2024 року у місті Івано-Франківськ, Україна.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Агрономічні параметри та їх вплив на врожайність

У сучасному аграрному виробництві величезне значення набуває глибоке розуміння агрономічних параметрів, які безпосередньо впливають на ріст, розвиток та врожайність сільськогосподарських культур. Агрономічні параметри (Рисунок 1.1), такі як вміст азоту, фосфору та калію в ґрунті, температура, вологість, кислотність ґрунту (рН) та кількість опадів, відіграють ключову роль у визначенні продуктивності агросистем. Відповідне управління цими параметрами може значно підвищити ефективність використання ресурсів, оптимізувати умови для росту рослин і збільшити вихід продукції на одиницю площі.



Рисунок 1.1 - Агрономічні параметри

Детальне вивчення цих факторів важливе не лише для забезпечення високого урожаю, а й для забезпечення стійкості агроєкосистем до змін клімату та інших екологічних викликів. Наприклад, адаптація до мінливих умов вологості та температури вимагає від фермерів точного знання про взаємодію цих параметрів з конкретними типами культур. Також, розуміння

кислотності ґрунту та її впливу на мікроелементний баланс дозволяє ефективніше планувати заходи щодо внесення добрив.

Азот є одним з найважливіших макроелементів для рослинного росту та розвитку. Він є ключовим компонентом хлорофілу, білків та амінокислот, що є основою для росту рослинної тканини. Відсутність достатньої кількості азоту в ґрунті може призвести до хлорозу (пожовтіння листя), що є ознакою дефіциту цього елемента. Важливо збалансувати кількість азоту, оскільки його надлишок може спричинити швидке зростання зелені на шкоду розвитку плодів або коренів.

Фосфор критично важливий для енергетичного обміну в рослинах, включаючи процеси фотосинтезу та дихання. Він є складовою частиною АТФ, необхідної молекули для зберігання та передачі енергії в клітинах. Фосфор також впливає на ріст коренів і зрілість рослин. Недолік фосфору може проявлятися в затримці росту, темно-зелених або фіолетових листках і слабкому цвітінні.

Калій надзвичайно важливий для регулювання водного балансу рослин та активізації численних ферментів. Він сприяє фотосинтезу, білковому синтезу та регулюванню осмотичного тиску. Калій підвищує стійкість рослин до стресів, таких як посуха, холод та хвороби. Його недолік може призвести до хлорозу листя, в'янення та браку стійкості до стресових умов.

Температура впливає на швидкість метаболічних процесів у рослинах та може визначати сезонність їх росту та розвитку. Кожна рослина має певний оптимальний температурний діапазон, що сприяє найкращому зростанню. Екстремальні температури, як занадто високі, так і занадто низькі, можуть сповільнити ріст рослин, знизити їх продуктивність або навіть спричинити пошкодження тканин.

Вологість впливає на процеси транспірації в рослинах та може впливати на їх терморегуляцію та газообмін. Надмірна або недостатня вологість може призвести до розвитку хвороб, зокрема грибкових. Вологість також впливає на

вбирання рослинами води та поживних речовин з ґрунту, особливо важливо це в періоди інтенсивного росту.

pH ґрунту є важливим фактором, що впливає на доступність поживних речовин для рослин. Він визначає, наскільки ефективно рослини можуть абсорбувати різні макро- та мікроелементи з ґрунту. Кожен тип рослин має свій оптимальний рівень pH, при якому вони найефективніше засвоюють необхідні елементи. Неправильний рівень pH може блокувати засвоєння певних речовин, що призводить до дефіциту харчування.

Опади є первинним джерелом води для багатьох сільськогосподарських систем і впливають на вологість ґрунту та доступність води для рослин. Регулярні та достатні опади важливі для підтримки здорового росту рослин, особливо в регіонах, де зрошення недоступне або обмежене. Проте, надмірні опади можуть спричинити ерозію ґрунту, вимивання поживних речовин і навіть затоплення посівів.

Розуміння агрономічних параметрів та їхній вплив на врожайність є фундаментальним для розвитку ефективних агротехнологій. Вміст азоту, фосфору та калію в ґрунті, а також контроль над температурними умовами, вологістю, рівнем pH та кількістю опадів визначає здоров'я та продуктивність рослин. Ці параметри впливають не тільки на поточний стан рослин, а й на їхню здатність адаптуватися до змінних умов, забезпечуючи стійкість у довгостроковій перспективі. Ефективне управління цими параметрами може значно збільшити врожайність, оптимізувати використання ресурсів і підвищити стійкість агроєкосистем до екологічних викликів. Таким чином, глибоке знання та інтеграція цих агрономічних параметрів у практичне сільськогосподарське виробництво є ключем до успіху в сучасному аграрному секторі.

1.2 Огляд існуючих рішень

Застосування алгоритмів машинного навчання, зокрема LightGBM, революціонізує сільськогосподарську індустрію, забезпечуючи нові можливості для підвищення продуктивності та ефективності аграрного сектора. Особливий інтерес становить використання цих технологій для створення передових систем рекомендацій, які дозволяють адаптувати агротехнічні прийоми до змінних умов вирощування культур і кліматичних факторів. У цьому контексті, розгляд різних досліджень, які імплементують LightGBM для вирішення специфічних завдань, таких як вибір оптимальних культур, дозування добрив, а також прогнозування агрокліматичних умов, набуває актуальності. Ці дослідження не лише демонструють технічні аспекти використання алгоритмів, але й підкреслюють їхню практичну значущість у реальних умовах господарювання.

У дослідженні [1] (Mukherjee et al., 2022) розглядається використання різних методів машинного навчання для створення системи рекомендацій добрив для сільськогосподарських культур. Зокрема, було встановлено, що алгоритм LightGBM показав найвищу точність серед інших алгоритмів, що використовуються у дослідженні. Автори підкреслюють важливість точних рекомендацій для покращення продуктивності та ефективності сільськогосподарських систем. Дослідження також розглядає вплив різних параметрів ґрунту та кліматичних умов на вибір відповідних культур для вирощування, що допомагає зменшити ризики втрат врожаю.

Дослідження [2] (Faizanuddin et al., 2023) описує розробку системи рекомендацій для вибору сільськогосподарських культур в Індії, використовуючи алгоритм LightGBM. Автори тестували різні алгоритми і встановили, що LightGBM забезпечує найвищу точність у передбаченні відповідних культур. Це дослідження підкреслює значення машинного навчання у підвищенні ефективності сільського господарства і пропонує рішення для оптимізації використання ресурсів та збільшення врожайності.

У дослідженні [3] (Dutta et al., 2023) розглядається використання алгоритму LightGBM для рекомендацій фруктових культур. Система аналізує різні агрокліматичні показники і передбачає найкращі варіанти фруктових культур для вирощування. Автори відзначають високу точність та ефективність алгоритму у врахуванні численних факторів, що впливають на врожайність і якість фруктів, що значно покращує процес прийняття рішень у садівництві.

Дослідження [4] (Yan et al., 2021) фокусується на використанні алгоритму LightGBM для геномного відбору в селекції кукурудзи. Автори аналізували точність, стабільність моделі та ефективність обчислень, встановлюючи, що LightGBM є потужним інструментом для передбачення генетичних характеристик культур. Це дослідження підкреслює важливість використання передових методів машинного навчання для підвищення ефективності селекційних програм.

У статті [5] (Mao et al., 2024) описується нова модель прогнозування клімату в теплиці, яка використовує комбінацію BiGRU-Attention та LightGBM. Дослідження показує, що ця модель ефективно передбачає кліматичні умови, враховуючи вплив на врожайність культур у закритому ґрунті. Автори відзначають переваги LightGBM у підвищенні точності прогнозів та адаптації до змінних умов середовища.

Дослідження [7] (Kulkarni et al., 2022) аналізує впровадження штучного інтелекту в аграрному секторі, зокрема використання LightGBM для моніторингу ґрунтових поживних речовин та рекомендацій щодо вибору культур. Автори досліджують ефективність цього алгоритму у прогнозуванні оптимальних культур на основі даних про ґрунт та кліматичні умови. Вони також підкреслюють важливість впровадження таких систем для підвищення продуктивності сільськогосподарських угідь.

У дослідженні [8] (Shams et al., 2024) розглядається впровадження XAI-CROP для покращення систем рекомендацій культур. Використання

LightGBM забезпечує високу точність і стабільність моделі. Автори акцентують увагу на важливості пояснюваних моделей для покращення прийняття рішень у сільському господарстві та зниження ризиків, пов'язаних із змінами клімату.

Дослідження [9] (Hasan et al., 2023) описує систему рекомендацій для вибору сільськогосподарських культур, що базується на використанні ансамблевих методів машинного навчання, включаючи LightGBM. Автори підкреслюють ефективність цієї моделі у підвищенні точності прогнозування та оптимізації використання ресурсів.

У дослідженні [10] (Kumari & Sambhav, 2023) розглядається розробка системи рекомендацій культур з використанням LightGBM. Автори аналізують вплив різних параметрів, таких як поживні речовини в ґрунті та кліматичні умови, на ефективність вибору культур. Вони підкреслюють переваги LightGBM у підвищенні точності прогнозування та стійкості аграрних систем.

Дослідження [11] (Halbe et al., 2023) описує розробку інтелектуальної системи рекомендацій щодо вибору культур та добрив з використанням алгоритму LightGBM. Автори досліджують ефективність цієї системи у покращенні рішень щодо сільськогосподарських культур, враховуючи різні агрокліматичні показники та поживні речовини ґрунту.

Сучасне сільське господарство швидко трансформується завдяки застосуванню передових технологій машинного навчання, які сприяють підвищенню продуктивності та оптимізації ресурсів. Одним із ключових напрямків є розробка систем рекомендацій для вибору культур, добрив та умов вирощування, що базуються на алгоритмі LightGBM. Цей огляд має на меті аналізувати як різні дослідження впроваджують LightGBM для рішення проблем агрономії та які переваги це приносить. Таблиця 1.1 забезпечує порівняння п'яти сучасних досліджень, що використовують LightGBM у сільському господарстві.

Таблиця 1.1 – Порівняння найближчих досліджень до даного

Автори	Мета дослідження	Основні результати	Практичні застосування
Mukherjee et al. [1]	Рекомендації добрив для сільськогосподарських культур	Висока точність LightGBM порівняно з іншими алгоритмами	Вибір культур залежно від ґрунту та клімату
Faizanuddin et al. [2]	Система рекомендацій культур в Індії	Найвища точність LightGBM у передбаченні культур	Оптимізація використання ресурсів, збільшення врожайності
Dutta et al. [3]	Рекомендації фруктових культур	Ефективність алгоритму у передбаченні врожайності та якості фруктів	Покращення прийняття рішень у садівництві
Yan et al. [4]	Геномний відбір у селекції кукурудзи	Висока точність, стабільність моделі, ефективність обчислень	Підвищення ефективності селекційних програм
Mao et al. [5]	Прогнозування клімату в теплиці	Ефективне прогнозування кліматичних умов	Адаптація до змінних умов середовища, вплив на врожайність

Аналіз представлених досліджень показує, що алгоритм LightGBM демонструє значну ефективність у прогнозуванні відповідних агрокліматичних умов для вирощування різноманітних культур. Використання цього алгоритму сприяє точному аналізу великих обсягів даних, забезпечуючи високу швидкість обчислень та зниження помилок у рекомендаціях, що особливо важливо для динамічно змінних умов сучасного аграрного виробництва. Відмінності між дослідженнями також підкреслюють різноманітність застосування LightGBM, від геномного відбору культур до управління кліматичними умовами в теплицях.

На відміну від розглянутих досліджень, наше власне дослідження впроваджує більш широкий спектр аналітичних інструментів поряд із LightGBM, включаючи глибші нейронні мережі та методи пояснювального штучного інтелекту (XAI), що дозволяє не тільки забезпечити високу точність передбачень, але й зробити процес рішення прозорішим та зрозумілішим для

кінцевого користувача. Це сприяє кращому прийняттю технологій кінцевими користувачами, забезпечуючи більш ефективне і стає сільське господарство.

1.3 Вибір перспективного шляху і постановка задачі дослідження

У сучасних умовах глобалізації та зміни клімату сільське господарство стикається з численними викликами, які вимагають інноваційного підходу до управління агропромисловими процесами. Забезпечення сталого розвитку агросектору і збільшення врожайності при одночасному зниженні впливу на довкілля є однією з ключових проблем. В цьому контексті, розвиток і впровадження передових інформаційних систем, заснованих на аналізі великих даних про ґрунт та кліматичні умови через машинне навчання, набуває особливої актуальності.

Основна мета використання алгоритмів машинного навчання у аграрному секторі полягає у створенні ефективних моделей для точного прогнозування ідеальних умов для культивування різних культур. Завдяки аналізу даних про ґрунт і клімат, можна не тільки підбирати оптимальні види культур для конкретних умов, але й точно розраховувати необхідні дози добрив та інших агрохімічних засобів захисту рослин. Це, у свою чергу, сприяє зниженню витрат на агрохімії та підвищенню екологічності агровиробництва.

Впровадження модулів рекомендацій, заснованих на машинному навчанні, дозволяє агровиробникам адаптуватися до змінюваних кліматичних умов шляхом оперативного реагування на зміни температурних режимів та вологості. Це особливо актуально в умовах глобального потепління, коли традиційні підходи до ведення сільського господарства можуть бути неефективними або навіть шкідливими для довкілля.

Наукові дослідження у сфері машинного навчання відкривають нові перспективи для оптимізації сільськогосподарських технологій. Використання складних алгоритмів, таких як LightGBM, дозволяє аналізувати

великі набори даних з високою точністю і швидкістю, що є вирішальним для вчасного ухвалення рішень. Крім того, ці алгоритми можуть постійно оновлюватись та навчатись на нових даних, що забезпечує їхню актуальність та адаптивність до постійно змінних умов господарювання.

Таким чином, розвиток і застосування модулів рекомендацій для агрокультур на основі даних про ґрунт і клімат є критично важливим для досягнення цілей сталого розвитку в аграрному секторі. Ці технології не тільки сприяють збільшенню врожайності і покращенню якості агропродукції, але й відіграють ключову роль у забезпеченні продовольчої безпеки на глобальному рівні.

Метою цієї бакалаврської роботи є розробка та валідація модуля рекомендацій для агрокультур, який використовує алгоритми машинного навчання для аналізу даних про ґрунт та кліматичні умови, з метою оптимізації вирощування різних сільськогосподарських культур. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Провести детальний аналіз агрономічних параметрів і їх впливу на врожайність, визначивши ключові фактори для модуля рекомендацій.
2. Здійснити огляд існуючих рішень у сфері машинного навчання для агрокультур, аналізуючи їхні переваги та недоліки.
3. Розробити архітектуру модуля рекомендацій, що використовує машинне навчання для аналізу даних про ґрунт і клімат.
4. Впровадити ансамблевий підхід з використанням бустингу для підвищення точності прогнозування.
5. Інтегрувати і оптимізувати алгоритм Light Gradient Boosting Machine (LightGBM) для використання у модулі рекомендацій.
6. Розробити програмно-технологічне забезпечення для реалізації алгоритму рекомендацій, включаючи збір і обробку даних.
7. Провести візуалізацію та аналіз набору даних для оцінювання агрономічних умов і адаптації рекомендацій.

8. Оцінити ефективність і точність моделі LightGBM у класифікації сільськогосподарських культур на основі зібраних даних і виконати коригування моделі залежно від отриманих результатів.

Завдання 1-2 будуть розглянуті в теоретичних розділах роботи, а завдання 3-8 – у практичних розділах, що охоплюють розробку, оцінку та тестування алгоритму.

2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ РЕКОМЕНДАЦІЙ ДЛЯ АГРОКУЛЬТУР НА ОСНОВІ ДАНИХ ПРО ҐРУНТ ТА КЛІМАТИЧНІ УМОВИ З ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

2.1 Архітектура модуля рекомендацій на основі машинного навчання

Модуль (Рисунок 2.1) рекомендацій для агрокультур аналізує дані про ґрунт та кліматичні умови для того, щоб рекомендувати оптимальні культури для вирощування в конкретних умовах. Використовуючи алгоритми машинного навчання, цей модуль допомагає агрономам приймати рішення, засновані на даних, з метою підвищення урожайності та стійкості культур до несприятливих умов.

Компоненти системи:

1. Інтерфейс користувача (UI): Інтерфейс, через який користувачі вводять дані про ґрунт, клімат та інші відповідні параметри.
2. Модуль збору даних: Збирає вхідні дані та обробляє їх перед передачею в модуль аналізу.
3. Система управління базами даних (DBMS): Зберігає історичні дані про врожаї, ґрунти, кліматичні умови та результати попередніх аналізів.
4. Модуль аналізу даних: Виконує обробку та аналіз даних, використовуючи статистичні методи та машинне навчання для виявлення закономірностей та тенденцій.
5. Модуль машинного навчання: Використовує алгоритми машинного навчання, такі як бустинг або нейронні мережі, для прогнозування оптимальних культур на основі аналізу даних.
6. Модуль рекомендацій: Генерує рекомендації на основі виводів модулю аналізу та прогнозів модулю машинного навчання.
7. API для інтеграції: Дозволяє іншим системам та додаткам інтегрувати та використовувати функціональність модуля рекомендацій.

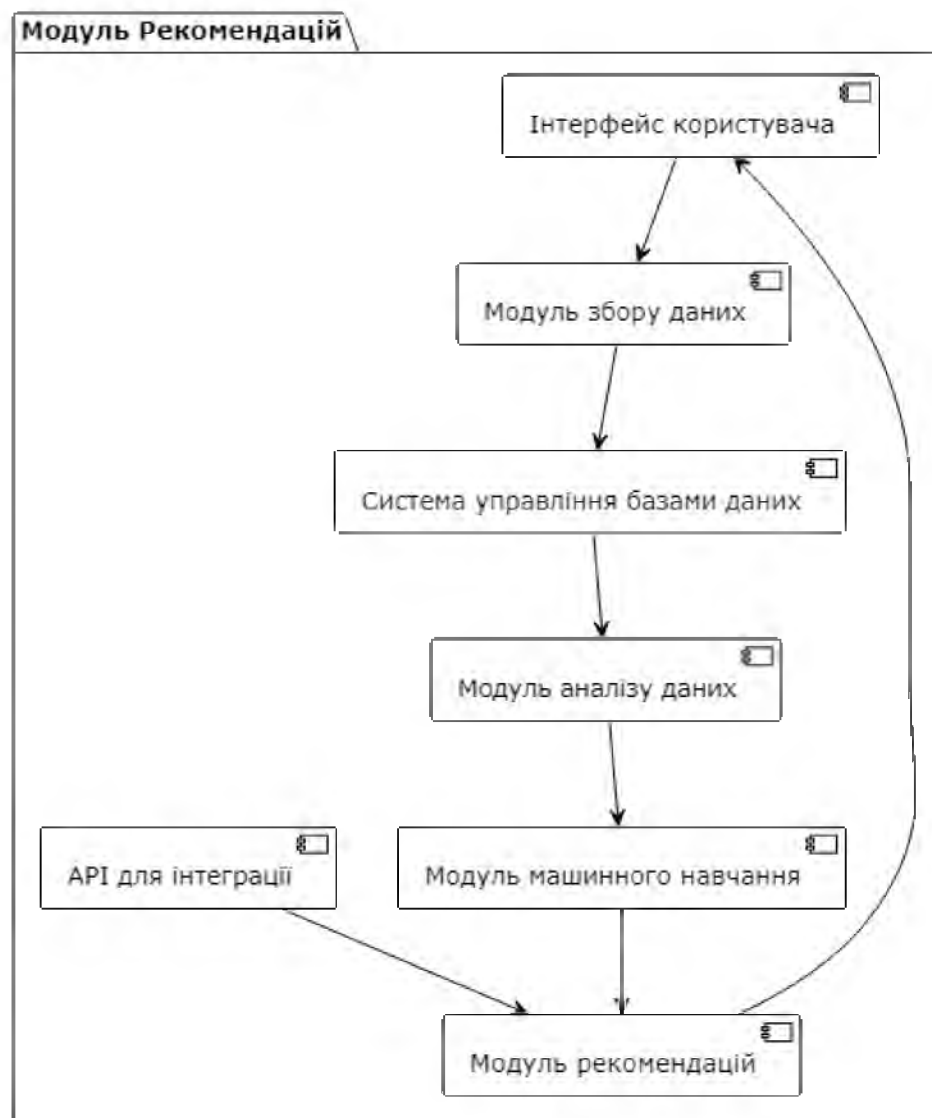


Рисунок 2.1 - Архітектура модуля рекомендацій на основі машинного навчання

Послідовність роботи:

1. Користувач вводить дані через UI.
2. Дані передаються в модуль збору даних.
3. Оброблені дані зберігаються в DBMS.
4. Модуль аналізу обробляє дані та передає результати в модуль машинного навчання.
5. Модуль машинного навчання генерує прогнози та передає їх у модуль рекомендацій.

6. Модуль рекомендацій видає кінцеві рекомендації користувачу.
7. API забезпечує інтеграцію з іншими системами.

Архітектура модуля рекомендацій для агрокультур, заснована на аналізі даних про ґрунт і кліматичні умови через алгоритми машинного навчання, представляє собою комплексну систему для підтримки рішень у сфері агрономії. Використання передових технологій, зокрема, машинного навчання, дозволяє не тільки оптимізувати вибір культур під конкретні умови, але й забезпечує можливість адаптуватися до змінюваних кліматичних умов, що є ключовим для стійкого розвитку сільськогосподарського виробництва. Ця система сприяє ефективному використанню ресурсів, підвищенню урожайності, а також зниженню ризиків, пов'язаних із несприятливими умовами вирощування. Завдяки інтеграції з іншими системами через API, модуль може легко вписуватися в існуючі агротехнологічні рішення, забезпечуючи гнучкість і масштабованість впровадження. Таким чином, розробка та впровадження таких систем є важливим кроком у напрямку технологічної модернізації аграрного сектора.

2.2 Бустинговий підхід ансамблевого машинного навчання

Бустинг — це метод ансамблевого машинного навчання, який використовується для покращення стабільності та точності моделей машинного навчання. Він базується на ідеї комбінування слабких моделей-предикторів для створення більш міцного та точного ансамблю. Основна мета бустингу — систематично вдосконалювати прогнози шляхом послідовного навчання моделей на різних вибірках даних.

Етапи бустингу:

1. Ініціалізація: Бустинг починається з ініціалізації базового класифікатора, який навчається на весь набір даних або на підмножину даних. Цей класифікатор зазвичай є досить слабким, що

означає, що його точність вища за випадковий вибір, але все ж недостатня для комплексних задач.

На початку процесу бустингу ініціалізується базовий класифікатор $f_0(x)$. Це може бути дуже простий класифікатор, наприклад, рішення, яке базується на одній змінній або невелике дерево рішень. Ініціалізаційна модель визначає початкове наближення для цільової функції.

2. Ваги для даних: Після навчання першого класифікатора, бустинговий алгоритм змінює ваги тренувальних даних. Об'єкти, які були неправильно класифіковані, отримують більшу вагу, в той час як ваги правильно класифікованих об'єктів зменшуються. Це змушує наступний класифікатор зосередитися на тих прикладах, які були неправильно класифіковані попередньою моделлю.

На кожному етапі t , після навчання класифікатора f_t , обчислюється вектор ваг w_i для кожного тренувального прикладу i , який базується на помилках попереднього класифікатора. Наприклад, в алгоритмі AdaBoost ваги обчислюються так:

$$w_i^{(t+1)} = w_i^{(t)} \cdot \exp(-\alpha_t \cdot y_i \cdot f_t(x_i))$$

де α_t — це коефіцієнт, який пропорційний до точності класифікатора на поточному кроці, а y_i — істинні мітки.

3. Послідовне додавання моделей: Нові моделі додаються одна за одною. Кожна наступна модель навчається з урахуванням попередніх помилок, а її вплив на загальний ансамбль визначається на основі точності її прогнозів. Всі моделі працюють разом, де кожна наступна модель виправляє помилки попередніх.

Кожен наступний класифікатор f_{t+1} навчається з використанням змінених ваг w_i , що змушує його фокусуватися на тих прикладах, які були неправильно класифіковані попередніми моделями. Це послідовне уточнення продовжується до досягнення заданої кількості ітерацій або до того моменту, коли помилки будуть нижче певного порога.

4. Зважене голосування або сума: Після того як всі моделі навчені, бустинг використовує їх прогнози, зважуючи їх на основі точності кожної моделі. Прогнози можуть бути комбіновані за допомогою методів зваженого голосування (для класифікації) або зваженої суми (для регресії).

Фінальний прогноз моделі ансамблю отримується шляхом агрегації виходів окремих моделей, які зважуються на їхній внесок у загальну точність:

$$F(x) = \sum_{t=1}^T \alpha_t f_t(x)$$

де T — кількість класифікаторів у моделі.

Цей підхід дозволяє досягти вищої точності, ніж будь-який із класифікаторів окремо, оскільки помилки одного класифікатора можуть бути компенсовані правильними прогнозами інших.

2.3 Алгоритм Light Gradient Boosting Machine

LightGBM (Light Gradient Boosting Machine) є ефективним алгоритмом градієнтного бустингу, розробленим Microsoft для високої швидкості виконання і низького використання пам'яті. Він особливо популярний у задачах з великими обсягами даних і вимогами до високої продуктивності обчислень.

В LightGBM, модель f є сумою ансамблю дерев рішень:

$$f(x) = \sum_{i=1}^N h_i(x)$$

де $h_i(x)$ представляє i -те дерево рішень, а N – кількість дерев.

Для тренування моделі, LightGBM мінімізує відповідну функцію втрат. Наприклад, для задачі класифікації, часто використовується логістична функція втрат:

$$L(y, f(x)) = \sum_{i=1}^m \log(1 + \exp(-y_i f(x_i)))$$

де y_i – істинні мітки класів, а x_i – характеристики.

На відміну від інших алгоритмів, які перебирають кожну характеристику для визначення найкращих точок розбиття, LightGBM використовує методи на основі гістограм, що значно прискорює процес тренування. LightGBM агрегує значення в точки даних у дискретні кошики, які представляють гістограму, а потім використовує ці гістограми для оптимізації розбиття в деревах.

LightGBM використовує такі покращення:

- GOSS (Gradient-based One-Side Sampling): для зменшення кількості даних без значної втрати точності, шляхом пріоритезації тих зразків, що мають великий градієнт.
- EFB (Exclusive Feature Bundling): для зменшення кількості характеристик, групуючи взаємовиключні характеристики в одну, що зменшує витрати пам'яті і збільшує швидкість.

Таким чином, LightGBM представляє собою потужний інструмент для обробки великих датасетів з високою продуктивністю, що робить його вибором для багатьох промислових застосувань у сфері машинного навчання.

2.4 Алгоритм рекомендацій для агрокультур на основі даних про ґрунт та кліматичні умови з використанням алгоритмів машинного навчання

У сучасному світі агрономії велике значення має використання даних для оптимізації процесів вирощування рослин. Аналіз даних дозволяє глибше зрозуміти потреби різних культур у мінералах та оптимальних умовах для їх росту, що, в свою чергу, сприяє підвищенню урожайності та стійкості рослин до несприятливих умов. Використання таких технологій, як Python та різноманітні бібліотеки для обробки даних та візуалізації, відкриває перед агрономами нові можливості для проведення глибокого аналізу та прийняття обґрунтованих рішень. Цей підхід не тільки сприяє ефективному управлінню ресурсами, але й допомагає адаптуватися до змін клімату, зберігаючи при цьому стабільність виробництва продуктів харчування.

Покроковий алгоритм для аналізу даних про рекомендації щодо вирощування рослин, використовуючи Python та різні бібліотеки для обробки даних та візуалізації (Рисунок 2.1):

1. Імпорт бібліотек: Завантажте необхідні бібліотеки (pandas, numpy, matplotlib, seaborn, plotly).
2. Завантаження даних: Завантажте набір даних, використовуючи `pandas.read_csv()` і покажіть перші кілька рядків, щоб перевірити, чи все завантажено коректно.
3. Перевірка на наявність пропущених значень: Використовуйте `isnull()` та `any()` для перевірки наявності пропущених даних у наборі.
4. Візуалізація даних: Створіть візуалізації за допомогою `matplotlib`, `seaborn` та `plotly`, щоб зрозуміти розподіл та кореляції між різними ознаками. Відобразіть вміст нітрогену, фосфору, калію та інші характеристики за різними культурами.
5. Обробка даних: Поділіть дані на незалежні (X) та залежні (y) змінні. Поділіть набір даних на тренувальні та тестові дані.

6. Моделювання: Використовуйте модель, наприклад LightGBM, для тренування на тренувальних даних.



Рисунок 2.1 - Алгоритм для аналізу даних про рекомендації щодо вирощування рослин

7. Прогнозування та оцінка моделі: Використовуйте модель для прогнозування на тестових даних і оцініть точність моделі за допомогою метрик, таких як точність класифікації та матриця помилок.
8. Відображення результатів: Покажіть результати класифікації та інші візуалізації для кращого розуміння результатів моделі.
9. Аналіз результатів: Проведіть подальший аналіз результатів, зокрема порівняння важливості ознак та перевірку моделі на перенавчання.

Цей алгоритм допомагає зрозуміти та проаналізувати, як різні фактори впливають на вибір культур для вирощування, базуючись на їх вимогах до мікроелементів та умов зростання.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ РЕКОМЕНДАЦІЙ ДЛЯ АГРОКУЛЬТУР НА ОСНОВІ ДАНИХ ПРО ҐРУНТ ТА КЛІМАТИЧНІ УМОВИ З ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

3.1 Опис набору даних

Для аналізу агрономічних даних була використана база даних [5], що містить інформацію про вміст азоту (N), фосфору (P) та калію (K) у ґрунті, а також про температуру, вологість, рН ґрунту та кількість опадів. Ці дані були використані для рекомендації вирощування певних культур відповідно до зазначених умов. Загалом в датасеті представлено 2200 записів, що охоплюють 22 види культур, такі як рис, кукурудза, банани, апельсини та інші.

Аналіз даних розпочався з перевірки на наявність пропущених значень, які виявилися відсутніми у всіх полях датасету, що свідчить про високу якість даних для подальших аналізів. Основні статистичні характеристики кожної з культур, такі як середні показники вмісту N, P, K, температури, вологості, рН та опадів, були виведені у вигляді сводної таблиці (Таблиця 3.1).

Таблиця 3.1 – Середні значення по параметрах

label	K	N	P	humidity	ph	rainfall	temperature
apple	199,89	20,8	134,22	92,33	5,93	112,65	22,63
banana	50,05	100,23	82,01	80,36	5,98	104,63	27,38
blackgram	19,24	40,02	67,47	65,12	7,13	67,88	29,97
chickpea	79,92	40,09	67,79	16,86	7,34	80,06	18,87
coconut	30,59	21,98	16,93	94,84	5,98	175,69	27,41

Таблиця 3.1 представляє агреговані середні значення ключових параметрів ґрунту та кліматичних умов для п'яти видів сільськогосподарських культур. Аналізуючи ці дані, ми можемо зробити певні висновки про агрономічні умови, які оптимальні для кожної з цих культур. Наприклад,

високий вміст калію (K) у ґрунті для яблук (199,89 кг/га) може свідчити про потребу цієї культури в калії для підтримки здоров'я рослин та поліпшення якості плодів. Водночас значення фосфору (P) та азоту (N) для яблук також перевищують середні показники для інших культур, підкреслюючи важливість цих елементів у вирощуванні яблунь.

Для культур, таких як банан, значення азоту (N) є найвищим серед розглянутих, сягаючи 100,23 кг/га, що підкреслює високі потреби бананів у азоті для підтримки швидкого росту та розвитку великих зелених листків. Калій для бананів також є важливим (50,05 кг/га), оскільки він сприяє водному балансу та поліпшує якість плодів. Температурні умови для бананів (27,38°C) та вологість (80,36%) також ідеально підходять для тропічних умов, в яких ця культура традиційно вирощується.

У випадку чорного граму (blackgram) відзначаються значно нижчі показники калію та фосфору, з середнім значенням азоту. Цікаво, що чорний грам має значно вищий показник рН (7,13), що є більш алкалічним порівняно з іншими культурами. Це може свідчити про потребу цієї культури в менш кислих ґрунтах. Відносно низька вологість (65,12%) та висока температура (29,97°C) також вказують на здатність цієї культури адаптуватися до більш сухих та теплих умов.

Нагортаючи на горох (chickpea), спостерігаємо високий рівень фосфору та калію порівняно з азотом, а також дуже низьку вологість (16,86%), що підкреслює його здатність вирощуватися в аридних умовах. Високий рівень рН (7,34) також вказує на алкалічність ґрунту, яка є підходящою для гороху.

Остання розглянута культура — кокос, демонструє високі вимоги до вологості (94,84%), що є характерним для тропічних регіонів. Вміст калію, фосфору та азоту відносно низький, що може свідчити про низькі поживні потреби цієї культури. Високі значення опадів (175,69 мм) та температура (27,41°C) також відповідають кліматичним умовам, типовим для вирощування кокоса.

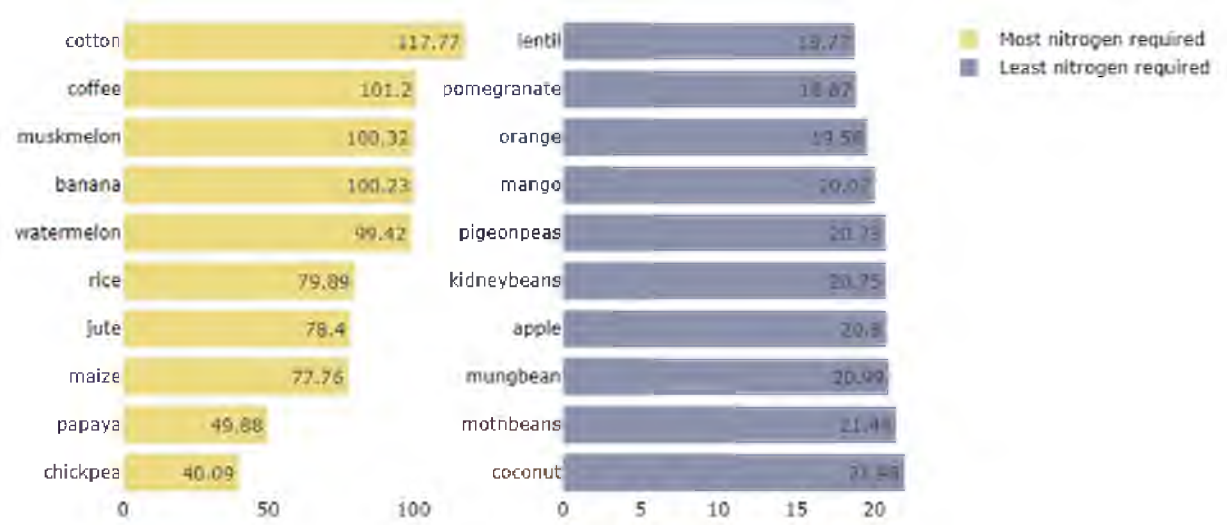
3.2 Візуалізація та аналіз даних

У сучасному сільському господарстві ефективне управління агрономічними ресурсами є ключовим фактором для підвищення врожайності та якості сільськогосподарської продукції. Аналіз взаємозв'язків між хімічними характеристиками ґрунту, кліматичними умовами та потребами різних культур у поживних речовинах дозволяє оптимізувати агротехнічні заходи, забезпечуючи більш раціональне використання добрив та водних ресурсів. У цьому дослідженні представлено графічний аналіз потреб у основних макроелементах (азот, фосфор, калій) для різних сільськогосподарських культур, а також досліджено залежність між температурою, вологістю та рівнем опадів, що створює основу для розробки більш ефективних стратегій вирощування.

На рисунку наведено графічний аналіз потреби різних сільськогосподарських культур у азоті (N). Діаграма, розділена на дві частини, показує культури, що потребують найбільшу (ліворуч) та найменшу (праворуч) кількість азоту. Як видно з рисунка, культура, що має найбільшу потребу в азоті, — це бавовна (117,77 кг/га), за нею слідує кава (101,2 кг/га) та мускусна диня (100,32 кг/га). Висока потреба в азоті цих культур може бути пов'язана з їхніми біологічними потребами у швидкому зростанні та розвитку, що потребує значної кількості цього елемента (Рисунок 1).

З іншого боку, культури, які мають найменшу потребу в азоті, включають кокос (21,98 кг/га), мольний квасоля (21,44 кг/га) та маш (20,99 кг/га). Низькі значення споживання азоту можуть свідчити про їхню здатність ефективно використовувати азот або ж про те, що ці культури ростуть в умовах, де концентрація цього елемента є природно нижчою.

Nitrogen (N)

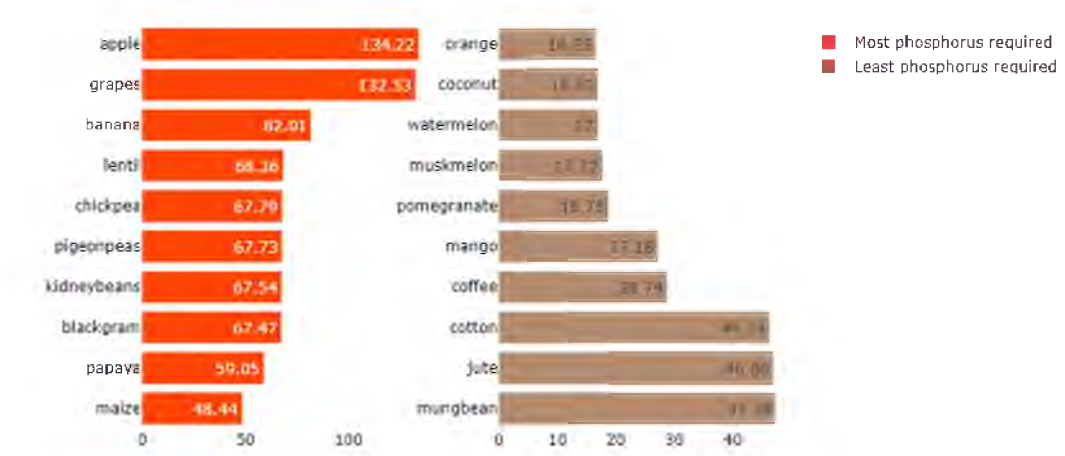


Рисунки 3.1 - Аналіз потреби різних сільськогосподарських культур у азоті (N)

На представленій діаграмі відображено потребу в фосфорі (P) для різних сільськогосподарських культур, які поділені на дві категорії: культури з найвищою та найнижчою потребою у фосфорі. Найбільшу потребу в фосфорі демонструють яблука (134,22 кг/га) та виноград (132,53 кг/га), що вказує на високу залежність цих культур від достатнього забезпечення фосфором для здорового росту та розвитку плодів. Фосфор важливий для процесів передачі енергії та синтезу фосфоліпідів в рослинах, особливо важливих для плодоносних культур (Рисунок 3.2).

З іншого боку, культури з найнижчою потребою в фосфорі, як наприклад апельсин (16,55 кг/га), кокос (16,93 кг/га) та кавун (17 кг/га), можуть бути більш пристосованими до умов з обмеженими ресурсами фосфору або менш залежними від його високих рівнів для продуктивності. Це може свідчити про відмінності в метаболічних шляхах або стратегіях засвоєння поживних речовин серед різних видів культур, що вимагає додаткового дослідження для оптимізації умов вирощування та добрива (Рисунок 3.2).

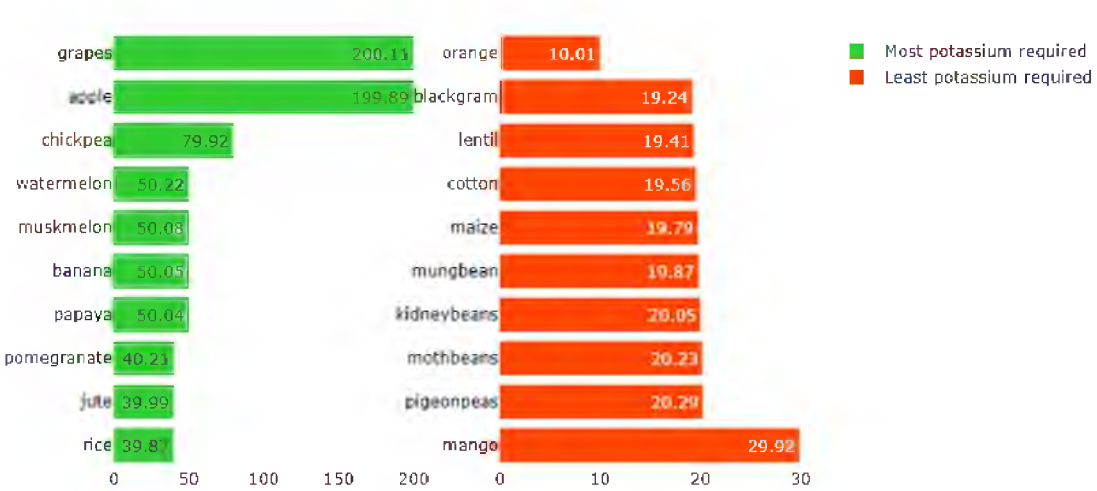
Phosphorus (P)



Рисунки 3.2 - Аналіз потреби різних сільськогосподарських культур у фосфорі (P)

На представленому графіку видно розподіл потреб у калії (K) для різних культур. Виноград (200,11 кг/га) та яблука (199,89 кг/га) є лідерами за потребами в калії, що відображає важливість цього макроелемента для підтримки водного балансу та енергетичного обміну в рослинах, які активно ростуть та плодоносять. На протилежному полюсі, культури як манго (29,92 кг/га) та горох (20,29 кг/га) вимагають значно менше калію, що може бути пов'язано з їхніми більш низькими водними вимогами або меншою інтенсивністю обміну речовин, особливо в фазі активного росту (Рисунок 3.3).

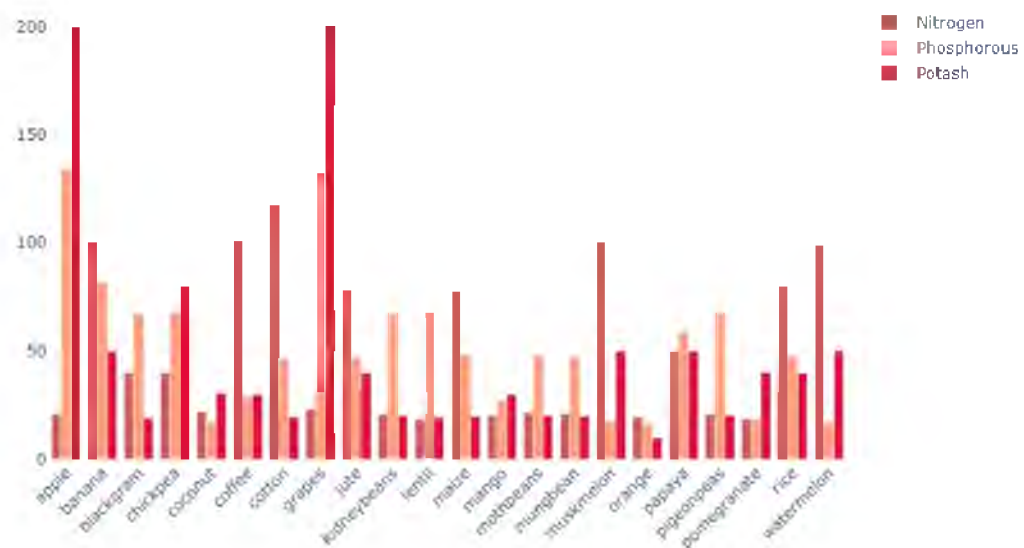
Potassium (K)



Рисунки 3.3 - Аналіз потреби різних сільськогосподарських культур у калії (K)

На діаграмі представлено порівняльний аналіз вмісту азоту (N), фосфору (P) та калію (K) у різних сільськогосподарських культурах. З діаграми видно, що найбільші відмінності в споживанні елементів спостерігаються у культур яблук і винограду, де показники N та K значно вищі порівняно з іншими культурами. Це може свідчити про інтенсивні вимоги цих культур до поживних речовин для підтримки активного росту і розвитку плодів. З іншого боку, культури як манго і горох мають значно нижчі показники по всіх трьох макроелементах, що може вказувати на менші поживні потреби або вищу ефективність використання поживних речовин (Рисунок 3.4).

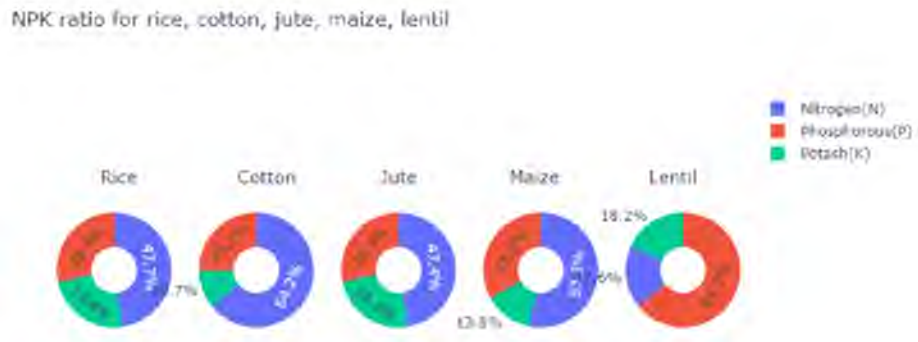
N, P, K values comparision between crops



Рисунки 3.4 - Порівняльний аналіз вмісту азоту (N), фосфору (P) та калію (K)

На діаграмах (Рисунок 3.5) представлено відносні співвідношення азоту (N), фосфору (P) та калію (K) для п'яти вибраних культур: рису, бавовни, джуту, кукурудзи та сочевиці. Видно, що споживання NPK варіюється в залежності від типу культури. Наприклад, для кукурудзи основним компонентом є азот (N), що відіграє ключову роль у підтримці зеленої маси та загального росту. У випадку бавовни, співвідношення фосфору є відносно

вищим, що може свідчити про його роль у розвитку кореневої системи та цвітінні. Сочевиця, з іншого боку, має вищий вміст калію, що необхідно для водного балансу та покращення якості насіння (Рисунок 3.5).



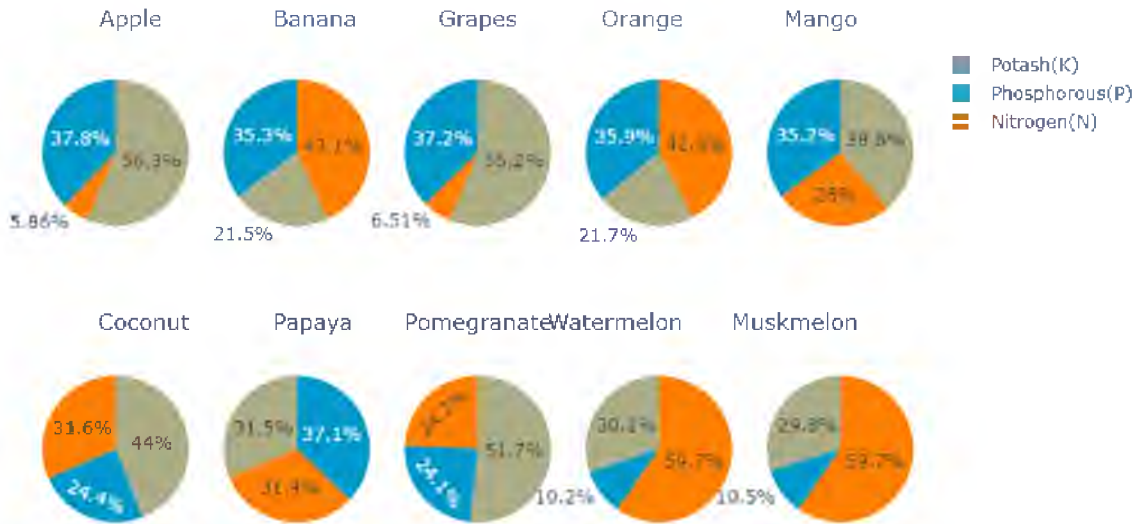
Рисунки 3.5 - Відносне співвідношення азоту (N), фосфору (P) та калію (K) для п'яти вибраних культур

На графіку (Рисунок 3.6) представлено співвідношення азоту (N), фосфору (P) та калію (K) для різних фруктових культур. Важливо відзначити, що для більшості фруктів споживання калію (K) є домінуючим, що підкреслює його ключову роль у регуляції водного балансу та активації ферментів, необхідних для зростання плодів. Яблука та банани показують високу потребу в азоті, що сприяє їх інтенсивному росту листя та розвитку плодів. Наприклад, виноград вимагає значного вмісту калію для підтримки високого якісного врожаю. Водночас, у папайї і кавуна спостерігається високий відсоток калію, що відображає їх великі водні потреби та залежність від калію для підтримки тургорного тиску у великих плодах (Рисунок 3.6).

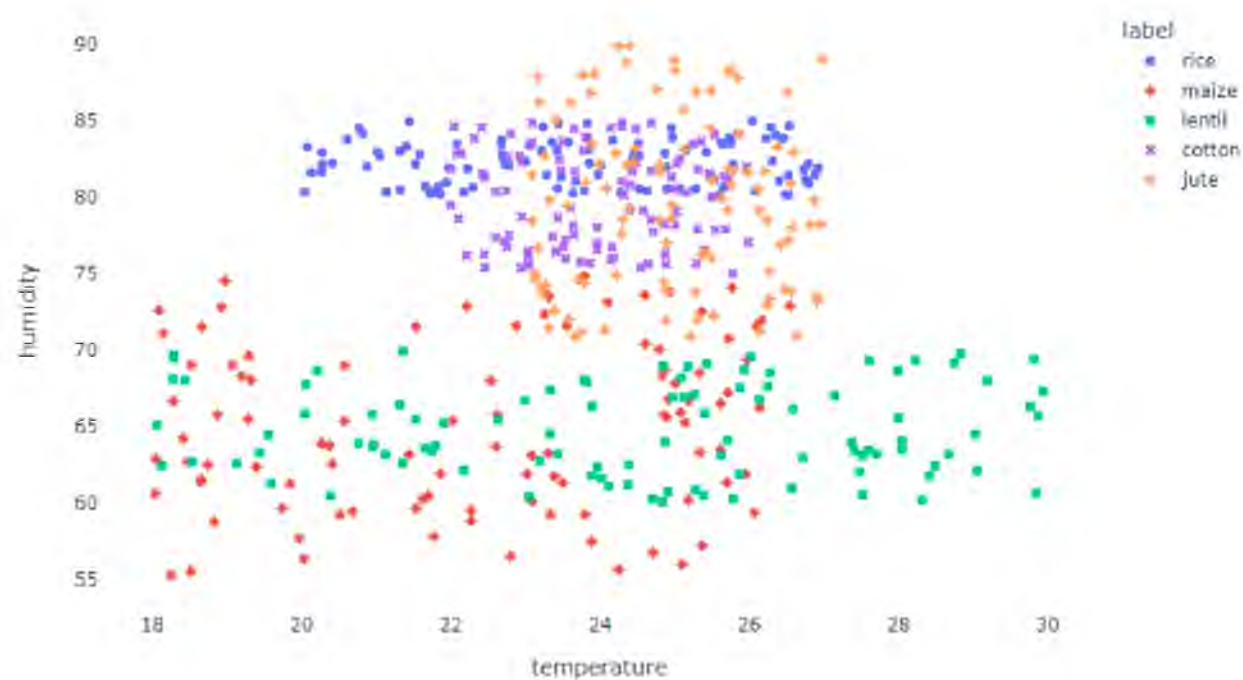
На наведеній діаграмі (Рисунок 3.7) розсіювання представлені зв'язки між температурою та вологістю для п'яти сільськогосподарських культур: рису, кукурудзи, льону, бавовни та сочевиці. Рис виявляється найбільш чутливим до вологісних умов, оскільки займає вищий діапазон вологості (близько 80-90%) при різних температурах, що відображає його потребу в багатих на воду умовах для оптимального росту. В той час як сочевиця займає нижній діапазон вологості (близько 60-70%) при середніх температурах, що вказує на її високу стійкість до сухості. Кукурудза та бавовна показують

помірну залежність від вологості при температурах від 18 до 30 градусів Цельсія, демонструючи їх здатність адаптуватися до широкого спектру кліматичних умов (Рисунок 3.8).

NPK ratio for fruits

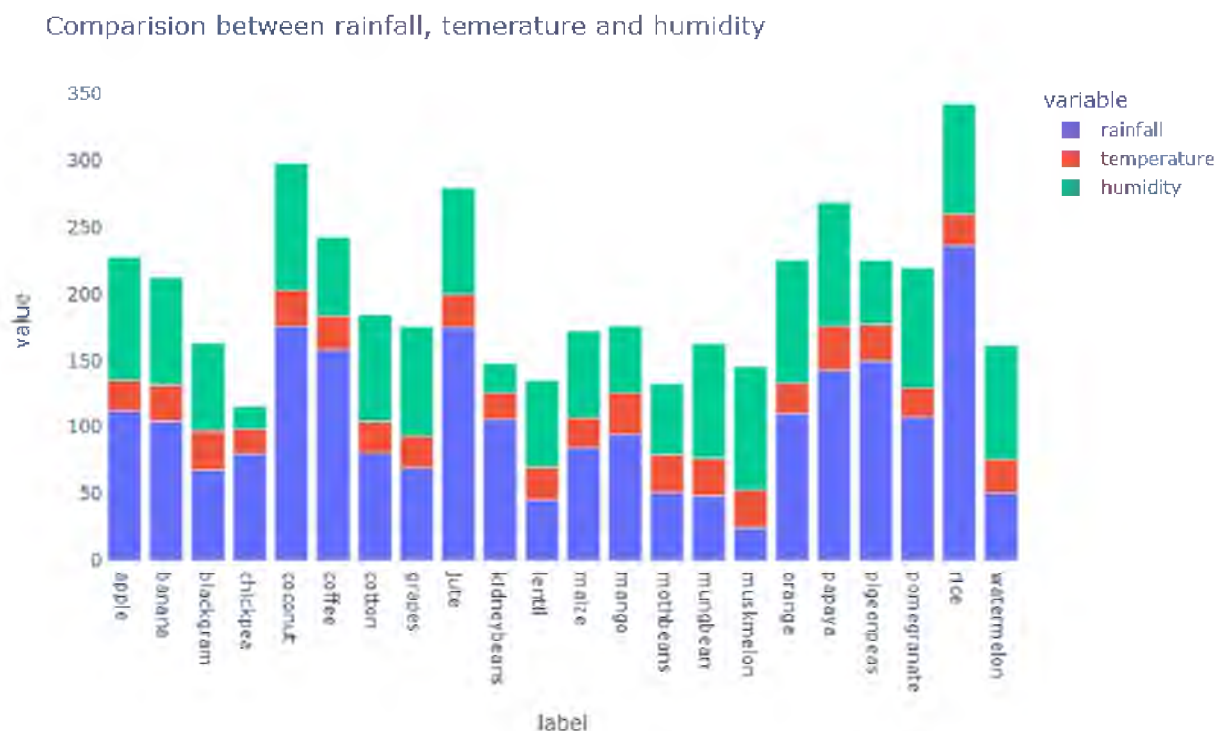


Рисунки 3.6 - Співвідношення азоту (N), фосфору (P) та калію (K) для різних фруктових культур



Рисунки 3.7 - Діаграма розсіювання представлені зв'язки між температурою та вологістю

На діаграмі (Рисунки 3.8) зображено порівняння середніх значень опадів, температури та вологості для різних сільськогосподарських культур. Видно, що культури, такі як кокос та рис, потребують високого рівня опадів, що перевищує 250 мм, поряд з високою вологістю, що є критичним для їхнього успішного росту. Водночас такі культури, як сочевиця та чорна квасоля, мають значно нижчі потреби в опадах, що робить їх більш придатними для вирощування в посушливих регіонах. Температурні вимоги також варіюються, зокрема, банан та манго показують більшу чутливість до вищих температур, тоді як яблука потребують більш помірного клімату.

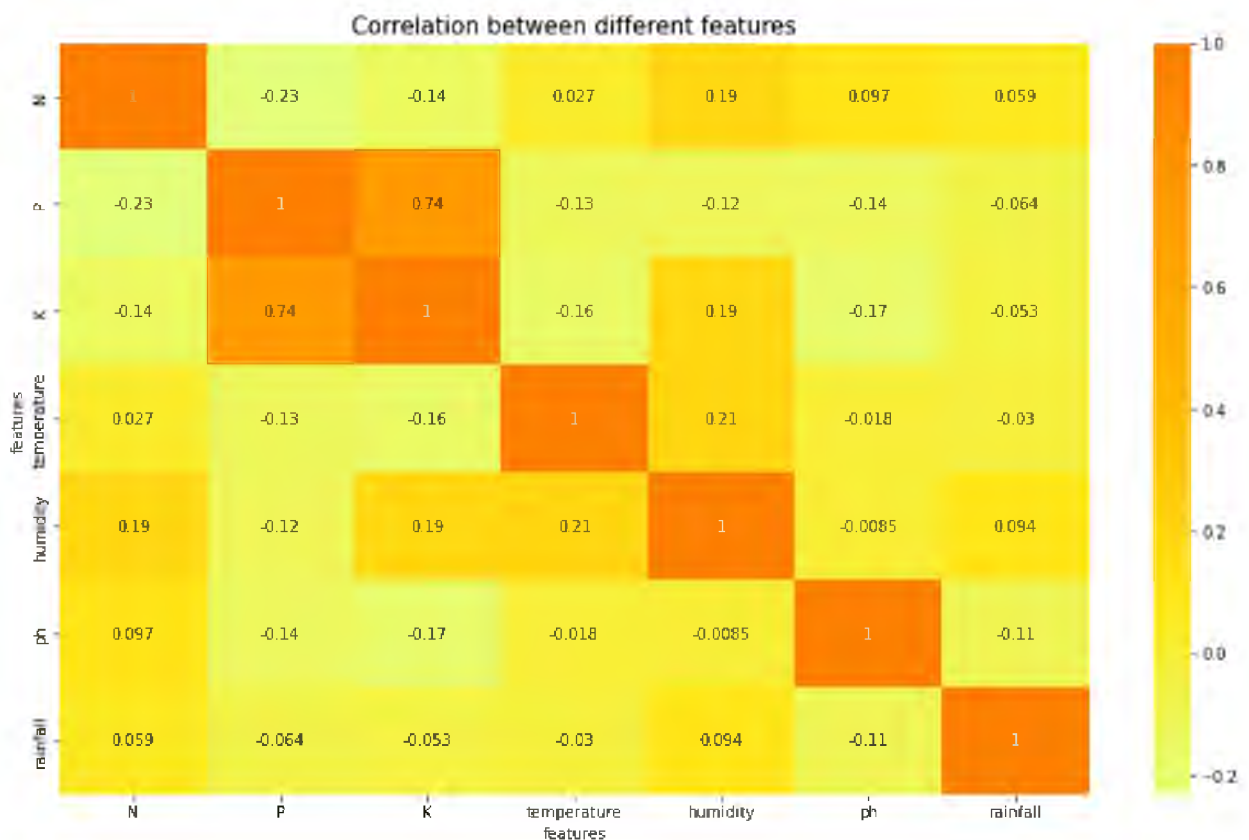


Рисунки 3.8 - Порівняння середніх значень опадів, температури та вологості для різних сільськогосподарських культур

На теплокарті представлені кореляційні зв'язки між різними агрономічними параметрами. Високі позитивні кореляції (наприклад, 0.74 між певними параметрами) ілюструють сильну залежність між ними, що може вказувати на спільні біохімічні шляхи або екологічні чинники, що впливають на ці характеристики. Наприклад, сильна кореляція між хімічними властивостями ґрунту може вказувати на їх спільне значення для здоров'я

рослин і урожайності. Це демонструє, як взаємозв'язки між фізичними та хімічними параметрами ґрунту можуть впливати на ріст та розвиток рослин (Рисунок 3.9).

З іншого боку, низькі та негативні кореляції (наприклад, -0.23) можуть вказувати на те, що певні параметри взаємодіють протилежно або не залежать один від одного. Негативна кореляція може вказувати на компенсаційні механізми у рослин, де високі рівні одного параметра можуть балансуватися низькими рівнями іншого, щоб підтримувати гомеостаз. Розуміння цих взаємозв'язків допомагає в розробці більш ефективних стратегій управління агросистемами, наприклад, у правильному застосуванні добрив, щоб оптимізувати співвідношення поживних речовин в ґрунті для підвищення урожайності.



Рисунки 3.9 - Теплокарта кореляційних зв'язків між різними агрономічними параметрами

Аналіз графічних даних та кореляційних зв'язків між агрономічними параметрами вказує на суттєві відмінності в потребах різних культур до

поживних речовин та кліматичних умов. Висока потреба у фосфорі для плодоносних культур, таких як яблука та виноград, підкреслює важливість цього елемента для їхнього росту та розвитку. Водночас, культури з низькими потребами в азоті, такі як кокос та маш, демонструють здатність ефективно використовувати цей елемент або рости в умовах з низьким його вмістом. Ці результати сприяють кращому розумінню оптимальних умов для вирощування різних культур та можуть бути використані для розробки цілеспрямованих рекомендацій з управління агросистемами, що підвищує ефективність сільського господарства та стійкість до змін клімату.

3.3 Аналіз ефективності моделі LightGBM у класифікації сільськогосподарських культур

Для створення моделі машинного навчання було обрано LightGBM, яка відзначається високою швидкістю та ефективністю в обробці великих обсягів даних. Незалежні змінні (X) включають агрономічні параметри, такі як вміст азоту (N), фосфору (P), калію (K), температура, вологість, pH ґрунту та кількість опадів. Цільова змінна (y) — це мітка культури. Використання цих параметрів дозволяє моделі прогнозувати найкращу культуру для вирощування в даних умовах. Розподіл даних між навчальною та тестовою вибірками забезпечив надійну оцінку моделі, з 70% даних, відведених на навчання, та 30% на тестування.

Після підготовки даних модель LightGBM була побудована та навчена на навчальній вибірці. Завдяки своїй здатності обробляти нерівномірно розподілені дані та враховувати взаємозв'язки між параметрами, модель LightGBM швидко зійшлася до високої точності. Після навчання моделі було проведено прогнозування на тестовій вибірці, де модель показала високу точність 98.94% (Таблиця 3.10).

```
# view accuracy
from sklearn.metrics import accuracy_score

accuracy=accuracy_score(y_pred, y_test)
print('LightGBM Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test, y_pred)))
```

LightGBM Model accuracy score: 0.9894

Рисунки 3.11 – Оцінка точності моделі LightGBM

Розрахунок точності моделі на навчальній вибірці показав значення 100%, що свідчить про повну відповідність моделі даним, на яких вона навчалася. Це може вказувати на можливу перенавченість моделі, коли модель занадто точно відображає тренувальні дані, що може знижувати її узагальнюючу здатність на нових даних. Висока точність на тестовій вибірці підтверджує здатність моделі до ефективного прогнозування, проте невеликий розрив між точністю на тренувальних та тестових даних вказує на потенційний ризик перенавчання (Рисунок 3.12).

```
y_pred_train = model.predict(X_train)
print('Training-set accuracy score: {0:0.4f}'.format(accuracy_score(y_train, y_pred_train)))
```

Training-set accuracy score: 1.0000

Рисунки 3.12 – Оцінка точності навчального набору

Для перевірки перенавченості було додатково оцінено показники моделі на обох вибірках. Значення точності на тренувальній вибірці становило 100%, а на тестовій — 98.94%. Такі результати свідчать про відмінну продуктивність моделі, проте для уникнення ризику перенавчання можливо знадобиться подальше налаштування параметрів моделі або застосування додаткових методів регуляризації, таких як крос-валідація або зменшення складності моделі (Таблиця 3.13).

```
# print the scores on training and test set

print('Training set score: {:.4f}'.format(model.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(model.score(X_test, y_test)))
```

```
Training set score: 1.0000
Test set score: 0.9894
```

Рисунки 3.13 – Оцінка тренувального та тестового набору

Загалом, модель LightGBM продемонструвала високу ефективність у задачі прогнозування сільськогосподарських культур на основі агрономічних параметрів. Використання цієї моделі дозволяє створити надійну систему рекомендацій для фермерів, що може значно підвищити ефективність сільськогосподарського виробництва. Надалі, для покращення моделі, можна розглянути застосування додаткових параметрів або алгоритмів ансамблювання для подальшого підвищення точності та узагальнюючої здатності моделі.

Матриця помилок, яка представлена на Рисунку 3.15, є критичним інструментом у вимірюванні ефективності моделі класифікації. Вона надає візуальне і кількісне представлення про те, як добре модель здійснює прогнози по відношенню до фактичних класів. У цій матриці кожен рядок відповідає фактичним класам, тоді як кожен стовпець відповідає прогнозованим класам. Числа на головній діагоналі показують кількість правильних прогнозів для кожного класу, тоді як числа поза головною діагоналлю відображають кількість помилок.

На Рисунку 3.15 спостерігається висока концентрація значень на діагоналі, що свідчить про високу точність моделі. Проте, незначні числа в недіагональних комірках вказують на деякі помилки класифікації, які можуть бути викликані шумом у даних або недостатньою репрезентативністю деяких класів в навчальній вибірці. Аналіз цих помилок може допомогти в

ідентифікації можливих векторів для покращення моделі, наприклад, через перебалансування набору даних або зміну параметрів моделювання.

Високий рівень точності, який показує модель на тестових даних (98,94%), підкріплюється зображенням матриці помилок. Це підкреслює здатність LightGBM ефективно розрізняти між різними класами, незважаючи на високу складність і різноманітність даних. Надійність моделі в реальних умовах може бути додатково перевірена за допомогою крос-валідації, що забезпечить більш глибоке розуміння її стійкості до нових даних.

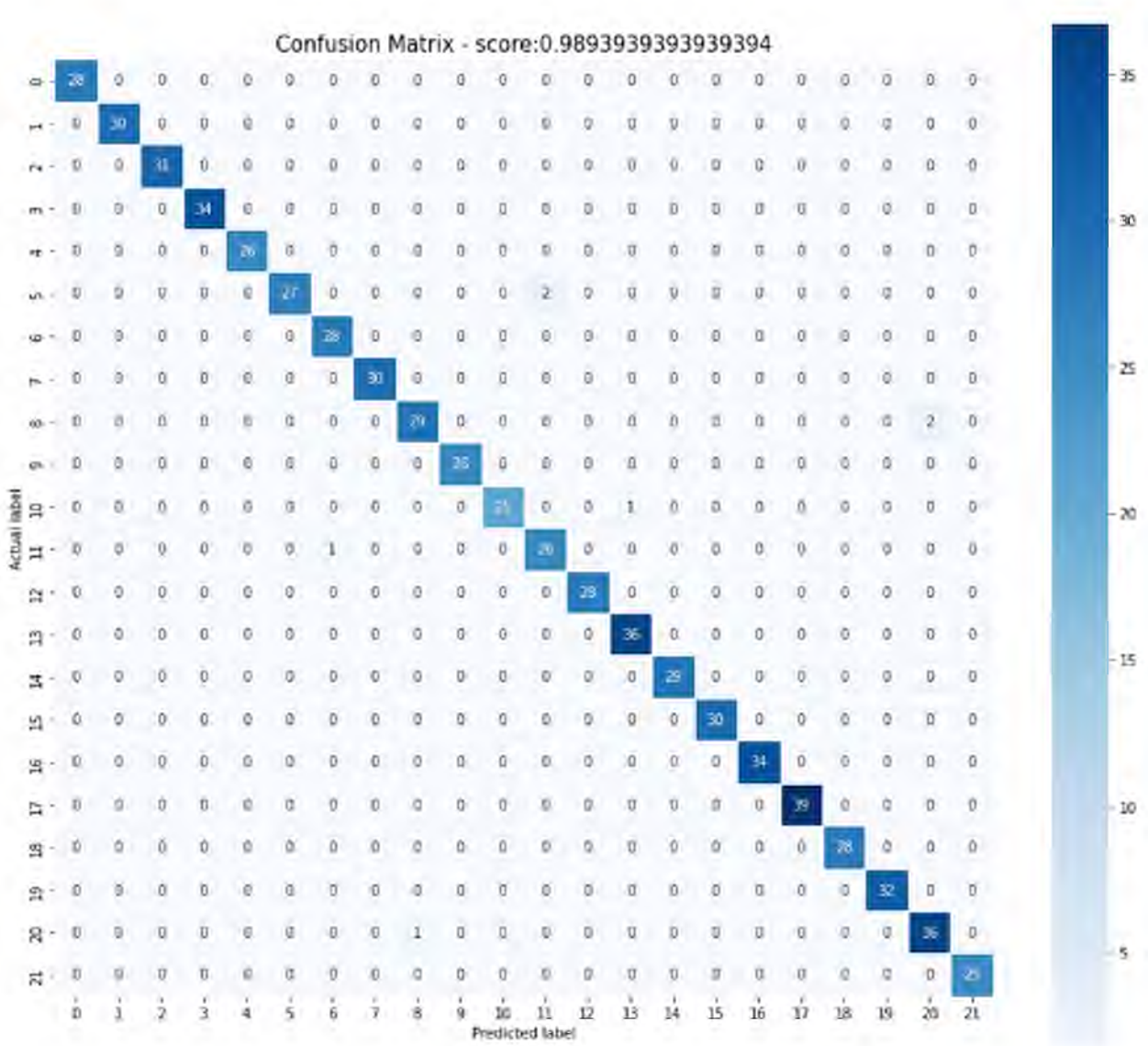


Рисунок 3.14 – Confusion-matrix

Загалом, ретельний аналіз матриці помилок і відповідних метрик точності дозволяє не тільки оцінити поточну ефективність моделі, але й вказує на шляхи її покращення. Це включає в себе налаштування параметрів, використання додаткових методів для обробки даних та збільшення об'єму та якості навчальних даних, щоб зміцнити здатність моделі до узагальнення на нових даних. Такий підхід сприятиме створенню більш надійних і точних систем прогнозування для сільського господарства.

Деталізація метрик класифікації на основі звіту `classification_report` відображає високу точність моделі LightGBM у різних категоріях сільськогосподарських культур. Звіт включає метрики точності (`precision`), повноти (`recall`) та F1-балу для кожної культури, що дозволяє глибоко оцінити ефективність моделі в умовах мультикласової класифікації. Відзначається, що для більшості культур, таких як яблуко, банан, чорна квасоля, та інші, модель показала ідеальні результати із точністю та повнотою, що дорівнюють 1.00, свідчаючи про відмінну здатність моделі правильно ідентифікувати ці культури (Рисунок 3.15).

Проте, деякі культури, такі як кава і рис, мають трохи нижчі показники F1-балу (0.96 для кави і 0.96 для рису), що може свідчити про певні труднощі моделі в розпізнаванні цих культур з максимальною точністю через можливу схожість агрономічних параметрів з іншими культурами або недостатньо представленими даними в навчальній вибірці. Це підкреслює потребу в подальшій оптимізації моделі або збалансуванні даних для цих культур.

Крім того, аналіз метрик макро та взважених середніх значень показує загальну ефективність моделі на високому рівні з показниками, що наближаються до 0.99. Це свідчить про високу універсальність та надійність моделі у прогнозуванні правильних культур, що робить її цінним інструментом для застосування в сучасних агротехнологіях. Використання цієї моделі може значно підвищити продуктивність та ефективність аграрного

виробництва, дозволяючи фермерам оптимізувати використання ресурсів та збільшити урожайність.

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	28
banana	1.00	1.00	1.00	30
blackgram	1.00	1.00	1.00	31
chickpea	1.00	1.00	1.00	34
coconut	1.00	1.00	1.00	26
coffee	1.00	0.93	0.96	29
cotton	0.97	1.00	0.98	28
grapes	1.00	1.00	1.00	30
jute	0.97	0.94	0.95	31
kidneybeans	1.00	1.00	1.00	26
lentil	1.00	0.95	0.98	22
maize	0.93	0.96	0.95	27
mango	1.00	1.00	1.00	28
mothbeans	0.97	1.00	0.99	36
mungbean	1.00	1.00	1.00	29
muskmelon	1.00	1.00	1.00	30
orange	1.00	1.00	1.00	34
papaya	1.00	1.00	1.00	39
pigeonpeas	1.00	1.00	1.00	28
pomegranate	1.00	1.00	1.00	32
rice	0.95	0.97	0.96	37
watermelon	1.00	1.00	1.00	25
accuracy			0.99	660
macro avg	0.99	0.99	0.99	660
weighted avg	0.99	0.99	0.99	660

Рисунки 3.15 – Класифікаційний звіт

Наостанок, проведення прогнозування на нових даних з довільно вибраними параметрами для культури рису демонструє практичну застосовність моделі (Рисунок 3.16). Подання вхідних даних до моделі та отримання точного прогнозу підтверджує її здатність ефективно використовуватися в реальних умовах, де важливим є швидке та точне рішення про вибір культур на основі доступних агрономічних даних.

```
newdata=model.predict([[90, 42, 43, 20.879744, 75, 5.5,220]])
newdata
array(['rice'], dtype=object)
```

Рисунок 3.16 - Прогнозування з довільно вибраними параметрами для культури рису

Загалом, модель LightGBM продемонструвала високу ефективність у задачі прогнозування сільськогосподарських культур на основі агрономічних параметрів. Використання цієї моделі дозволяє створити надійну систему рекомендацій для фермерів, що може значно підвищити ефективність сільськогосподарського виробництва. Надалі, для покращення моделі, можна розглянути застосування додаткових параметрів або алгоритмів ансамблювання для подальшого підвищення точності та узагальнюючої здатності моделі.

ВИСНОВКИ

У підсумку, детальний аналіз агрономічних параметрів та їх впливу на врожайність підкреслює критичну роль точного управління ресурсами в аграрному секторі. Вміння адаптувати сільськогосподарські технології до змінних агрокліматичних умов, застосування передових алгоритмів машинного навчання, як LightGBM, в системах рекомендацій дозволяє оптимізувати вирощування культур, підвищувати їхню врожайність та ефективність. Розуміння кожного аспекту агрономічних параметрів, від поживних речовин у ґрунті до впливу кліматичних факторів, є ключовим для забезпечення стійкості та адаптивності агроєкосистем у довгостроковій перспективі. Таке глибоке інтегроване знання сприяє не тільки досягненню оптимальних врожаїв, але й вирішенню глобальних викликів у сфері продовольства та стійкого розвитку.

Система рекомендацій на основі машинного навчання для агрокультур, використовуючи дані про ґрунт і клімат, являє собою перспективний напрямок у сучасному агрономічному виробництві. Завдяки точному аналізу агрономічних даних і впровадженню алгоритмів машинного навчання, таких як LightGBM, можна не тільки значно підвищити урожайність, але й оптимізувати використання ресурсів та зменшити екологічний вплив агропромисловості. Це дозволяє аграріям адаптуватися до змінних умов з максимальною ефективністю, що є ключовим для забезпечення продовольчої безпеки в умовах глобальних змін.

Архітектура модуля рекомендацій, що включає аналітичні підрозділи та інтеграційні інтерфейси, сприяє гнучкості та масштабованості агрономічних досліджень. Застосування передових технологій дозволяє не тільки вирішувати актуальні завдання, але й антиципувати майбутні виклики, що ставить аграрну науку на передові позиції у вирішенні глобальних проблем сталого розвитку. В результаті, сільське господарство перетворюється з

традиційної галузі в сферу високотехнологічних інновацій, що забезпечує продуктивність і адаптивність до змінюваних умов довкілля.

Завершальний аналіз програмно-технологічного забезпечення, спрямованого на оптимізацію вирощування агрокультур за допомогою алгоритмів машинного навчання, демонструє значний потенціал у використанні комплексних даних про ґрунт та кліматичні умови. Впровадження таких технологій дозволяє не тільки підвищити точність агрономічних прогнозів, але й забезпечити більш ефективне використання ресурсів, зниження витрат на добрива та зрошення, а також збільшення урожайності за рахунок адаптації культур до специфічних умов вирощування.

Значний внесок у цю систему вносить використання алгоритму LightGBM, який ефективно аналізує великі обсяги даних, забезпечуючи високу швидкість обробки та точність прогнозування. Результати, отримані завдяки цьому алгоритму, підтверджують його придатність для задач класифікації агрокультур на основі їхніх агрономічних вимог, що робить технологію не тільки науково обґрунтованою, але й вкрай корисною для практичного застосування у сфері сільськогосподарського виробництва.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mukherjee, S., Dholaria, S., Ratn, S., & Gudibandi, S. (2022). Predictive and Preventive methods in Agriculture using Machine Learning. Отримано з ResearchGate: https://www.researchgate.net/profile/Subbhashit-Mukherjee/publication/360513688_Predictive_and_Preventive_methods_in_Agriculture_using_Machine_Learning/links/627b4444973bbb29cc7444e2/Predictive-and-Preventive-methods-in-Agriculture-using-Machine-Learning.pdf
2. Faizanuddin, M., Fathima, A., & Islam, M. U. (2023). Revolutionizing Indian Agriculture through Machine Learning-Based Crop Recommendation System. Отримано з ResearchGate: https://www.researchgate.net/profile/Mohd-Faizaanuddin/publication/378124693_Revolutionizing_Indian_Agriculture_through_Machine_Learning-Based_Crop_Recommendation_system/links/65c7a5c179007454976fa4ae/Revolutionizing-Indian-Agriculture-through-Machine-Learning-Based-Crop-Recommendation-system.pdf
3. Dutta, M., Gupta, D., Juneja, S., Shah, A., & Shaikh, A. (2023). Boosting of Fruit Choices Using Machine Learning-Based Pomological Recommendation System. Отримано з Springer: <https://link.springer.com/article/10.1007/s42452-023-05462-0>
4. Yan, J., Xu, Y., Cheng, Q., Jiang, S., Wang, Q., Xiao, Y., & Ma, C. (2021). LightGBM: Accelerated Genomically Designed Crop Breeding through Ensemble Learning. Отримано з Springer: <https://link.springer.com/article/10.1186/s13059-021-02492-y>
5. Mao, X., Ren, N., Dai, P., Jin, J., Wang, B., & Kang, R. (2024). A Variable Weight Combination Prediction Model for Climate in a Greenhouse Based on BiGRU-Attention and LightGBM. Отримано з ScienceDirect: <https://www.sciencedirect.com/science/article/pii/S0168169924002096>

6. Crop Recommendation Dataset. *Kaggle: Your Machine Learning and Data Science Community*. URL: <https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset> (date of access: 13.05.2024).
7. Kulkarni, M. S., Nerurkar, M. T., & Kulkarni, P. (2022). Implementation of Artificial Intelligence for Agriculture. Отримано з ResearchGate: https://www.researchgate.net/profile/Pappu-Sindhuja/publication/375090926_19ImpactofWebTechnologyMarketingonConsumerDurableItems/links/6540dc8eff8d8f507cdc15e1/19ImpactofWebTechnologyMarketingonConsumerDurableItems.pdf#page=168
8. Shams, M. Y., Gamel, S. A., & Talaat, F. M. (2024). Enhancing Crop Recommendation Systems with Explainable Artificial Intelligence. Отримано з Springer: <https://link.springer.com/article/10.1007/s00521-023-09391-2>
9. Hasan, M., Marjan, M. A., Uddin, M. P., & Afjal, M. I. (2023). Ensemble Machine Learning-Based Recommendation System for Effective Prediction of Suitable Agricultural Crop Cultivation. Отримано з Frontiers in Plant Science: <https://www.frontiersin.org/articles/10.3389/fpls.2023.1234555/full>
10. Kumari, H., & Sambhav, S. (2023). A Crop Recommendation System for Better Sustainability Using Machine Learning Techniques. Отримано з IET Digital Library: <https://digital-library.theiet.org/content/conferences/10.1049/icp.2023.1512>
11. Halbe, S., Lad, D., & Panchal, T. (2023). Krishi Mitra-Intelligent Crop And Fertilizer Recommender. Отримано з IEEE Xplore: <https://ieeexplore.ieee.org/abstract/document/10454948/>
12. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

13. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

ДОДАТОК А

ПСЕВДОКОД АЛГОРИТМУ

ПОЧАТОК

Імпорт необхідних бібліотек (pandas, numpy, matplotlib, seaborn, plotly)

Функція завантаження _даних():

Завантажити набір даних за допомогою pandas.read_csv()

Вивести перші кілька рядків даних для перевірки

Функція перевірка _даних():

Якщо виявлені пропущені значення у наборі даних:

Обробити пропущені значення або видалити рядки/стовпці

Функція візуалізація _даних():

Використати matplotlib і seaborn для створення графіків

Використати plotly для інтерактивних візуалізацій

Відобразити розподіл нітрогену, фосфору, калію за культурами

Функція обробка _даних():

Розділити дані на незалежні змінні (X) і залежні змінні (y)

Розділити набір даних на тренувальні і тестові дані

Функція моделювання():

Ініціалізувати модель LightGBM

Навчити модель на тренувальних даних

Функція прогнозування():

Прогнозування на тестових даних

Оцінити точність моделі (метрики класифікації, матриця помилок)

Функція відображення _результатів():

Вивести результати класифікації

Показати додаткові візуалізації для аналізу результатів

Функція аналіз _результатів():

Порівняти важливість ознак

Перевірити модель на перенавчання

КІНЕЦЬ

ДОДАТОК Б

КОД ДЛЯ РЕАЛІЗАЦІЇ

```

import pandas as pd
import numpy as np
import random

import matplotlib.pyplot as plt
import seaborn as sns

import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots

colorarr = ['#0592D0', '#Cd7f32', '#E97451', '#Bdb76b', '#954535', '#C2b280', '#808000', '#C2b280', '#E4d008',
'#9acd32', '#Eedc82', '#E4d96f',
'#32cd32', '#39ff14', '#00ff7f', '#008080', '#36454f', '#F88379', '#Ff4500', '#Ffb347', '#A94064', '#E75480',
'#Ffb6c1', '#E5e4e2',
'#Fafoe6', '#8c92ac', '#Dbd7d2', '#A7a6ba', '#B38b6d']

# %%[markdown]
# <h1 style='color:blue'>Import dataset</h1>

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:03.981458Z","iopub.execute_input":"2021-10-03T08:53:03.981697Z","iopub.status.idle":"2021-10-03T08:53:04.024307Z","shell.execute_reply.started":"2021-10-03T08:53:03.981668Z","shell.execute_reply":"2021-10-03T08:53:04.023373Z"}}
cropdf = pd.read_csv("../input/crop-recommendation-dataset/Crop_recommendation.csv")
cropdf.head()

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.025577Z","iopub.execute_input":"2021-10-03T08:53:04.025806Z","iopub.status.idle":"2021-10-03T08:53:04.032867Z","shell.execute_reply.started":"2021-10-03T08:53:04.025779Z","shell.execute_reply":"2021-10-03T08:53:04.031744Z"}}
cropdf.shape

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.034758Z","iopub.execute_input":"2021-10-03T08:53:04.035534Z","iopub.status.idle":"2021-10-03T08:53:04.049166Z","shell.execute_reply.started":"2021-10-03T08:53:04.035484Z","shell.execute_reply":"2021-10-03T08:53:04.04837Z"}}
cropdf.columns

# %%[markdown]
# <h1 style='color:blue'>Check Null and NA values</h1>

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.050901Z","iopub.execute_input":"2021-10-03T08:53:04.05118Z","iopub.status.idle":"2021-10-03T08:53:04.06878Z","shell.execute_reply.started":"2021-10-03T08:53:04.05113Z","shell.execute_reply":"2021-10-03T08:53:04.067767Z"}}
cropdf.isnull().any()

# %%[markdown]
# <h1 style='color:blue'>List of various crops</h1>

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.070471Z","iopub.execute_input":"2021-10-03T08:53:04.070854Z","iopub.status.idle":"2021-10-03T08:53:04.086021Z","shell.execute_reply.started":"2021-10-03T08:53:04.070822Z","shell.execute_reply":"2021-10-03T08:53:04.08533Z"}}
print("Number of various crops: ", len(cropdf['label'].unique()))
print("List of crops: ", cropdf['label'].unique())

```

```

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.08685Z","iopub.execute_input":"2021-10-03T08:53:04.087066Z","iopub.status.idle":"2021-10-03T08:53:04.107811Z","shell.execute_reply.started":"2021-10-03T08:53:04.087041Z","shell.execute_reply":"2021-10-03T08:53:04.106811Z"}}
cropdf['label'].value_counts()

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.109064Z","iopub.execute_input":"2021-10-03T08:53:04.109445Z","iopub.status.idle":"2021-10-03T08:53:04.150743Z","shell.execute_reply.started":"2021-10-03T08:53:04.109413Z","shell.execute_reply":"2021-10-03T08:53:04.149786Z"}}
crop_summary = pd.pivot_table(cropdf,index=['label'],aggfunc='mean')
crop_summary.head()

# %%[markdown]
# <h1 style='color:blue'>Data Visualization and analysis</h1>

# %%[markdown]
# <h3 style='color:red'>Nitrogen Analysis</h3>

# %%[code] {"_kg_hide-input":false,"execution":{"iopub.status.busy":"2021-10-03T08:53:04.151893Z","iopub.execute_input":"2021-10-03T08:53:04.1525Z","iopub.status.idle":"2021-10-03T08:53:04.452706Z","shell.execute_reply.started":"2021-10-03T08:53:04.152463Z","shell.execute_reply":"2021-10-03T08:53:04.451879Z"}}
crop_summary_N = crop_summary.sort_values(by='N', ascending=False)

fig = make_subplots(rows=1, cols=2)

top = {
    'y': crop_summary_N['N'][0:10].sort_values().index,
    'x': crop_summary_N['N'][0:10].sort_values()
}

last = {
    'y': crop_summary_N['N'][-10:].index,
    'x': crop_summary_N['N'][-10:]
}

fig.add_trace(
    go.Bar(top,
        name="Most nitrogen required",
        marker_color=random.choice(colorarr),
        orientation='h',
        text=top['x']),

    row=1, col=1
)

fig.add_trace(
    go.Bar(last,
        name="Least nitrogen required",
        marker_color=random.choice(colorarr),
        orientation='h',
        text=last['x']),

    row=1, col=2
)

fig.update_traces(texttemplate='%{text}!', textposition='inside')
fig.update_layout(title_text="Nitrogen (N)",
    plot_bgcolor='white',
    font_size=12,
    font_color='black',

```

```

        height=500)

fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()

# %%[markdown]
#

# %%[markdown]
# <h3 style='color:red'>Phosphorus Analysis</h3>

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.455578Z","iopub.execute_input":"2021-10-03T08:53:04.455828Z","iopub.status.idle":"2021-10-03T08:53:04.510058Z","shell.execute_reply.started":"2021-10-03T08:53:04.45582Z","shell.execute_reply":"2021-10-03T08:53:04.509131Z"}}
crop_summary_P = crop_summary.sort_values(by='P', ascending=False)

fig = make_subplots(rows=1, cols=2)

top = {
    'y': crop_summary_P['P'][0:10].sort_values().index,
    'x': crop_summary_P['P'][0:10].sort_values()
}

last = {
    'y': crop_summary_P['P'][-10:].index,
    'x': crop_summary_P['P'][-10:]
}

fig.add_trace(
    go.Bar(top,
        name="Most phosphorus required",
        marker_color=random.choice(colorarr),
        orientation='h',
        text=top['x']),

    row=1, col=1
)

fig.add_trace(
    go.Bar(last,
        name="Least phosphorus required",
        marker_color=random.choice(colorarr),
        orientation='h',
        text=last['x']),

    row=1, col=2
)

fig.update_traces(texttemplate='%{text}!', textposition='inside')
fig.update_layout(title_text="Phosphorus (P)",
    plot_bgcolor='white',
    font_size=12,
    font_color='black',
    height=500)

fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()

```

```

# %% [markdown]
# <h3 style='color:Red'>Potassium analysis</h3>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.511338Z","iopub.execute_input":"2021-10-03T08:53:04.511553Z","iopub.status.idle":"2021-10-03T08:53:04.562721Z","shell.execute_reply.started":"2021-10-03T08:53:04.511528Z","shell.execute_reply":"2021-10-03T08:53:04.561881Z"}}
crop_summary_K = crop_summary.sort_values(by='K', ascending=False)

fig = make_subplots(rows=1, cols=2)

top = {
    'y': crop_summary_K['K'][0:10].sort_values().index,
    'x': crop_summary_K['K'][0:10].sort_values()
}

last = {
    'y': crop_summary_K['K'][-10:].index,
    'x': crop_summary_K['K'][-10:]
}

fig.add_trace(
    go.Bar(top,
        name="Most potassium required",
        marker_color=random.choice(colorarr),
        orientation='h',
        text=top['x']),

    row=1, col=1
)

fig.add_trace(
    go.Bar(last,
        name="Least potassium required",
        marker_color=random.choice(colorarr),
        orientation='h',
        text=last['x']),

    row=1, col=2
)

fig.update_traces(texttemplate='%{text}!', textposition='inside')
fig.update_layout(title_text="Potassium (K)",
    plot_bgcolor='white',
    font_size=12,
    font_color='black',
    height=500)

fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()

# %% [markdown]
# <h3 style='color:Red'>N, P, K values comparision between crops</h3>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.564056Z","iopub.execute_input":"2021-10-03T08:53:04.564387Z","iopub.status.idle":"2021-10-

```

```

03T08:53:04.586937Z","shell.execute_reply.started":"2021-10-
03T08:53:04.564346Z","shell.execute_reply":"2021-10-03T08:53:04.585873Z"}}
fig = go.Figure()
fig.add_trace(go.Bar(
    x=crop_summary.index,
    y=crop_summary['N'],
    name='Nitrogen',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=crop_summary.index,
    y=crop_summary['P'],
    name='Phosphorous',
    marker_color='lightsalmon'
))
fig.add_trace(go.Bar(
    x=crop_summary.index,
    y=crop_summary['K'],
    name='Potash',
    marker_color='crimson'
))

fig.update_layout(title="N, P, K values comparision between crops",
    plot_bgcolor='white',
    barmode='group',
    xaxis_tickangle=-45)

fig.show()

# %% [markdown]
# <h3 style='color:Red'>NPK ratio for rice, cotton, jute, maize, lentil </h3>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.588021Z","iopub.execute_input":"2021-10-
03T08:53:04.588259Z","iopub.status.idle":"2021-10-03T08:53:04.685014Z","shell.execute_reply.started":"2021-
10-03T08:53:04.588232Z","shell.execute_reply":"2021-10-03T08:53:04.68421Z"}}
labels = ['Nitrogen(N)', 'Phosphorous(P)', 'Potash(K)']
fig = make_subplots(rows=1, cols=5, specs=[[{'type':'domain'}, {'type':'domain'},
    {'type':'domain'}, {'type':'domain'},
    {'type':'domain'}]])

rice_npk = crop_summary[crop_summary.index=='rice']
values = [rice_npk['N'][0], rice_npk['P'][0], rice_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values, name="Rice"), 1, 1)

cotton_npk = crop_summary[crop_summary.index=='cotton']
values = [cotton_npk['N'][0], cotton_npk['P'][0], cotton_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values, name="Cotton"), 1, 2)

jute_npk = crop_summary[crop_summary.index=='jute']
values = [jute_npk['N'][0], jute_npk['P'][0], jute_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values, name="Jute"), 1, 3)

maize_npk = crop_summary[crop_summary.index=='maize']
values = [maize_npk['N'][0], maize_npk['P'][0], maize_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values, name="Maize"), 1, 4)

lentil_npk = crop_summary[crop_summary.index=='lentil']
values = [lentil_npk['N'][0], lentil_npk['P'][0], lentil_npk['K'][0]]

```

```

fig.add_trace(go.Pie(labels=labels, values=values,name="Lentil"),1, 5)

fig.update_traces(hole=.4, hoverinfo="label+percent+name")
fig.update_layout(
    title_text="NPK ratio for rice, cotton, jute, maize, lentil",
    annotations=[dict(text='Rice',x=0.06,y=0.8, font_size=15, showarrow=False),
                  dict(text='Cotton',x=0.26,y=0.8, font_size=15, showarrow=False),
                  dict(text='Jute',x=0.50,y=0.8, font_size=15, showarrow=False),
                  dict(text='Maize',x=0.74,y=0.8, font_size=15, showarrow=False),
                  dict(text='Lentil',x=0.94,y=0.8, font_size=15, showarrow=False)])
fig.show()

# %% [markdown]
# <h3 style='color:Red'>NPK ratio for fruits</h3>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.686507Z","iopub.execute_input":"2021-10-03T08:53:04.686958Z","iopub.status.idle":"2021-10-03T08:53:04.771624Z","shell.execute_reply.started":"2021-10-03T08:53:04.686917Z","shell.execute_reply":"2021-10-03T08:53:04.770682Z"}}
labels = ['Nitrogen(N)', 'Phosphorous(P)', 'Potash(K)']
specs = [[{'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}], [
    {'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}]]
fig = make_subplots(rows=2, cols=5, specs=specs)
cafe_colors = ['rgb(255, 128, 0)', 'rgb(0, 153, 204)', 'rgb(173, 173, 133)']

apple_npk = crop_summary[crop_summary.index=='apple']
values = [apple_npk['N'][0], apple_npk['P'][0], apple_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Apple", marker_colors=cafe_colors),1, 1)

banana_npk = crop_summary[crop_summary.index=='banana']
values = [banana_npk['N'][0], banana_npk['P'][0], banana_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Banana", marker_colors=cafe_colors),1, 2)

grapes_npk = crop_summary[crop_summary.index=='grapes']
values = [grapes_npk['N'][0], grapes_npk['P'][0], grapes_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Grapes", marker_colors=cafe_colors),1, 3)

orange_npk = crop_summary[crop_summary.index=='orange']
values = [orange_npk['N'][0], orange_npk['P'][0], orange_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Orange", marker_colors=cafe_colors),1, 4)

mango_npk = crop_summary[crop_summary.index=='mango']
values = [mango_npk['N'][0], mango_npk['P'][0], mango_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Mango", marker_colors=cafe_colors),1, 5)

coconut_npk = crop_summary[crop_summary.index=='coconut']
values = [coconut_npk['N'][0], coconut_npk['P'][0], coconut_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Coconut", marker_colors=cafe_colors),2, 1)

papaya_npk = crop_summary[crop_summary.index=='papaya']
values = [papaya_npk['N'][0], papaya_npk['P'][0], papaya_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Papaya", marker_colors=cafe_colors),2, 2)

pomegranate_npk = crop_summary[crop_summary.index=='pomegranate']
values = [pomegranate_npk['N'][0], pomegranate_npk['P'][0], pomegranate_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Pomegranate", marker_colors=cafe_colors),2, 3)

```

```

watermelon_npk = crop_summary[crop_summary.index=='watermelon']
values = [watermelon_npk['N'][0], watermelon_npk['P'][0], watermelon_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values, name="Watermelon", marker_colors=cafe_colors), 2, 4)

muskmelon_npk = crop_summary[crop_summary.index=='muskmelon']
values = [muskmelon_npk['N'][0], muskmelon_npk['P'][0], muskmelon_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values, name="Muskmelon", marker_colors=cafe_colors), 2, 5)

fig.update_layout(
    title_text="NPK ratio for fruits",
    annotations=[dict(text='Apple', x=0.06, y=1.08, font_size=15, showarrow=False),
                  dict(text='Banana', x=0.26, y=1.08, font_size=15, showarrow=False),
                  dict(text='Grapes', x=0.50, y=1.08, font_size=15, showarrow=False),
                  dict(text='Orange', x=0.74, y=1.08, font_size=15, showarrow=False),
                  dict(text='Mango', x=0.94, y=1.08, font_size=15, showarrow=False),
                  dict(text='Coconut', x=0.06, y=0.46, font_size=15, showarrow=False),
                  dict(text='Papaya', x=0.26, y=0.46, font_size=15, showarrow=False),
                  dict(text='Pomegranate', x=0.50, y=0.46, font_size=15, showarrow=False),
                  dict(text='Watermelon', x=0.74, y=0.46, font_size=15, showarrow=False),
                  dict(text='Muskmelon', x=0.94, y=0.46, font_size=15, showarrow=False)])
fig.show()

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:04.773001Z","iopub.execute_input":"2021-10-03T08:53:04.773326Z","iopub.status.idle":"2021-10-03T08:53:05.65346Z","shell.execute_reply.started":"2021-10-03T08:53:04.773279Z","shell.execute_reply":"2021-10-03T08:53:05.652446Z"}}
crop_scatter = cropdf[(cropdf['label']=='rice') |
                      (cropdf['label']=='jute') |
                      (cropdf['label']=='cotton') |
                      (cropdf['label']=='maize') |
                      (cropdf['label']=='lentil')]

fig = px.scatter(crop_scatter, x="temperature", y="humidity", color="label", symbol="label")
fig.update_layout(plot_bgcolor='white')
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:05.654546Z","iopub.execute_input":"2021-10-03T08:53:05.654765Z","iopub.status.idle":"2021-10-03T08:53:05.747489Z","shell.execute_reply.started":"2021-10-03T08:53:05.65474Z","shell.execute_reply":"2021-10-03T08:53:05.746906Z"}}
fig = px.bar(crop_summary, x=crop_summary.index, y=["rainfall", "temperature", "humidity"])
fig.update_layout(title_text="Comparision between rainfall, temerature and humidity",
                  plot_bgcolor='white',
                  height=500)

fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()

# %% [markdown]
# <h3 style='color:Red'>Correlation between different features </h3>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:05.748644Z","iopub.execute_input":"2021-10-03T08:53:05.749058Z","iopub.status.idle":"2021-10-03T08:53:06.321277Z","shell.execute_reply.started":"2021-10-03T08:53:05.749016Z","shell.execute_reply":"2021-10-03T08:53:06.32022Z"}}
fig, ax = plt.subplots(1, 1, figsize=(15, 9))

```

```

sns.heatmap(cropdf.corr(), annot=True,cmap='Wistia' )
ax.set(xlabel='features')
ax.set(ylabel='features')

plt.title('Correlation between different features', fontsize = 15, c='black')
plt.show()

# %% [markdown]
# <h1 style='color:blue'>Declare independent and target variables</h1>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:06.322599Z","iopub.execute_input":"2021-10-03T08:53:06.3229Z","iopub.status.idle":"2021-10-03T08:53:06.3287Z","shell.execute_reply.started":"2021-10-03T08:53:06.322868Z","shell.execute_reply":"2021-10-03T08:53:06.327738Z"}}
X = cropdf.drop('label', axis=1)
y = cropdf['label']

# %% [markdown]
# <h1 style='color:blue'>Split dataset into training and test set</h1>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:06.330339Z","iopub.execute_input":"2021-10-03T08:53:06.331206Z","iopub.status.idle":"2021-10-03T08:53:06.513542Z","shell.execute_reply.started":"2021-10-03T08:53:06.331112Z","shell.execute_reply":"2021-10-03T08:53:06.512547Z"}}
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,
                                                    shuffle = True, random_state = 0)

# %% [markdown] {"execution":{"iopub.status.busy":"2021-09-19T12:16:20.708318Z","iopub.execute_input":"2021-09-19T12:16:20.708652Z","iopub.status.idle":"2021-09-19T12:16:20.715198Z","shell.execute_reply.started":"2021-09-19T12:16:20.708621Z","shell.execute_reply":"2021-09-19T12:16:20.714015Z"}}
# <h1 style='color:blue'>LightGBM Model Building and Training</h1>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:06.514703Z","iopub.execute_input":"2021-10-03T08:53:06.515535Z","iopub.status.idle":"2021-10-03T08:53:08.167436Z","shell.execute_reply.started":"2021-10-03T08:53:06.515493Z","shell.execute_reply":"2021-10-03T08:53:08.166446Z"}}
# build the lightgbm model
import lightgbm as lgb

model = lgb.LGBMClassifier()
model.fit(X_train, y_train)

# %% [markdown]
# <h1 style='color:blue'>Model Prediction</h1>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:08.168646Z","iopub.execute_input":"2021-10-03T08:53:08.168884Z","iopub.status.idle":"2021-10-03T08:53:08.209424Z","shell.execute_reply.started":"2021-10-03T08:53:08.168857Z","shell.execute_reply":"2021-10-03T08:53:08.208407Z"}}
# predict the results
y_pred=model.predict(X_test)

# %% [markdown]
# <h1 style='color:blue'>View Accuracy</h1>

# %% [code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:08.210738Z","iopub.execute_input":"2021-10-03T08:53:08.211561Z","iopub.status.idle":"2021-10-03T08:53:08.219638Z","shell.execute_reply.started":"2021-10-03T08:53:08.211521Z","shell.execute_reply":"2021-10-03T08:53:08.218985Z"}}
# view accuracy
from sklearn.metrics import accuracy_score

```

```

accuracy=accuracy_score(y_pred, y_test)
print('LightGBM Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test, y_pred)))

# %%[markdown]
# <h1 style='color:blue'>Compare train and test set accuracy</h1>

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:08.220696Z","iopub.execute_input":"2021-10-03T08:53:08.222534Z","iopub.status.idle":"2021-10-03T08:53:08.310553Z","shell.execute_reply.started":"2021-10-03T08:53:08.222487Z","shell.execute_reply":"2021-10-03T08:53:08.309874Z"}}
y_pred_train = model.predict(X_train)
print('Training-set accuracy score: {0:0.4f}'.format(accuracy_score(y_train, y_pred_train)))

# %%[markdown]
# <h1 style='color:blue'>Check for Overfitting</h1>

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:08.314037Z","iopub.execute_input":"2021-10-03T08:53:08.316113Z","iopub.status.idle":"2021-10-03T08:53:08.437898Z","shell.execute_reply.started":"2021-10-03T08:53:08.316064Z","shell.execute_reply":"2021-10-03T08:53:08.43708Z"}}
# print the scores on training and test set

print('Training set score: {:.4f}'.format(model.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(model.score(X_test, y_test)))

# %%[markdown]
# <h1 style='color:blue'>Confusion-matrix</h1>

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:08.441822Z","iopub.execute_input":"2021-10-03T08:53:08.442437Z","iopub.status.idle":"2021-10-03T08:53:10.650801Z","shell.execute_reply.started":"2021-10-03T08:53:08.442395Z","shell.execute_reply":"2021-10-03T08:53:10.649662Z"}}
# view confusion-matrix
# Print the Confusion Matrix and slice it into four pieces

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(15,15))
sns.heatmap(cm, annot=True, fmt=".0f", linewidths=.5, square = True, cmap = 'Blues');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
all_sample_title = 'Confusion Matrix - score:'+str(accuracy_score(y_test,y_pred))
plt.title(all_sample_title, size = 15);
plt.show()

# %%[markdown]
# <h1 style='color:blue'>Classification Metrics</h1>

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:10.651946Z","iopub.execute_input":"2021-10-03T08:53:10.652191Z","iopub.status.idle":"2021-10-03T08:53:10.691621Z","shell.execute_reply.started":"2021-10-03T08:53:10.652162Z","shell.execute_reply":"2021-10-03T08:53:10.690594Z"}}
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

# %%[code] {"execution":{"iopub.status.busy":"2021-10-03T08:53:10.692823Z","iopub.execute_input":"2021-10-03T08:53:10.693073Z","iopub.status.idle":"2021-10-03T08:53:10.705786Z","shell.execute_reply.started":"2021-10-03T08:53:10.693044Z","shell.execute_reply":"2021-10-03T08:53:10.705116Z"}}
newdata=model.predict([[ 90, 42, 43, 20.879744, 75, 5.5,220]])
newdata

```

ДОДАТОК В
АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

МАТЕРІАЛИ VI МІЖНАРОДНОЇ
СТУДЕНТСЬКОЇ НАУКОВОЇ
КОНФЕРЕНЦІЇ

РОЗВИТОК
СУСПІЛЬСТВА ТА НАУКИ
В УМОВАХ ЦИФРОВОЇ
ТРАНСФОРМАЦІЇ



М. ІВАНО-ФРАНКІВСЬК,
УКРАЇНА

**31 ТРАВНЯ
2024 РІК**



**УДК 082:001
Р 64**



Голова оргкомітету: Кореньюк І.О.
Верстка: Зрада С.І.
Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 38 від 30.05.2024 року.



Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення №31 від 05.01.2024).

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Р 64

Розвиток суспільства та науки в умовах цифрової трансформації: матеріали VI Міжнародної студентської наукової конференції, м. Івано-Франківськ, 31 травня, 2024 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2024. — 510 с.

ISBN 978-617-8312-53-4

DOI 10.62732/liga-inter-31.05.2024

Викладено матеріали учасників VI Міжнародної мультидисциплінарної студентської наукової конференції «Розвиток суспільства та науки в умовах цифрової трансформації», яка відбулася 31 травня 2024 року у місті Івано-Франківськ, Україна.

УДК 082:001

© Колектив учасників конференції, 2024
© ГО «Молодіжна наукова ліга», 2024
© ТОВ «УКРЛОГОС Груп», 2024

ISBN 978-617-8312-53-4

DEVELOPMENT OF A WEBSITE FOR CUSTOMER SERVICE WITH EXTERNAL CALLS

Platonov D., *Scientific supervisor: Kuzomin O.* 299

ВСТАНОВЛЕННЯ ВИМОГ ПРИ УДОСКОНАЛЕННІ СИСТЕМИ ОБМІНУ ПОВІДОМЛЕННЯМИ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Гробовий Д.В., *Науковий керівник: Безкоровайний В.В.*..... 302

ІГРОВІ ТЕХНОЛОГІЇ ЯК ІНСТРУМЕНТ ПРОФЕСІЙНОЇ ОРІЄНТАЦІЇ АБІТУРІЄНТІВ ІТ-СПЕЦІАЛЬНОСТЕЙ

Коваль М.М., *Науковий керівник: Ковалюк Т.В.* 306

СЕКЦІЯ 14.

СИСТЕМНИЙ АНАЛІЗ, МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ

АНАЛІЗ ОСНОВНИХ ТЕНДЕНЦІЙ У ДЕМОГРАФІЧНИХ ПРОЦЕСАХ В КРАЇНАХ ЄС

Ратушинська І.В. 310

СЕКЦІЯ 15.

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

FRONTEND TECHNOLOGIES IN MODERN DEVELOPMENT

Palamar M.V., Sukhankova D.S., *Scientific advisor: Zapadynska I.H.*..... 314

THE PSYCHOLOGY OF COLOR IN WEB DESIGN: CHOOSING COLORS FOR EFFECTIVE USER INTERACTION

Inzhyievska V.A., Katrychuk I.O., *Scientific supervisor: Zapadynska I.H.* 317

THE ROLE OF ARTIFICIAL INTELLIGENCE IN THE DEVELOPMENT OF SOFTWARE

Bilyk I.D., *Scientific supervisor: Zapadynska I.H.* 320

АЛГОРИТМ РЕКОМЕНДАЦІЙ ДЛЯ АГРОКУЛЬТУР НА ОСНОВІ ДАНИХ ПРО ҐРУНТ ТА КЛІМАТИЧНІ УМОВИ З ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИНОГО НАВЧАННЯ

Потапчук А., *Науковий керівник: Биковий П.Є.* 322

АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ СУПРОВОДУ ТОВАРІВ В ІНТЕРНЕТ-МАГАЗИНІ

Баран А.Т., *Науковий керівник: Кім І.Р.* 324

ВИКОРИСТАННЯ МУЛЬТИ-АГЕНТНИХ СИСТЕМ ДЛЯ ДОСЛІДЖЕННЯ НЕВІДОМИХ СЕРЕДОВИЩ

Сагайдачний О.М., *Науковий керівник: Аксак Н.Г.* 327

ВИКОРИСТАННЯ ТАБЛИЧНОГО ПРОЦЕСОРА MS EXCEL ДЛЯ РОЗРАХУНКУ АМОРТИЗАЦІЙНИХ ВІДРАХУВАНЬ

Маслій А.А., *Науковий керівник: Дивак В.В.* 330

Потапчук Артур, здобувач вищої освіти факультету
комп'ютерних інформаційних технологій
Західноукраїнський національний університет, Україна

Науковий керівник: Биковий П.Є., канд. техн. наук, доцент, доцент
кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

АЛГОРИТМ РЕКОМЕНДАЦІЙ ДЛЯ АГРОКУЛЬТУР НА ОСНОВІ ДАНИХ ПРО ҐРУНТ ТА КЛІМАТИЧНІ УМОВИ З ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

У сучасному світі агрономії велике значення має використання даних для оптимізації процесів вирощування рослин. Аналіз даних дозволяє глибше зрозуміти потреби різних культур у мінералах та оптимальних умовах для їх росту, що, в свою чергу, сприяє підвищенню урожайності та стійкості рослин до несприятливих умов. Використання таких технологій, як Python та різноманітні бібліотеки для обробки даних та візуалізації, відкриває перед агрономами нові можливості для проведення глибокого аналізу та прийняття обґрунтованих рішень. Цей підхід не тільки сприяє ефективному управлінню ресурсами, але й допомагає адаптуватися до змін клімату, зберігаючи при цьому стабільність виробництва продуктів харчування.

Покроковий алгоритм для аналізу даних про рекомендації щодо вирощування рослин, використовуючи Python та різні бібліотеки для обробки даних та візуалізації (Рисунок 1):

1. Імпорт бібліотек: Завантажте необхідні бібліотеки (pandas, numpy, matplotlib, seaborn, plotly).

2. Завантаження даних: Завантажте набір даних, використовуючи `pandas.read_csv()` і покажіть перші кілька рядків, щоб перевірити, чи все завантажено коректно.

3. Перевірка на наявність пропущених значень: Використовуйте `isnull()` та `any()` для перевірки наявності пропущених даних у наборі.

4. Візуалізація даних: Створіть візуалізації за допомогою matplotlib, seaborn та plotly, щоб зрозуміти розподіл та кореляції між різними ознаками. Відобразіть вміст нітрогену, фосфору, калію та інші характеристики за різними культурами.

5. Обробка даних: Поділіть дані на незалежні (X) та залежні (y) змінні. Поділіть набір даних на тренувальні та тестові дані.

6. Моделювання: Використовуйте модель, наприклад LightGBM, для тренування на тренувальних даних.

7. Прогнозування та оцінка моделі: Використовуйте модель для прогнозування на тестових даних і оцініть точність моделі за допомогою метрик, таких як точність класифікації та матриця помилок.

8. Відображення результатів: Покажіть результати класифікації та інші візуалізації для кращого розуміння результатів моделі.

9. Аналіз результатів: Проведіть подальший аналіз результатів, зокрема порівняння важливості ознак та перевірку моделі на перенавчання.

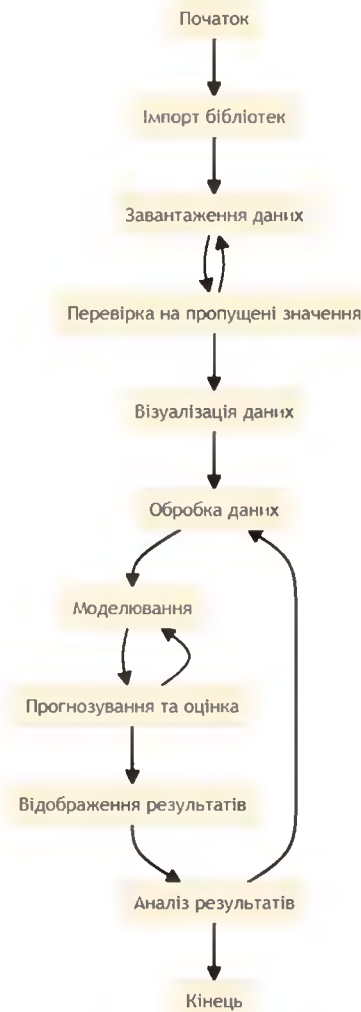


Рис. 1. Алгоритм для аналізу даних про рекомендації щодо вирощування рослин

Цей алгоритм допомагає зрозуміти та проаналізувати, як різні фактори впливають на вибір культур для вирощування, базуючись на їх вимогах до мікроелементів та умов зростання.

Список використаних джерел:

1. Архітектура програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій / М. Дивак та ін. Measuring and computing devices in technological processes. 2024. № 1. С. 125–139. URL: <https://doi.org/10.31891/2219-9365-2024-77-15> (дата звернення: 13.05.2024).