МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

МИСЬ Андрій Русланович

Інтелектуальний модуль прогнозування врожайності ягід / Intelligent berry yield prediction module

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Дипломний проект

Виконав студент групи КН-41
А. Р. Мись

Науковий керівник:
к.т.н., доцент Т.В. Лендюк

Дипломний проект допущено до захисту
«___»_____ 2023 р.

Завідувач кафедри
_____ М.П. Комар

Тернопіль – 2023

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
«____» _____2022р.

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ

Мисю Андрію Руслановичу
(прізвище, ім'я, по батькові)

1. Тема проекту: Інтелектуальний модуль прогнозування врожайності ягід / Intelligent berry yield prediction module
керівник проекту к.т.н., доцент Т.В. Лендюк

затверджені наказом по університету від 08 грудня 2022 р. № 491.

2. Строк подання студентом закінченого проекту 01 червня 2023 р.

3. Вихідні дані до проекту: технічне завдання.

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

- Вивчення та аналіз сфери дослідження
- Перегляд та аналіз існуючих рішень в даній області
- Оцінка методології ансамблевого навчання
- Розгляд XGBoost, LightGBM та CatBoost як потенційних інструментів для прогнозування врожайності ягід
- Розробка алгоритму навчання моделей для прогнозування врожайності ягід
- Розробка алгоритму валідації моделей для перевірки точності прогнозування
- Реалізація та валідація ансамблевого навчання моделей для прогнозування врожайності ягід

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- алгоритм навчання моделей для прогнозування врожайності ягід
- алгоритм валідації моделей для перевірки точності прогнозування

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Н. контроль	к.т.н., доцент Т.В. Лендюк		

7. Дата видачі завдання 08 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області і постановка задачі дослідження	30.12.2022	
2	Алгоритмічне забезпечення інтелектуального модуля прогнозування врожайності ягід	24.03.2023	
3	Програмно-технологічне забезпечення інтелектуального модуля прогнозування врожайності ягід	12.05.2023	
4	Повне завершення та оформлення дипломного проекту	01.06.2023	

Студент _____ А. Р. Мись
(підпис)

Керівник проекту_____ Т.В. Лендюк
(підпис)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 62 с., 7 рис., 1 табл., 2 додатки, 48 джерел.

Метою дипломного проекту є розробка інтелектуального модуля прогнозування врожайності ягід, заснованого на алгоритмах машинного навчання.

Об'єктом дослідження є процес прогнозування врожайності ягід.

Предметом дослідження є використання ансамблевих моделей машинного навчання, зокрема XGBoost, LightGBM та CatBoost, для створення точного прогнозу врожайності ягід.

Розроблено та досліджено програмне забезпечення для інтелектуального модуля прогнозування врожайності ягід.

Розроблений модуль може допомогти фермерам і аграріям планувати свою діяльність, оптимізувати витрати та підвищувати продуктивність. Модуль також може стати корисним інструментом для дослідників і аналітиків в агросекторі.

ПРОГНОЗУВАННЯ ВРОЖАЙНОСТІ ЯГІД, АЛГОРИТМИ МАШИННОГО НАВЧАННЯ, АНСАМБЛЕВЕ НАВЧАННЯ, XGBOOST, LIGHTGBM, CATBOOST, ВАЛІДАЦІЯ МОДЕЛЕЙ.

ABSTRACT

The bachelor's thesis report: 62 pages, 7 figures, 1 table, 2 appendices, 48 references.

The aim of the diploma project is to develop an intelligent module for predicting the yield of berries based on machine learning algorithms.

The object of the research is the process of predicting the yield of berries.

The subject of the research is the use of ensemble machine learning models, specifically XGBoost, LightGBM, and CatBoost, to create an accurate forecast of berry yield.

Software has been developed and studied for the intelligent module for predicting the yield of berries.

The developed module can assist farmers and agriculturalists in planning their activities, optimizing costs, and increasing productivity. The module can also be a useful tool for researchers and analysts in the agricultural sector.

BERRY YIELD PREDICTION, MACHINE LEARNING ALGORITHMS, ENSEMBLE LEARNING, XGBOOST, LIGHTGBM, CATBOOST, MODEL VALIDATION.

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Інтелектуальний модуль прогнозування врожайності ягід.

1.2 Область застосування – сільське господарство та агросектор.

2. ОСНОВА ДЛЯ РОЗРОБЛЕННЯ

Основою для розроблення є завдання на дипломний проект, затверджене кафедрою інформаційно-обчислювальних систем і управління факультету комп’ютерних інформаційних технологій Західноукраїнського національного університету.

3. ПРИЗНАЧЕННЯ РОЗРОБЛЕНОГО КОМПЛЕКСУ

Метою дипломного проекту є розробка інтелектуального модуля прогнозування врожайності ягід, заснованого на алгоритмах машинного навчання. Цей модуль покликаний підвищити точність прогнозування врожайності, враховуючи велику кількість взаємопов'язаних факторів.

4. ДЖЕРЕЛА РОЗРОБЛЕННЯ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні статті, журнали, Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Основні функціональні вимоги до програмної системи:

– вибір даних (пакетних даних / даних у реальному часі) з різних архітектур великих даних;

– провести попередню обробку даних для інтелектуального аналізу тексту.

– Реалізувати та провести валідацію ансамблевого навчання моделей для прогнозування врожайності ягід.

5.2 Вимоги до апаратних засобів:

– кластер запускається на фізичній машині Macbook Pro з 16 гігабайт оперативної пам’яті та процесором 2.3 GHz Intel Core i5.

5.3 Вимоги до програмних засобів:

– для розробки програмне забезпечення - Python 3.7;

– для створення графічного інтерфейсу користувача використано – tkinter, Adobe XD;

– для реалізації моделей навчання – фреймворк XGBoost, LightGBM та CatBoost.

6. ПОРЯДОК КОНТРОЛЮ

6.1 Представлення дипломного проекту на попередній захист.

6.2 Представлення дипломного проекту на захист.

Завдання прийняв до виконання _____ А. Р. Мись

(підпис) (прізвище та ініціали)

Керівник дипломного проекту_____ Т.В. Лендюк

(підпис) (прізвище та ініціали)

ЗМІСТ

ВСТУП

Актуальність теми. Прогнозування врожайності ягід є важливим аспектом сільськогосподарської промисловості. Це не тільки допомагає фермерам у плануванні та оптимізації виробництва, але також сприяє стабільності ринку, забезпечуючи достатній обсяг продукції для задоволення попиту.

Однак, врожайність ягід може відчутно відрізнятися від року до року через різноманітність факторів, включаючи погодні умови, шкідники та хвороби, а також практики управління. Тому розробка надійних методів прогнозування врожайності стає все більш актуальною.

Використання технологій штучного інтелекту, таких як алгоритми машинного навчання, може значно покращити точність прогнозів, враховуючи велику кількість факторів та їх взаємодію.

Це особливо актуально в умовах глобальних змін клімату, які можуть призвести до непередбачуваних відхилень у врожайності ягід. За допомогою штучного інтелекту, фермери та сільськогосподарські підприємства можуть адаптуватися до цих змін, оптимізуючи свої стратегії для забезпечення максимальної врожайності.

Таким чином, розробка інтелектуального модуля прогнозування врожайності ягід є актуальною та важливою задачею, яка може мати значний вплив на сільськогосподарську промисловість та продовольчу безпеку.

Метою дипломного проекту є розробка інтелектуального модуля прогнозування врожайності ягід, заснованого на алгоритмах машинного навчання. Цей модуль покликаний підвищити точність прогнозування врожайності, враховуючи велику кількість взаємопов'язаних факторів.

Для досягнення поставленої мети було необхідно провести ряд теоретичних досліджень, визначити та послідовно вирішити наступні завдання:

1. Вивчення та аналіз сфери дослідження

2. Перегляд та аналіз існуючих рішень в даній області
3. Оцінка методології ансамблевого навчання
4. Розгляд XGBoost, LightGBM та CatBoost як потенційних інструментів для прогнозування врожайності ягід
5. Розробка алгоритму навчання моделей для прогнозування врожайності ягід
6. Розробка алгоритму валідації моделей для перевірки точності прогнозування
7. Реалізація та валідація ансамблевого навчання моделей для прогнозування врожайності ягід

Об'єктом дослідження є процес прогнозування врожайності ягід**.**

Предметом дослідження є використання ансамблевих моделей машинного навчання, зокрема XGBoost, LightGBM та CatBoost, для створення точного прогнозу врожайності ягід.

Методи дослідження. Використовуються методи аналізу даних, ансамблевого машинного навчання та крос-валідації. Основні інструменти - XGBoost, LightGBM та CatBoost, використовуються для розробки і валідації моделей..

Практичне значення дипломного проекту полягає в розробці інтелектуального модуля прогнозування врожайності ягід. Цей модуль може допомогти фермерам і аграріям планувати свою діяльність, оптимізувати витрати та підвищувати продуктивність. Модуль також може стати корисним інструментом для дослідників і аналітиків в агросекторі.

Структура та обсяг роботи. Дипломний проект складається з вступу, трьох розділів, висновків та списку використаних джерел. Загальний обсяг роботи становить 62 сторінок комп'ютерного тексту, включаючи 7 рисунків та 1 таблицю. У списку використаних джерел наведено 48 найменувань, які займають 6 сторінок

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області

Метою інтелектуального модуля прогнозування врожайності ягід є аналіз та прогнозування врожайності лохини. Лохина є плодовою рослиною, яка вирощується для отримання ягід, які мають велике значення в харчовій та фармацевтичній промисловості.

Дика лохина є найважливішою плодовою культурою штату Мен, яка росте на гірських кислих піщаних ґрунтах. Дика лохина (також відома як низькокущова лохина) є найважливішою культурою для більшості людей і вважається найбільшим виробником серед інших культур у США [1] . Головна перевага дикої лохини перед культурною полягає в тому, що вона не садиться, а росте природним шляхом на кам’янистих гірках і лісах. Культурну лохину, навпаки, необхідно садити [2] . Дика чорниця дає різноманітні культури, оскільки вона захищена від селекційної селекції та генної інженерії. Ручні комбайни використовують граблі, щоб зішкребти ягоди з рослин [3]через пересічену місцевість і висоту дикоростучої чорниці. Найкращий час для збору дикої чорниці в штаті Мен — з кінця липня до початку вересня. [4] . Види комах-запилювачів і погодні умови є основними факторами, що впливають на врожайність дикої лохини [5] . Кількість комах-запилювачів, оптимальний час для запилення дикої лохини (близько 6:00 – 17:00), види бджіл (медоносна бджола (HB), джміль (BB), Андрена (AD) і Osmia (OS)) та погодні умови. (низькі температури, волога погода, сильний вітер, заморозки та підвищення температури) є факторами, що впливають на врожайність чорниці дикої [6]. Отримати надійні прогнози врожайності для досягнення цілей Продовольчої та сільськогосподарської організації (FAO) важко. Основною метою ФАО є викорінення глобального голоду. Кілька

вчених зосередилися на точному землеробстві шляхом прогнозування врожайності за допомогою методології машинного навчання (ML) [7] – [10] . Однак лише кілька досліджень використовували EMLA з різними топологіями [11] , [12] . Для MLA потрібні великі та високоякісні набори даних для точної оцінки врожайності [13] . Моделювання систем сільськогосподарського виробництва має важливе значення, особливо коли збір великомасштабних часових і географічних даних є надзвичайно дорогим, складним або в деяких випадках неможливим.

У кількох дослідженнях якість даних, які використовуються під час навчання, є недостатньою, що може вплинути на результати експерименту [14] , що погіршує точність прогнозу врожайності [15] . Щоб вирішити ці проблеми, були використані підходи комп'ютерного імітаційного моделювання, які імітують вимірювання на основі сцен для створення даних [16] . Для підтвердження комп'ютерної імітаційної моделі тестуються припущення моделі. Аналіз чутливості використовується для припущень даних. Його використовували для демонстрації інтерактивного впливу домінування бджолиної спільноти та метеорологічних змінних на вихід дикої чорниці [17]. Центр моделювання, експериментів і ефективного дизайну обробки даних у Військово-морських аспірантурах визначає обробку даних як практику накопичення наборів даних за допомогою моделювання та комп'ютерного моделювання.

Таким чином, мета обробки даних полягає в тому, щоб дати особам, які приймають рішення, зрозуміти складні проблеми за допомогою моделювання для генерування даних [18] .

Obsie та ін. [17] використовували обчислювальні експерименти для отримання навчальних даних, які згодом можуть бути використані для навчання та оцінки MLA.

Для емпіричних наборів даних EMLA перевершила індивідуальні MLA з точки зору точності. Основна перевага використання EMLA полягає в тому, що вони можуть вирішувати складні проблеми [19] . Основна ідея

використання EMLA полягає в тому, щоб перетворити слабкого учня на потужного для покращення підтримки прийняття рішень [20] . Кілька досліджень поєднували різні методи МН для досягнення високої точності. Методи стекування використовуються для реалізації моделей класифікації, таких як PredT4SE-Stack [21] і NeuroPpred-Fuse [22] , а також зважених методів стекування кількох метамоделей [23] для регресії. Кілька методів з різними архітектурами використовувалися для допомоги точному землеробству через прогнозування врожайності [12]. Техніка стекування отримує всі вихідні дані передбачення з інших MLA і передає їх у метамодель для отримання найкращого рішення. Автори [17] довели, що дані моделювання в поєднанні з кліматичними даними корисні для навчання MLA, оскільки ці дані мають високу якість і вони створюють набори даних, які важко зібрати вручну. Основна мета полягає в тому, щоб уникнути дорогих наборів даних у реальному часі, отриманих із супутникових зображень із середньою вартістю США$ 1/км ^{2} [24] .

Цей модуль використовує дані про різні фактори, що впливають на врожайність лохини, такі як розмір клону лохини, густота бджіл (медоносних та диких), температурні умови, кількість днів з опадами та інші. З використанням цих даних модуль проводить аналіз та прогнозує врожайність лохини.

Аналіз предметної області дозволяє зрозуміти, які фактори мають найбільший вплив на врожайність лохини і як ці фактори взаємодіють між собою. Це допомагає визначити оптимальні умови для вирощування лохини та розробити стратегії для підвищення врожайності.

Інтелектуальний модуль прогнозування врожайності ягід лохини є потужним інструментом для сільськогосподарських досліджень, розробки нових методів вирощування та оптимізації врожайності лохини.

1.2 Огляд існуючих рішень

У країнах, що розвиваються, урожайність зазвичай оцінюють вручну. Однак інші країни зосередилися на використанні нових технологій, включаючи MLA та алгоритми глибокого навчання, для прогнозування врожайності.

Протягом 30 років на західному узбережжі Індії збиралися часові ряди даних про врожайність рису в поєднанні з метеорологічними характеристиками. Чотири MLA були використані для прогнозування врожайності рису. Загалом, оператор найменшого абсолютного скорочення та відбору (LASSO) досяг найкращих результатів [25] .

Гібридна множинна лінійна регресія (MLR) – штучна нейронна мережа (ANN) була використана для прогнозування врожайності [26] . У цій мережі, замість довільної ініціалізації ваги та зміщення, ваги вхідного рівня та зміщення ініціалізуються з використанням коефіцієнтів MLR та зміщення. Площа посіву, регіон водної системи, споживання добрив і точки інтересу водної системи були включені до створених сільськогосподарських даних. Серед кліматичних характеристик, включених до набору даних, були кліматичні опади, найвищі та найнижчі температури та сонячна радіація. MLR–ANN було застосовано до цих наборів даних для точного прогнозування врожайності рису.

П’ять MLA, а саме лінійна регресія (LR), LASSO, XGBoost, LGBM і випадковий ліс (RF), були об’єднані за допомогою оптимізованого зваженого ансамблю для прогнозування даних про врожайність кукурудзи, включаючи управління культурами, клімат, умови ґрунту та історичну врожайність. П’ять MLA у поєднанні з оптимізованим зваженим ансамблем перевершили існуючі MLA [12] .

Шаххоссейні та ін. [27] представили нову ансамблеву згортову нейронну мережу (CNN). Крім того, глибокі нейронні мережі використовувалися для прогнозування врожайності кукурудзи. Було розглянуто набір даних, що

містить дату посіву, дані про погоду, характеристики ґрунту та історичну врожайність кукурудзи.

Урожайність зернових була передбачена за допомогою MLA, навченого з набором даних, що містить дані про погоду, дистанційне зондування, клімат і дані про врожайність [28] , і набір даних був об'єднаний із супутниковими даними та історичною врожайністю. Ці чотири вимірювання параметрів були об'єднані, щоб сформувати єдиний набір даних. Було використано одну лінійну MLA і три нелінійні MLA. XGBoost перевершив інші MLA для цієї програми.

Лінійні та нелінійні моделі були використані для оцінки врожайності сої та кукурудзи за допомогою зображень Landsat та SPOT. Було вивчено дві моделі, нейронну мережу та MLR [29] , і результати показали, що коли були включені всі зображення, моделі створили сильну відповідність, перевищуючи нейронні мережі з точки зору точності.

Хакі та ін. [30] представили YieldNet, яка є сучасною моделлю CNN, яка поєднує в собі передбачувану врожайність кукурудзи та сої шляхом перемикання ваги хребта, включаючи екстрактор, використовуючи новий метод глибокого навчання. Набір даних було створено за допомогою даних дистанційного зондування, що дозволяє безперервно контролювати культуру протягом усього циклу росту.

Набори даних були створені за допомогою імітаційної моделі запилення дикої чорниці. Ця інформація містить дані про чотири види бджіл, розміри їхніх клонів (CS) і погодні дані. Для прогнозування врожайності дикої чорниці використовували чотири MLA. XGBoost [17] досяг найкращої продуктивності за всіма показниками оцінювання.

В процесі аналізу було зроблено огляд декількох досліджень, в яких використовувалися дані моделювання для прогнозування врожайності. Зокрема, статті [17] та [31] досліджували врожайність кукурудзи протягом 5 років за допомогою симулятора систем сільськогосподарського виробництва в Середньому Заході США.

Ці дослідження показали значну перевагу використання ансамблевого моделювання при прогнозуванні врожайності. Ансамблеві моделі комбінують прогнози кількох базових моделей, що дозволяє отримати більш точний та стабільний результат. Вони враховують різноманітність даних та різні підходи до моделювання, що сприяє зменшенню помилок та підвищенню точності прогнозування.

Ансамблеві моделі мають широке застосування в сільському господарстві та дозволяють вирішувати складні задачі, пов'язані з прогнозуванням врожайності. Використання таких моделей допомагає зрозуміти вплив різних факторів на врожайність та розробити ефективні стратегії вирощування.

Таким чином, ансамблеве моделювання є потужним інструментом у дослідженнях прогнозування врожайності, оскільки воно забезпечує більш точні та надійні результати, що можуть бути використані для покращення сільськогосподарського виробництва.

1.3 Оцінка підходів ансамблевого навчання

Ансамблювання є потужним підходом в машинному навчанні, який дозволяє поєднувати прогнози декількох моделей з метою отримання кращої і точнішої прогнозної здатності. Ідея полягає в тому, що комбінування прогнозів з різних моделей може знизити помилку та покращити загальну проціс прогнозування.

Основна мета ансамблювання полягає у використанні комбінації слабких моделей (тобто моделей з невеликою прогнозною здатністю) для створення потужної моделі, яка може робити кращі прогнози. Це досягається шляхом використання різних стратегій комбінування, які можуть варіюватися залежно від методу ансамблювання.

Основні методи ансамблювання включають:

1. Бутстреп-агрегація (Bagging): цей метод полягає у використанні декількох незалежних моделей навчання на різних підмножинах тренувальних даних замість однієї моделі.

   Переваги бутстреп-агрегації (Bagging):

   - Зменшення дисперсії: Використання декількох незалежних моделей дозволяє зменшити варіацію прогнозів. Кожна модель навчається на різних підмножинах даних, тому вони можуть виявляти різні взаємозв'язки в даних.
   - Уникнення перенавчання: Бутстреп-агрегація допомагає уникнути перенавчання, оскільки моделі навчаються на різних підмножинах даних. Це дозволяє їм більш загально узагальнювати залежності в даних.
   - Врахування різноманітності: Кожна модель навчається на випадково вибраних підмножинах даних, що сприяє різноманітності моделей. Це може поліпшити прогностичні здібності ансамблю.

   Недоліки бутстреп-агрегації:

   - Збільшення обчислювальних витрат: Оскільки бутстреп-агрегація використовує декілька незалежних моделей, це може призвести до збільшення обчислювальних витрат, особливо якщо модель досить складна або навчається на великому обсязі даних.
   - Затримка в навчанні: Тренування декількох моделей може зайняти більше часу, ніж тренування однієї моделі. Це може бути проблемою, коли швидкість тренування є критичним фактором.
   - Втрата інтерпретованості: Бутстреп-агрегація використовує незалежні моделі, тому важче інтерпретувати результати прогнозів порівняно з однією моделлю. Це може бути проблемою, якщо потрібно зрозуміти вплив окремих змінних на прогнози.
   - Врахування ваги: Бутстреп-агрегація може надати вагу кожній моделі в ансамблі, залежно від її ефективності. Це дозволяє приділити

більшу увагу до добре працюючих моделей і меншу увагу до менш ефективних моделей.

- Використання для класифікації та регресії: Бутстреп-агрегація може застосовуватись як для задач класифікації, так і для задач регресії. Це робить його універсальним методом для прогнозування в різних задачах.
- Відповідність до паралельного обчислення: Бутстреп-агрегація може бути ефективно розпаралелена, оскільки кожна модель може бути навчена незалежно. Це дозволяє використовувати паралельні обчислення для зменшення часу навчання.
- Стійкість до шуму: Бутстреп-агрегація може бути стійким до шуму в даних, оскільки вона використовує середнє або голосування з кількох моделей. Вона може зменшити вплив випадкових помилок або викидів на кінцевий прогноз.

Ураховуючи ці переваги та недоліки, бутстреп-агрегація є потужним методом ансамблевого навчання, який може поліпшити якість прогнозів і допомогти уникнути перенавчання. Вона широко використовується в різних областях, включаючи машинне навчання, статистику і прогнозування.

2. Бустінг (Boosting): цей метод базується на послідовному навчанні слабких моделей навчання (наприклад, дерева рішень), при цьому кожна наступна модель намагається скоригувати помилки попередньої моделі. Переваги методу бустінгу:

- Висока точність: Бустінг зазвичай дає дуже точні результати, оскільки кожна наступна модель намагається скоригувати помилки попередньої моделі. Це дозволяє досягти високої точності прогнозів.
- Здатність до моделювання складних залежностей: Бустінг може моделювати складні залежності між змінними, оскільки він може використовувати композицію багатьох слабких моделей, які разом утворюють потужну прогностичну модель.

- Відсутність перенавчання: Бустінг зазвичай має менше схильності до перенавчання порівняно з іншими методами машинного навчання. Це досягається завдяки послідовному навчанню моделей і фокусуванню на скоригуванні помилок.
- Застосовується як для класифікації, так і для регресії: Бустінг може бути використаний як для задач класифікації, так і для задач регресії, що робить його універсальним методом для прогнозування в різних типах задач.

Недоліки методу бустінгу:

- Схильність до перенавчання з великою кількістю моделей: Якщо кількість моделей (ітерацій) в бустінгу велика, існує ризик перенавчання, особливо якщо дані мають велику кількість шуму або викидів.
- Вимоги до обчислювальних ресурсів: Бустінг може бути вимогливим до обчислювальних ресурсів, особливо якщо використовуються складні моделі або великий обсяг даних.
- Чутливість до випадкових помилок у даних: Бустінг може бути чутливим до випадкових помилок або неправильних даних.
- Вимога до налаштування параметрів: Бустінг має деякі параметри, які потребують налаштування, наприклад, кількість ітерацій або глибина дерев. Неправильне налаштування цих параметрів може призвести до недооптимізованих моделей або перенавчання.
- Потреба у великій кількості даних: Бустінг, особливо глибокі моделі, може потребувати велику кількість даних для досягнення хороших результатів. Відсутність достатньої кількості даних може обмежувати ефективність методу.
- Час навчання: Бустінг вимагає послідовного навчання моделей, що може займати багато часу, особливо при великій кількості ітерацій або складних моделях.

Враховуючи ці переваги та недоліки, бустінг залишається потужним методом ансамблевого навчання, який може забезпечувати високу точність та здатність моделювання складних залежностей. Однак, перед використанням бустінгу необхідно враховувати його обмеження та вимоги щодо даних та обчислювальних ресурсів.

3. Стекінг (Stacking): цей метод полягає в тому, щоб використовувати вихідні прогнози декількох моделей як вхідні дані для іншої моделі.

Переваги стекінгу:

- Комбінація сильних сторін моделей: Стекінг дозволяє поєднувати прогнози кількох моделей, використовуючи їх сильні сторони. Це може призвести до поліпшення прогнозів і зниження помилок.
- Здатність моделювати складні залежності: Завдяки використанню різних моделей і їх комбінації, стекінг може краще моделювати складні залежності в даних, що дозволяє отримати більш точні прогнози.
- Адаптивність до різних типів даних: Стекінг може працювати з різними типами моделей, такими як лінійні моделі, дерева рішень, або нейронні мережі. Це дозволяє використовувати найкращі моделі для конкретного типу даних і отримувати кращі результати.

Недоліки стекінгу:

- Складність налаштування: Використання стекінгу вимагає налаштування кількох моделей та визначення оптимального способу комбінування їх прогнозів. Це може бути часо- і ресурсоємним процесом.
- Підвищений ризик перенавчання: Якщо стекінг не належним чином налаштований, може виникнути ризик перенавчання, коли модель надмірно пристосовується до тренувальних даних і втрачає здатність узагальнювати до нових даних.
- Великі обчислювальні витрати: Стекінг включає в себе тренування кількох моделей та їх комбінування. Це може вимагати значних

обчислювальних ресурсів та часу, особливо якщо використовуються складні моделі або великі набори даних.

Загалом, стекінг є потужним засобом ансамблевого навчання, який може покращити точність прогнозування і моделювати складні залежності в даних. Однак, використання стекінгу вимагає налаштування моделей та може призвести до підвищеного ризику перенавчання. Також, використання стекінгу може бути обчислювально витратним процесом. Якщо належним чином налаштувати та застосувати стекінг, він може бути потужним інструментом для покращення прогнозування і отримання кращих результатів.

У загальному, ансамбль методів машинного навчання може допомогти підвищити точність прогнозування в порівнянні з окремими моделями. Бутстреп-агрегація (Bagging) може знизити варіацію та покращити стійкість моделі до перенавчання, тоді як бустінг (Boosting) може покращити точність шляхом зменшення помилок та підсилення слабких моделей навчання. Однак, бустінг може бути чутливим до шуму в даних та перенавчання, тому необхідно враховувати ці фактори при виборі підходів для ансамблю. Стекінг (Stacking) може покращити точність, проте він вимагає більш складного процесу навчання та може стати більш обчислювально витратним. Враховуючи ці фактори, бустінг може бути одним з найбільш ефективних підходів до ансамблю методів машинного навчання.

Існує кілька алгоритмів бустингу, які використовуються для підвищення точності моделей машинного навчання. Основні алгоритми бустингу включають:

1. AdaBoost (Adaptive Boosting): Цей алгоритм навчає послідовні слабкі класифікатори на даних, з фокусом на екземплярах, які були неправильно класифіковані попередніми моделями. Важливість кожного екземпляра змінюється залежно від того, наскільки часто він був неправильно класифікований, дозволяючи наступним моделям навчатися на важчих екземплярах.

2. Gradient Boosting: Цей алгоритм будує ансамбль моделей шляхом послідовного навчання нових моделей, які коригують помилки попередніх моделей. Він використовує градієнт функції втрати для пошуку оптимального напрямку корекції.
    - Gradient Boosting Machines (GBM): Це популярний варіант градієнтного бустингу, який використовує дерев'яні моделі рішень як базові моделі. Він навчає модель на резидуалах попередньої моделі і додає її прогнози до загального ансамблю.
    - XGBoost (Extreme Gradient Boosting): Це оптимізований варіант градієнтного бустингу, який включає ряд технічних поліпшень, що роблять його швидшим та ефективнішим. Він використовує регуляризацію, оптимізований процес навчання та автоматичний підбір параметрів для покращення точності моделі.
    - Існує кілька алгоритмів бустингу, які використовуються для підвищення точності моделей машинного навчання. Основні алгоритми бустингу включають:
    - AdaBoost (Adaptive Boosting): Цей алгоритм навчає послідовні слабкі класифікатори на даних, з фокусом на екземплярах, які були неправильно класифіковані попередніми моделями. Важливість кожного екземпляра змінюється залежно від того, наскільки часто він був неправильно класифікований, дозволяючи наступним моделям навчатися на важчих екземплярах.
    - Gradient Boosting: Цей алгоритм будує ансамбль моделей шляхом послідовного навчання нових моделей, які коригують помилки попередніх моделей. Він використовує градієнт функції втрати для пошуку оптимального напрямку корекції.
    - Gradient Boosting Machines (GBM): Це популярний варіант градієнтного бустингу, який використовує дерев'яні моделі рішень як базові моделі. Він навчає модель на резидуалах попередньої моделі і додає її прогнози до загального ансамблю.

- XGBoost (Extreme Gradient Boosting): Це оптимізований варіант градієнтного бустингу, який включає ряд технічних поліпшень, що роблять його швидшим та ефективнішим. Він використовує регуляризацію, оптимізований процес навчання та автоматичний підбір параметрів для покращення точності моделі.

Для прогнозування врожайності ягід, бустингові алгоритми є дуже ефективними і потужними інструментами. Зокрема, XGBoost, LightGBM і CatBoost є популярними алгоритмами бустингу, які можуть бути використані для досягнення високої точності та надійних прогнозів.

XGBoost є оптимізованим варіантом градієнтного бустингу, який включає ряд технічних поліпшень і автоматичний підбір параметрів. Він є потужним і гнучким алгоритмом з високою швидкістю та точністю прогнозування.

LightGBM також є оптимізованим варіантом градієнтного бустингу, який прагне досягти високої швидкості та ефективності. Він особливо підходить для великих наборів даних, оскільки використовує гістограмний підхід та зменшує вимоги до пам'яті.

CatBoost є алгоритмом бустингу, спеціалізованим на роботі з категоріальними ознаками. Він автоматично обробляє категоріальні дані і показує добрі результати навіть без їх попереднього кодування. CatBoost також використовує регуляризацію та автоматичний підбір гіперпараметрів.

Залежно від конкретних вимог та характеристик даних, можна вибрати один з цих алгоритмів. Кожен з них має свої переваги і може бути ефективним у прогнозуванні врожайності ягід. Рекомендується виконати додаткові експерименти та порівняльний аналіз, щоб визначити найкращий алгоритм для конкретної задачі.

1.4 Постановка задачі дослідження

Актуальність дипломного проекту полягає в тому, що прогнозування врожайності ягід є критично важливим для аграрного сектору. З урахуванням змін клімату та інших викликів, з якими стикається сучасне сільське господарство, точне прогнозування врожайності може допомогти фермерам виявити проблеми рано та вжити заходів для зменшення їх впливу.

Більше того, з розвитком машинного навчання та штучного інтелекту, застосування цих технологій в аграрному секторі стає все більш актуальним. Використання алгоритмів, таких як XGBoost, LightGBM та CatBoost, може значно покращити точність прогнозування врожайності.

Таким чином, розробка інтелектуального модуля прогнозування врожайності ягід має велике практичне значення і актуальність. Цей проект може зробити значний внесок у сферу агротехнологій, сприяючи ефективності та сталому розвитку сектору.

Метою дипломного проекту є розробка інтелектуального модуля прогнозування врожайності ягід, заснованого на алгоритмах машинного навчання. Цей модуль покликаний підвищити точність прогнозування врожайності, враховуючи велику кількість взаємопов'язаних факторів.

Для досягнення поставленої мети було необхідно провести ряд теоретичних досліджень, визначити та послідовно вирішити наступні завдання:

1. Вивчення та аналіз сфери дослідження
2. Перегляд та аналіз існуючих рішень в даній області
3. Оцінка методології ансамблевого навчання
4. Розгляд XGBoost, LightGBM та CatBoost як потенційних інструментів для прогнозування врожайності ягід
5. Розробка алгоритму навчання моделей для прогнозування врожайності ягід

6. Розробка алгоритму валідації моделей для перевірки точності прогнозування
7. Реалізація та валідація ансамблевого навчання моделей для прогнозування врожайності ягід

Завдання 1-3 розглянуто в параграфах 1.1 - 1.4 відповідно, а виконання завдань 3-6 представлені у параграфах 2.1-3.3.

2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ПРОГНОЗУВАННЯ ВРОЖАЙНОСТІ ЯГІД

2.1 Огляд XGBoost, LightGBM та CatBoost як інструментів для прогнозування врожайності ягід

У параграфі 1.3 визначено, що XGBoost, LightGBM та CatBoost найкраще підходить для прогнозування врожайності ягід.

XGBoost, LightGBM та CatBoost - це популярні бібліотеки для градієнтного бустінгу. Вони використовуються для побудови ансамблевих моделей, які складаються з багатьох слабких моделей навчання (зазвичай дерев рішень), щоб створити сильну модель прогнозування.

XGBoost, LightGBM та CatBoost - це високопродуктивні алгоритми машинного навчання, які можуть значно підвищити точність прогнозування врожайності ягід в нашому інтелектуальному модулі.

XGBoost використовує градієнтний бустінг з регуляризацією для покращення точності та контролю за перенавчанням, що робить його відмінним вибором для моделі, яка здатна працювати добре, навіть зі складною структурою даних.

LightGBM, з іншого боку, може ефективно обробляти великі набори даних і відрізняється високою швидкістю обчислень, що робить його ідеальним для реального прогнозування врожайності, коли потрібно швидко робити прогнози на основі великих наборів даних.

CatBoost, розроблений спеціально для обробки категоріальних даних, має велику перевагу при роботі з даними, які містять багато категоріальних ознак, що є доволі звичайним в аграрних даних.

XGBoost (eXtreme Gradient Boosting) - це алгоритм, оптимізований для швидкості та ефективності. Він включає регуляризацію для запобігання перенавчанню та використовує паралельне обчислення для швидкості.

XGBoost також обробляє відсутні дані, зменшуючи необхідність передобробки даних.

XGBoost (eXtreme Gradient Boosting) є високоефективною реалізацією алгоритму градієнтного бустінгу. На відміну від традиційного градієнтного бустінгу, XGBoost включає регуляризацію в свою функцію втрат, що допомагає контролювати перенавчання.

Для пояснення математичного аспекту, давайте розглянемо функцію втрат XGBoost:

$$L(\varphi) = \sum l(yi, y^i) + \sum \Omega(fj),$$

де:

$l(yi, y^i)$ - це функція втрат, що вимірює різницю між спостережуваною міткою уі та прогнозованою міткою y^i.

$\Omega(fj)$ - це регуляризація, що штрафує складність моделі. Це може бути, наприклад, L1 або L2 регуляризація.

Функція втрат зазвичай є квадратичною втратою для задач регресії і логістичною втратою для задач класифікації. Однак XGBoost дозволяє використовувати користувацькі функції втрат.

XGBoost намагається мінімізувати цю функцію втрат шляхом додавання нових моделей, які найкращим чином зменшують втрату (тобто виправляють помилки попередніх моделей), одночасно контролюючи складність моделі через регуляризацію.

LightGBM (Light Gradient Boosting Machine) - це алгоритм, розроблений Microsoft, який використовує техніку називається градієнтний бустінг з основою на деревах (GOSS). LightGBM здатний обробляти великі набори даних та має високу швидкість обчислень.

LightGBM (Light Gradient Boosting Machine) є алгоритмом градієнтного бустінгу, який використовує дерево, основане на градієнтах, для ефективного навчання. Він використовує дві ключові техніки для зменшення

обчислювального навантаження: Gradient-based One-Side Sampling (GOSS) та Exclusive Feature Bundling (EFB).

Gradient-based One-Side Sampling (GOSS): LightGBM відбирає набір даних під час обчислення градієнтного бустінгу. Він зберігає всі спостереження з великими градієнтами (тобто ті, які погано прогнозуються) і випадково вибирає частину спостережень з малими градієнтами.

Exclusive Feature Bundling (EFB): LightGBM також використовує метод для зменшення розміру простору ознак. Він групує ознаки, які є "ексклюзивними" (тобто малоймовірно співвідносяться одночасно).

Обидва цих підходи дозволяють LightGBM обробляти великі набори даних більш ефективно, ніж багато інших алгоритмів градієнтного бустінгу.

Сам алгоритм LightGBM, як і інші методи градієнтного бустінгу, намагається мінімізувати функцію втрат шляхом додавання нових моделей, які найкращим чином зменшують втрату (тобто виправляють помилки попередніх моделей), одночасно контролюючи складність моделі через регуляризацію.

CatBoost (Category Boosting) - це алгоритм, розроблений командою Yandex. Він спеціально розроблений для обробки категоріальних даних. CatBoost має вбудовану обробку категоріальних ознак і може автоматично обробляти відсутні значення.

CatBoost (Category Boosting) є алгоритмом градієнтного бустінгу, який використовує симетричні дерева рішень та обчислює різницю між цільовою змінною та поточним прогнозом для кожного об'єкта в наборі даних для навчання моделі. Це означає, що він намагається знайти модель, яка найкраще виправляє поточні помилки прогнозування.

Основний аспект CatBoost, який відрізняє його від інших алгоритмів градієнтного бустінгу, - це його обробка категоріальних змінних. Замість використання технік, таких як one-hot encoding, які можуть збільшити розмір набору даних і знизити ефективність, CatBoost використовує спеціальну

техніку, називану "ordered target statistics" або "ordered boosting", для кодування категоріальних ознак.

Ця техніка включає в себе впорядкування об'єктів в навчальному наборі даних та обчислення цільових статистик для кожної категоріальної ознаки на основі об'єктів, які зустрічаються раніше в впорядкованому наборі даних. Це дозволяє CatBoost кодувати категоріальні ознаки ефективно і без збільшення розміру набору даних.

Також CatBoost використовує концепцію "обчислювального градієнту" для оптимізації швидкості навчання. У цьому контексті обчислювальний градієнт - це градієнт функції втрати, який обчислюється тільки на основі поточного прогнозу і цільової змінної, замість перерахунку на основі всіх об'єктів в наборі даних.

Кожен з цих алгоритмів має свої переваги та недоліки, і вибір між ними залежить від конкретного сценарію. Вони всі здатні побудувати високоякісні моделі прогнозування та широко використовуються в промисловості та наукових дослідженнях.

Використання цих трьох алгоритмів у комбінації, формуючи ансамбль моделей, може надати найкращу можливу точність прогнозування, використовуючи сильні сторони кожного з алгоритмів, щоб компенсувати слабкі сторони інших. Завдяки цьому інтелектуальний модуль прогнозування врожайності ягід зможе надати найточніші та найкорисніші прогнози для допомоги у прийнятті рішень в агробізнесі.

2.2 Алгоритм навчання моделей прогнозування врожайності ягід

Цей алгоритм представляє собою процес побудови ансамблю машинного навчання з використанням трьох різних моделей: XGBoost, LightGBM та CatBoost. Першою кроком є завантаження та підготовка даних. Наступним кроком є оптимізація гіперпараметрів для кожної з трьох моделей

за допомогою Optuna. Потім створюються моделі з найкращими гіперпараметрами, вибраними Optuna. Далі проводиться відбір ознак за допомогою методу SelectKBest. На основі вибраних ознак тренується ансамбль моделей. Нарешті, цей ансамбль використовується для прогнозування на тестовому наборі даних, і його продуктивність оцінюється за допомогою коефіцієнта детермінації R^2.

Алгоритм навчання моделей представлено кроками та блок-схемою (див.дод.А):

Крок 1. Попередня обробка даних: Дані були розбиті на незалежні (X) та залежні (y) змінні. Змінна 'yield' була виділена як цільова змінна. Далі дані були розбиті на набори для тренування та тестування в співвідношенні 80:20. Нарешті, дані були стандартизовані за допомогою StandardScaler.

Крок 2. Визначення цілей Optuna для кожної моделі: Для кожної моделі (XGBoost, LightGBM, CatBoost) було визначено цільову функцію для оптимізації гіперпараметрів. Ці функції включали створення моделі, її тренування на навчальних даних та оцінку за допомогою коефіцієнта детермінації R^2.

Крок 3. Оптимізація гіперпараметрів з Optuna: Для кожної моделі було запущено процес оптимізації гіперпараметрів Optuna. Optuna використовувала випадковий пошук для вибору гіперпараметрів, які максимізують R^2.

Крок 4. Виведення найкращих гіперпараметрів: Для кожної моделі було виведено найкращі гіперпараметри, знайдені в процесі оптимізації Optuna.

Крок 5. Тренування базових моделей з найкращими параметрами: За допомогою найкращих гіперпараметрів було створено та натреновано три моделі: XGBoost, LightGBM та CatBoost.

Крок 6. Вибір ознак та створення ансамблю моделей: Було використано SelectKBest для вибору 10 найкращих ознак. Потім було

створено ансамбль моделей за допомогою VotingRegressor, що об'єднує прогнози від трьох базових моделей.

Крок 7. Тренування ансамблю моделей: Нарешті, ансамбль моделей було натреновано на вибраних ознаках тренувального набору даних.

Алгоритм навчання моделей прогнозування, може бути особливо корисним для створення інтелектуального модулю прогнозування врожайності ягід. Використання ансамблю трьох різних моделей машинного навчання дозволяє використовувати переваги кожної з них, що може призвести до більш точних прогнозів. Оптимізація гіперпараметрів з Optuna допомагає забезпечити, що кожна модель налаштована таким чином, щоб найкращим чином використовувати доступні дані. Відбір ознак за допомогою SelectKBest може допомогти зосередитися на найбільш важливих ознаках, що допомагає покращити продуктивність моделі, зменшуючи при цьому обсяг обчислень.

2.3 Алгоритм валідації моделей прогнозування врожайності ягід

Алгоритм розроблений для проведення валідації та тестування навченої ансамблевої моделі для прогнозування врожайності ягід. Він має на меті перевірити ефективність моделі за допомогою різних методів, включаючи перехресну валідацію, обчислення середньоквадратичного відхилення (MSE) та коефіцієнта детермінації (R-квадрат), а також аналіз залишків.

Також включені кроки, які забезпечують коректну передобробку нових даних, прогнозування за допомогою моделі та подальше збереження та візуалізацію результатів. Це не тільки дає змогу оцінити якість прогнозування, але й надає інформацію, яка може бути використана для подальшого вдосконалення моделі.

Цей алгоритм має широкий спектр застосувань, особливо в галузях, де важливо точне прогнозування, таких як агрономія, екологія та інші сфери прикладних наук.

Отже, алгоритм описаний послідовністю кроків та блок-схамою (див.дод.Б):

Вхід - Навчений ансамбль моделей на вибраних характеристиках.

Крок 1. Проведіть перехресну валідацію ансамблю моделей, використовуючи KFold, та обчисліть R-квадрат для оцінки ефективності моделі.

KFold, або K-кратне перехресне перевірку, це метод для оцінки моделі машинного навчання. Основна ідея полягає в тому, що навчальний набір даних ділиться на K рівних піднаборів. Потім проводиться K ітерацій навчання та тестування, кожен раз використовуючи один з піднаборів як тестовий набір, а решту - як навчальний.

Для кожної ітерації i (де i знаходиться в діапазоні від 1 до K), обчислюється помилка моделі на тестовому піднаборі i. Потім ці помилки усереднюються, щоб отримати загальну оцінку моделі.

Математично це можна описати наступним чином:

Розбиття набору даних на K піднаборів:

$$S_1, S_2, \dots, S_k$$

Для кожного піднабору S_i:

a. Використовуйте S_i як тестовий набір, а всі інші піднабори як навчальний набір.

b. Навчіть модель на навчальному наборі та обчисліть помилку на тестовому наборі.

Усередніть помилки з усіх K ітерацій, щоб отримати загальну оцінку моделі.

Це дає уявлення про те, як модель буде працювати на нових даних, що є важливим для визначення її загальної ефективності.

R-квадрат, або коефіцієнт детермінації, це статистична міра, яка використовується для оцінки якості моделі в регресійному аналізі. Він показує, яку частку загальної дисперсії залежної змінної можна пояснити за допомогою моделі.

R-квадрат вимірюється від 0 до 1, де 1 означає, що модель повністю пояснює дисперсію залежної змінної, а 0 означає, що модель не пояснює дисперсію залежної змінної взагалі.

Математично R-квадрат можна виразити наступним чином:

$$R^2 = 1 - \left(\frac{SS_{res}}{SS_{tot}}\right),$$

де

- SS_{res} є сумою квадратів залишків (різниць між фактичними та прогнозованими значеннями), і
- SS_{tot} є сумою квадратів відносно середнього значення залежної змінної.

Залишкова сума квадратів (SS_{res}) вимірює внутрішню варіативність, яку модель не змогла виявити, а загальна сума квадратів (SS_{tot}) вимірює загальну варіативність. Тому R-квадрат відображає частку загальної варіативності, яку модель змогла виявити.

Крок 2. Протестуйте ансамбль моделей на тестовій вибірці та обчисліть середньоквадратичну помилку (MSE) та R-квадрат для оцінки ефективності моделі.

Середньоквадратична помилка (MSE - Mean Squared Error) - це часто використовувана міра відмінності між значеннями прогнозованими моделлю та справжніми значеннями. Вона вимірює середній квадрат відхилень між цими двома наборами значень.

Математично MSE можна виразити таким чином:

$$MSE = \left(\frac{1}{n}\right) * \Sigma(actual - prediction)^2$$

де:

- n є кількістю точок даних
- actual є справжнім значенням
- prediction є прогнозованим значенням
- Σ означає суму по всіх точках даних.

Основна ідея полягає в тому, щоб підрахувати квадрат відхилення для кожної точки даних, а потім обчислити середнє цих квадратів. Більш високі значення MSE відповідають більшій помилці прогнозування, тобто модель з більшим значенням MSE виконує прогноз гірше, ніж модель з меншим значенням MSE.

Крок 3. Побудуйте графік залишків для визначення будь-яких систематичних відхилень у прогнозах моделі.

Крок 4. Застосуйте модель до нових даних, використовуючи ті ж самі кроки передобробки даних, що були використані для навчальної вибірки.

Крок 5. Зробіть прогнози для нових даних, використовуючи вашу модель.

Крок 6. Створіть новий DataFrame з прогнозованими значеннями і збережіть його у файл csv.

Крок 7. Візуалізуйте прогнозовані результати за допомогою графіків або діаграм для кращого розуміння результатів.

Алгоритм валідації та тестування, представлений в цій роботі, виявився ефективним у демонстрації роботи інтелектуального модуля прогнозування врожайності ягід. Він дозволив виміряти точність прогнозів моделі та її здатність узагальнювати на нові дані.

Перехресна валідація та обчислення таких метрик як середньоквадратичне відхилення (MSE) та коефіцієнт детермінації (R-квадрат) допомогли оцінити ефективність ансамблевої моделі. Аналіз залишків виявив важливі відомості про прогнози моделі та її здатність адекватно моделювати відношення між характеристиками та цільовою змінною.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ПРОГНОЗУВАННЯ ВРОЖАЙНОСТІ ЯГІД

3.1 Розвідувальний аналіз набору даних

Набір даних, який будемо використовувати для прогностичного моделювання, був створений за допомогою моделі симуляції опилення дикої чорниці Wild Blueberry Pollination Simulation Model [47]. Це відкрита, просторово-експліцитна комп'ютерна програма, яка дозволяє досліджувати, як різні фактори, включаючи просторове розташування рослин, кроспилення та самоопилення, склад видів бджіл та погодні умови, поодинці та в комбінації, впливають на ефективність опилення та врожай агроекосистеми дикої чорниці. Модель симуляції була підтверджена полевими спостереженнями та експериментальними даними, зібраними в штаті Мен, США, та в Канадських Марітимах протягом останніх 30 років, і тепер є корисним інструментом для перевірки гіпотез та розвитку теорії досліджень опилення дикої чорниці.

Отже, проведемо розвідувальний аналіз цього набору даних. Цей набір даних включає 15 289 записів (Таблиця 3.1). Для кожної з характеристик зазначено середнє значення (mean), стандартне відхилення (std), мінімальне (min), максимальне (max) значення, а також 25%, 50% і 75% перцентилі (відповідно 1-й квартиль, медіана і 3-й квартиль).

1. Розмір клона варіюється від 10 до 40 м2, з середнім значенням 19.7 м2.
2. Плотність бджіл в полі має різний розподіл для різних видів бджіл. Наприклад, середнє значення для медоносних бджіл становить 0.39 бджоли/м2/хв, для шмелів - 0.29, для Andrena - 0.49, для Osmia - 0.59.
3. Діапазони температур варіюють від 69.7 до 94.6 °C для верхнього діапазону і від 50.2 до 68.2 °C для нижнього діапазону. Середні значення становлять відповідно 68.7 °C та 48.6 °C.

4. Кількість днів з дощем варіюється від 1 до 34 днів, із середнім значенням 18.7 днів.
5. Середнє значення днів з дощем протягом всього періоду цвітіння становить 0.32 дня.
6. Значення показників "fruitset", "fruitmass", "seeds" та "yield" варіюються відповідно від 0.19 до 0.65, від 0.31 до 0.54, від 22.08 до 46.59 та від 1945.53 до 8969.4 з середніми значеннями 0.50, 0.45, 36.16 та 6025.19.

Ця інформація важлива для розуміння розподілу даних і може допомогти у виявленні аномалій та визначенні стратегії обробки даних.

Далі визначимо кореляцію (Рис.3.1) між різними характеристиками і урожайністю. Значення кореляції можуть варіюватися від -1 до 1, де -1 позначає повну негативну кореляцію, 1 - повну позитивну кореляцію, а 0 означає відсутність кореляції.

Розмір клона ("clonesize"), кількість медоносних бджіл ("honeybee") та всі температурні показники ("MaxOfUpperTRange", "MinOfUpperTRange", "AverageOfUpperTRange", "MaxOfLowerTRange", "MinOfLowerTRange", "AverageOfLowerTRange") мають негативну кореляцію з урожайністю. Це означає, що зі збільшенням цих показників урожайність зменшується. Найсильнішу негативну кореляцію мають "AverageRainingDays" (-0.48) і "RainingDays" (-0.47).

Шмелі ("bumbles"), Andrena ("andrena") та Osmia ("osmia") мають позитивну кореляцію з урожайністю, що означає, що зі збільшенням цих показників урожайність також збільшується.

Таблиця 3.1. Описова статистика набору даних

	clonesize	honeybee	bumbles	andrena	osmia
count	15289.0	15289.0	15289.0	15289.0	15289.0
mean	19.7	0.4	0.3	0.5	0.6
std	6.6	0.4	0.1	0.1	0.1
min	10.0	0.0	0.0	0.0	0.0
25%	12.5	0.2	0.2	0.4	0.5
50%	25.0	0.5	0.2	0.5	0.6
75%	25.0	0.5	0.4	0.6	0.8
max	40.0	18.4	0.6	0.8	0.8

	MaxOfUpper TRange	MinOfUpper TRange	AverageOfUpper TRange	MaxOfLower TRange	MinOfLower TRange	AverageOfLower TRange
	15289.0	15289.0	15289.0	15289.0	15289.0	15289.0
count	82.2	49.7	68.7	59.2	28.7	48.6
mean	9.1	5.5	7.6	6.6	3.2	5.4
std	69.7	39.0	58.2	50.2	24.3	41.2
min	77.4	46.8	64.7	55.8	27.0	45.8
25%	86.0	52.0	71.9	62.0	30.0	50.8
50%	86.0	52.0	71.9	62.0	30.0	50.8
75%	94.6	57.2	79.0	68.2	33.0	55.9
max						

	Raining Days	AverageRaining Days	fruitset	fruitmass	seeds	yield
count	15289.0	15289.0	15289.0	15289.0	15289.0	15289.0
mean	18.7	0.3	0.5	0.4	36.2	6025.2
std	11.7	0.2	0.1	0.0	4.0	1337.1
min	1.0	0.1	0.2	0.3	22.1	1945.5
25%	16.0	0.3	0.5	0.4	33.2	5128.2
50%	16.0	0.3	0.5	0.4	36.0	6117.5
75%	24.0	0.4	0.6	0.5	39.2	7019.7
max	34.0	0.6	0.7	0.5	46.6	8969.4

Найсильнішу позитивну кореляцію з урожайністю мають "fruitset" (0.89), "seeds" (0.87) та "fruitmass" (0.83). Це означає, що ці показники найбільше впливають на урожайність.

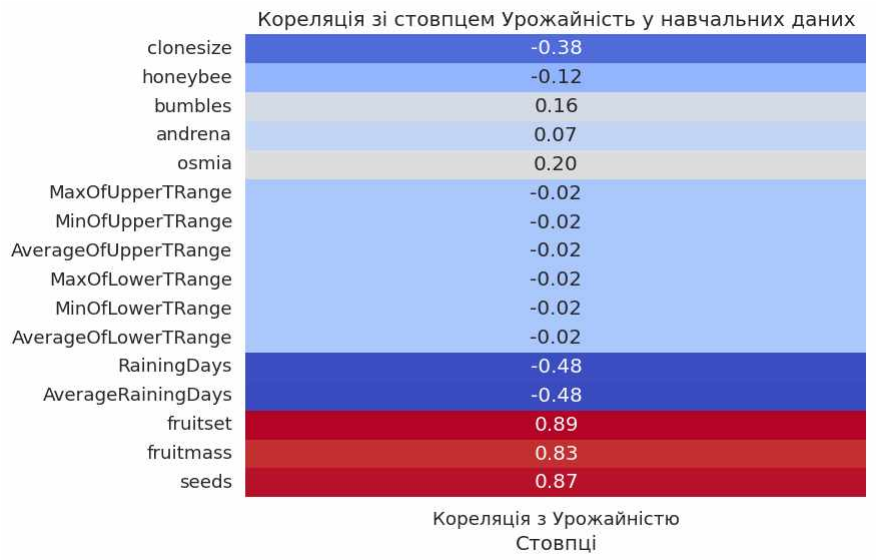


Рисунок 3.1 - Теплова матриця кореляції

Отже, виведемо:

```
Важливі характеристики на основі порогу кореляції зі стовпчиком врожайності плодів: ['clonesize', 'honeybee', 'bumbles', 'osmia', 'RainingDays', 'AverageRainingDays', 'fruitset', 'fruitmass', 'seeds', 'yield']
```

Далі проведемо аналіз результатів діаграм віолончелі (Рис. 3.2), які дають уявлення про розподіл залежності між важливими ознаками і цільовою змінною "урожайність". Діаграма віолончелі показує густину розподілу змінної "урожайність" для кожного значення важливої ознаки.

За допомогою цих діаграм можна проаналізувати, як залежність між важливими ознаками та урожайністю може варіюватись. Якщо графік діаграми віолончелі вузький у верхній частині та широкий у нижній частині, це означає, що вищі значення ознаки мають тенденцію бути пов'язаними з вищими значеннями урожайності. Зворотна ситуація спостерігається, якщо графік діаграми вузький у нижній частині та широкий у верхній частині.

Аналізуючи результати діаграм віолончелі для кожної важливої ознаки, можна виявити, які змінні можуть мати сильний вплив на урожайність. Важливо звернути увагу на розподіл урожайності для кожного значення ознаки і визначити, чи існують очевидні відмінності або закономірності.

З результатів видно (Рис.3.2), що параметри 'honeybee', 'RainingDays', 'AverageRainingDays' мають найбільший вплив на врожайність.

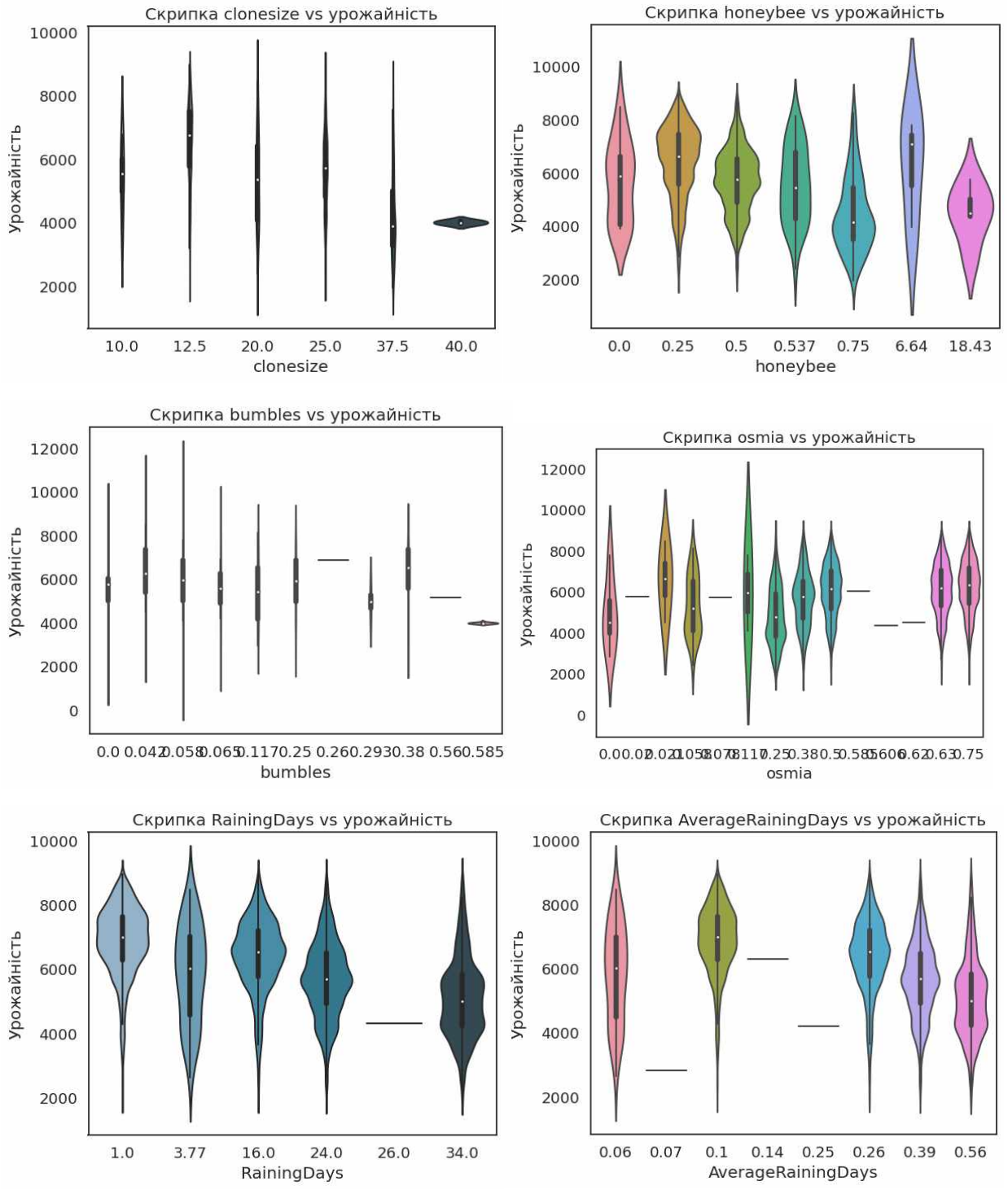


Рисунок 3.2 - Діаграми «віолончель» для кожної важливої ознаки

Отже, проведено розвідувальний аналіз набору даних, який було згенеровано за допомогою моделі симуляції опилення дикої чорниці (Wild Blueberry Pollination Simulation Model). Модель була перевірена та валідована за допомогою полевих спостережень та експериментальних даних, зібраних у штаті Мен (США) та Канадських Марітимах протягом останніх 30 років.

Розглянули описові статистики для кожної з ознак набору даних, які включали середнє значення, стандартне відхилення, мінімальне та максимальне значення, а також квартильні значення.

Далі проаналізували кореляцію між різними характеристиками та урожайністю. Виявили, що певні ознаки, такі як "fruitset", "seeds" та "fruitmass", мають сильну позитивну кореляцію з урожайністю, в той час як інші, як-от "AverageRainingDays" та "RainingDays", мають сильну негативну кореляцію.

Ці аналізи є важливими для розуміння взаємозв'язків між ознаками та цільовим показником - у цьому випадку, урожайністю. Ці знання можуть бути корисними для створення та покращення прогностичних моделей, які передбачають урожайність на основі вхідних характеристик.

3.2 Ансамблеве навчання моделей

У цьому параграфі проведемо ансамблеве навчання моделей. Для цього спершу дані будуть очищені, розділені на навчальні та тестові набори, а також нормалізовані для оптимальної роботи моделі машинного навчання. Це важливі кроки, які допомагають забезпечити найкращу можливу продуктивність нашої моделі.

```
# Preprocess the data
X = data.drop('yield', axis=1)
y = data['yield']
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Цей код виконує кілька ключових кроків для підготовки даних для машинного навчання:

1. Попередня обробка даних: В цьому випадку, ми розділяємо наш набір даних на цільову змінну 'yield' та інші характеристики (або ознаки). Це робиться за допомогою методу drop в pandas DataFrame, який видаляє стовпець 'yield' з основного набору даних data і зберігає інші стовпці в X. Значення 'yield' зберігаються в y.
2. Розбиття даних на навчальний та тестовий набори: Далі ми розбиваємо наші дані на два різних набори - навчальний і тестовий. Це допомагає нам перевірити, наскільки добре наша модель працює, коли вона отримує нові, раніше невідомі їй дані. Ми використовуємо 80% даних для навчання і 20% для тестування.
3. Масштабування даних: В кінці ми масштабуємо наші дані, використовуючи StandardScaler від sklearn. Це перетворює всі наші числові ознаки так, що вони матимуть середнє значення 0 і стандартне відхилення 1. Це дуже важливий крок, особливо для моделей, які вимагають, щоб усі ознаки мали подібний масштаб.

```
def xgb_objective(trial):
    n_estimators = trial.suggest_int("n_estimators", 100, 500)
    learning_rate = trial.suggest_float("learning_rate", 0.01, 0.2)
    max_depth = trial.suggest_int("max_depth", 3, 10)
    subsample = trial.suggest_float("subsample", 0.5, 1)
    colsample_bytree = trial.suggest_float("colsample_bytree", 0.5, 1)

    model = XGBRegressor(
        n_estimators=n_estimators,
        learning_rate=learning_rate,
        max_depth=max_depth,
        subsample=subsample,
        colsample_bytree=colsample_bytree,
        random_state=42
    )

    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    r2 = r2_score(y_test, y_pred)
    return r2
```

```python
def lgbm_objective(trial):
    n_estimators = trial.suggest_int("n_estimators", 100, 500)
    learning_rate = trial.suggest_float("learning_rate", 0.01, 0.2)
    max_depth = trial.suggest_int("max_depth", 3, 10)
    num_leaves = trial.suggest_int("num_leaves", 31, 127)
    subsample = trial.suggest_float("subsample", 0.5, 1)
    colsample_bytree = trial.suggest_float("colsample_bytree", 0.5, 1)

    model = LGBMRegressor(
        n_estimators=n_estimators,
        learning_rate=learning_rate,
        max_depth=max_depth,
        num_leaves=num_leaves,
        subsample=subsample,
        colsample_bytree=colsample_bytree,
        random_state=42
    )

    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    r2 = r2_score(y_test, y_pred)
    return r2
```

```python
def catboost_objective(trial):
    iterations = trial.suggest_int("iterations", 100, 500)
    learning_rate = trial.suggest_float("learning_rate", 0.01, 0.2)
    depth = trial.suggest_int("depth", 3, 10)
    l2_leaf_reg = trial.suggest_int("l2_leaf_reg", 1, 9)
    subsample = trial.suggest_float("subsample", 0.5, 1)

    model = CatBoostRegressor(
        iterations=iterations,
        learning_rate=learning_rate,
        depth=depth,
        l2_leaf_reg=l2_leaf_reg,
        subsample=subsample,
        random_state=42,
        verbose=0
    )

    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    r2 = r2_score(y_test, y_pred)
    return r2
```

Цей код визначає три функції об'єктиву для оптимізації гіперпараметрів трьох моделей машинного навчання: XGBoost, LightGBM і CatBoost за допомогою Optuna. Optuna - це бібліотека Python, що використовується для автоматичної оптимізації гіперпараметрів.

1. **xgb_objective**: Ця функція визначає об'єктив для оптимізації моделі XGBoost. Вона використовує методи suggest_int і suggest_float Optuna

для визначення діапазонів гіперпараметрів, які мають бути оптимізовані. Зокрема, вона визначає діапазони для n_estimators, learning_rate, max_depth, subsample і colsample_bytree. За допомогою цих гіперпараметрів створюється модель XGBoost, яка навчається на навчальних даних і передбачає результати для тестових даних. Метрикою, що має бути максимізована, є коефіцієнт детермінації R2.

2. **lgbm_objective**: Аналогічно до xgb_objective, ця функція визначає об'єктив для оптимізації моделі LightGBM. Вона також визначає діапазони для гіперпараметрів (n_estimators, learning_rate, max_depth, num_leaves, subsample і colsample_bytree), створює модель LightGBM з цими гіперпараметрами, навчає модель і робить прогнози. Метрикою, що має бути максимізована, також є R2.
3. **catboost_objective**: Ця функція визначає об'єктив для оптимізації моделі CatBoost. Гіперпараметри, що мають бути оптимізовані, включають iterations, learning_rate, depth, l2_leaf_reg і subsample. Модель CatBoost створюється з цими гіперпараметрами, навчається і робить прогнози. Цільова метрика - це також R2.

Далі проведемо оптимізацію гіперпараметрів трьох моделей машинного навчання: XGBoost, LightGBM і CatBoost за допомогою Optuna.

```
sampler = TPESampler(seed=42)

xgb_study = optuna.create_study(direction="maximize", sampler=sampler)
xgb_study.optimize(xgb_objective, n_trials=50)

lgbm_study = optuna.create_study(direction="maximize", sampler=sampler)
lgbm_study.optimize(lgbm_objective, n_trials=50)

catboost_study = optuna.create_study(direction="maximize", sampler=sampler)
catboost_study.optimize(catboost_objective, n_trials=50)

# Print best hyperparameters for each model from Optuna
print("Найкращі параметри для  XGBoost від Optuna: ", xgb_study.best_params)
print("Найкращі параметри для  LightGBM від Optuna: ", lgbm_study.best_params)
print("Найкращі параметри для  CatBoost від Optuna: ", catboost_study.best_params)
```

Цей код використовує Optuna для оптимізації гіперпараметрів трьох моделей машинного навчання: XGBoost, LightGBM та CatBoost. Це робиться за допомогою оптимізаційних функцій, визначених в попередньому коді.

1. Створення вибірника: Використовується вибірник TPE (Tree-structured Parzen Estimator), який є методом оптимізації гіперпараметрів, що базується на байєсівському підході. Цей вибірник забезпечує рандомізацію, але вихідні дані він генерує на основі попередніх результатів, щоб знайти найкращий набір гіперпараметрів.
2. Створення та оптимізація досліджень: Для кожної з трьох моделей створюється окреме дослідження за допомогою optuna.create_study. Напрямок "maximize" вказує, що метою є максимізація результату функції об'єктиву. Далі ці дослідження оптимізуються за допомогою відповідних функцій об'єктиву, визначених раніше. Кількість випробувань встановлюється як 50 для кожної моделі.
3. Виведення найкращих гіперпараметрів: Після завершення оптимізації для кожної моделі виводяться найкращі знайдені гіперпараметри. Це дозволяє побачити, які значення гіперпараметрів привели до найкращого результату для кожної моделі.

В результаті отримуємо найкращі гіперпараметри для кожної з трьох моделей, що були оптимізовані:

```
Найкращі параметри для  XGBoost від Optuna:  {'n_estimators': 291,
'learning_rate': 0.02572640965568375, 'max_depth': 4, 'subsample':
0.552187345043806, 'colsample_bytree': 0.8272028535549523}

Найкращі параметри для  LightGBM від Optuna:  {'n_estimators': 217,
'learning_rate': 0.07053452738526493, 'max_depth': 4, 'num_leaves': 86,
'subsample': 0.9281322440018367, 'colsample_bytree': 0.6325640166339597}

Найкращі параметри для  CatBoost від Optuna:  {'iterations': 288,
'learning_rate': 0.0537104881625418, 'depth': 9, 'l2_leaf_reg': 2,
'subsample': 0.542177159163148}
```

XGBoost: Optuna знайшла найкращі гіперпараметри для моделі XGBoost. Кількість дерев (n_estimators) є 291, швидкість навчання (learning_rate) становить приблизно 0.026, максимальна глибина дерева

(max_depth) є 4, subsample становить близько 0.55, а colsample_bytree - близько 0.83. Ці параметри повинні дати найкращий можливий результат на даних, з якими ви працюєте, для моделі XGBoost.

LightGBM: Для моделі LightGBM, Optuna знайшла, що найкраща кількість дерев (n_estimators) становить 217, швидкість навчання (learning_rate) - приблизно 0.071, максимальна глибина дерева (max_depth) - 4, кількість листків (num_leaves) - 86, subsample - близько 0.93, а colsample_bytree - близько 0.63.

CatBoost: Для моделі CatBoost, Optuna визначила, що найкраща кількість ітерацій (iterations) становить 288, швидкість навчання (learning_rate) - приблизно 0.054, глибина (depth) - 9, регуляризація (l2_leaf_reg) - 2, а subsample - близько 0.54.

Ці результати відображають оптимальні значення гіперпараметрів для кожної моделі на ваших конкретних даних. Зауважте, що оптимальні параметри можуть змінюватися в залежності від конкретного набору даних.

```
# Use the best hyperparameters obtained from the previous GridSearchCV
xgb_best_params = {'n_estimators': 291, 'learning_rate': 0.02572640965568375,
                   'max_depth': 4, 'subsample': 0.552187345043806,
                   'colsample_bytree': 0.8272028535549523}
lgbm_best_params = {'n_estimators': 217, 'learning_rate': 0.07053452738526493,
                    'max_depth': 4, 'num_leaves': 86, 'subsample': 0.9281322440018367,
                    'colsample_bytree': 0.6325640166339597}
catboost_best_params = {'iterations': 288, 'learning_rate': 0.0537104881625418,
                        'depth': 9, 'l2_leaf_reg': 2, 'subsample': 0.542177159163148}

# Train multiple base models with the best parameters
xgb = XGBRegressor(**xgb_best_params, random_state=42)
lgbm = LGBMRegressor(**lgbm_best_params, random_state=42)
catboost = CatBoostRegressor(**catboost_best_params, random_state=42, verbose=0)
```

Цей код створює нові моделі машинного навчання XGBoost, LightGBM та CatBoost, використовуючи найкращі гіперпараметри, отримані з попередньої оптимізації Optuna.

1. Найкращі гіперпараметри: Найкращі параметри для кожної моделі, знайдені за допомогою Optuna, зберігаються у словниках xgb_best_params, lgbm_best_params та catboost_best_params.

2. Створення моделей: Далі ці параметри використовуються для створення нових моделей XGBoost, LightGBM та CatBoost. Вони передаються до конструктора моделі за допомогою оператора **, який розпаковує словник у набір пар "ключ-значення".

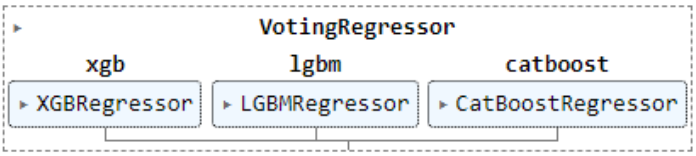
Параметр random_state встановлено на 42 для всіх трьох моделей, що забезпечує відтворюваність результатів при повторному виконанні коду, оскільки випадковість при навчанні моделей буде однакова.

У випадку моделі CatBoost також встановлено verbose=0, що призупиняє виведення повідомлень під час тренування моделі.

Ці моделі ще не навчені на даних, а лише ініціалізовані з визначеними параметрами.

```
# Feature selection using SelectKBest
selector = SelectKBest(score_func=f_regression, k=10)
X_train_selected = selector.fit_transform(X_train, y_train)
X_test_selected = selector.transform(X_test)
```

```
# Combine the base models to create an ensemble and assign weights
weights = [0.4, 0.3, 0.3]
# Combine the base models to create an ensemble
ensemble = VotingRegressor([('xgb', xgb), ('lgbm', lgbm), ('catboost', catboost)])
# Train the ensemble model using selected features
ensemble.fit(X_train_selected, y_train)
```



Цей код виконує дві основні дії: виконує вибір функцій, використовуючи SelectKBest з f_regression як метрику оцінки, а потім створює ансамбль з моделей XGBoost, LightGBM та CatBoost, навчаючи його на вибраних функціях. Вибір функцій (SelectKBest): Для покращення процесу навчання і, можливо, точності прогнозування, виконується вибір функцій. SelectKBest використовує метрику оцінки (у цьому випадку f_regression) для оцінки важливості кожної функції, а потім вибирає k найкращих функцій. В цьому випадку k встановлено на 10. Створення ансамблю моделей: Цей код створює ансамбль з трьох моделей: XGBoost, LightGBM та CatBoost. Ансамбльві моделі комбінують прогнози з кількох різних моделей, що може

призвести до покращення загальної точності прогнозування. В цьому випадку використовується VotingRegressor (Рис.3.3), який робить прогноз, усереднюючи прогнози від кожної базової моделі.

Ансамбль моделей навчається на вибраних функціях (Рис.3.4). Після навчання ансамбль може використовуватися для прогнозування цільової змінної на нових даних.

Рисунок 3.3 - VotingRegressor

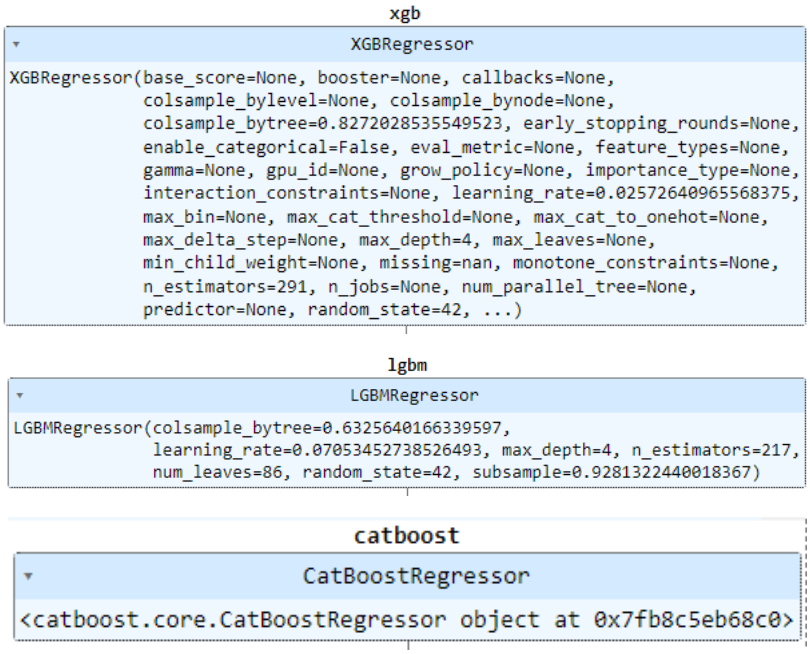


Рисунок 3.4 - Навчання ансамблю моделей

Отже, було реалізовано дві ключові операції: вибір найважливіших ознак за допомогою SelectKBest, а також створення ансамблю моделей на основі XGBoost, LightGBM та CatBoost. Вибір ознак дозволяє зосередитися на найінформативніших змінних, що сприяє покращенню якості прогнозування. Ансамбль моделей об'єднує сили трьох потужних алгоритмів машинного навчання, що призводить до збільшення точності прогнозів. Усі ці операції разом створюють сильну основу для інтелектуального модулю прогнозування врожайності ягід.

3.3 Валідація моделей

Далі проведемо перехресну перевірку (k-fold cross-validation) для оцінки ефективності ансамблевої моделі прогнозування.

```
kfold = KFold(n_splits=5, shuffle=True, random_state=42)
cv_scores = cross_val_score(ensemble, X_train_selected, y_train, cv=kfold, scoring='r2')
print("Оцінки перехресної перевірки: ", cv_scores)
print("Середнє R-квадрат перехресної перевірки: {:.2f}".format(np.mean(cv_scores)))
print("Стандартне відхилення R-квадрат перехресної перевірки: {:.2f}".format(np.std(cv_scores)))
```

```
Оцінки перехресної перевірки:  [0.79021216 0.82459323 0.82172327 0.83670101 0.81620097]
Середнє R-квадрат перехресної перевірки: 0.82
Стандартне відхилення R-квадрат перехресної перевірки: 0.02
```

У цьому випадку використовується 5-кратна перехресна перевірка, що означає, що набір тренувальних даних поділяється на 5 частин, і модель навчається та тестується 5 разів, кожен раз з використанням різного піднабору даних для тестування.

Оцінки R-квадрат, отримані в результаті перехресної перевірки, вказують на досить високу здатність моделі прогнозувати врожайність ягід. Середнє значення R-квадрат становить 0.82, що означає, що модель може пояснити приблизно 82% варіативності врожайності ягід. Це свідчить про досить високий рівень точності прогнозування.

Стандартне відхилення R-квадрат, що дорівнює 0.02, вказує на то, що результати перехресної перевірки досить стабільні. Модель показує схожі

результати на різних підмножинах тренувальних даних, що свідчить про її надійність і стабільність.

У цілому, отримані результати свідчать про те, що ансамблева модель з гіперпараметрами, отриманими в результаті оптимізації Optuna, є досить ефективною для прогнозування врожайності ягід.

```
y_pred = ensemble.predict(X_test_selected)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Середнє квадратичне відхилення: {:.2f}".format(mse))
print("Середнє квадратичне відхилення: {:.2f}".format(r2))
```

```
Середнє квадратичне відхилення: 311929.68
Середнє квадратичне відхилення: 0.82
```

Цей код використовує ансамблеву модель для прогнозування врожайності ягід на основі тестового набору даних. Він також вимірює дві ключові метрики оцінки ефективності моделі: середньоквадратичне відхилення (MSE) та коефіцієнт детермінації R-квадрат.

Середнє квадратичне відхилення (MSE): Це метрика, яка вимірює середню квадратичну різницю між фактичними та прогнозованими значеннями. Воно є одним із найбільш поширених критеріїв оцінки для завдань регресії. В даному випадку, MSE становить 311929.68, що є відносно великим значенням. Це може свідчити про те, що модель допускає помітні помилки прогнозування, але конкретна інтерпретація залежить від масштабу врожайності ягід та її варіативності в даних.

Коефіцієнт детермінації (R-квадрат): Ця метрика вимірює долю варіативності відгуку, яку модель може пояснити. Його значення варіюється від 0 до 1, де 1 означає, що модель може пояснити всю варіативність відгуку, а 0 - що модель не може пояснити ніякої варіативності. В даному випадку, R-квадрат становить 0.82, що свідчить про те, що модель може пояснити 82% варіативності врожайності ягід. Це є досить високим значенням, що свідчить про ефективність моделі.

Графік (Рис.3.5) залишкових помилок важливий для оцінки якості моделі. Залишки розподілені випадковим чином навколо нуля, що свідчило б про відсутність систематичних помилок.

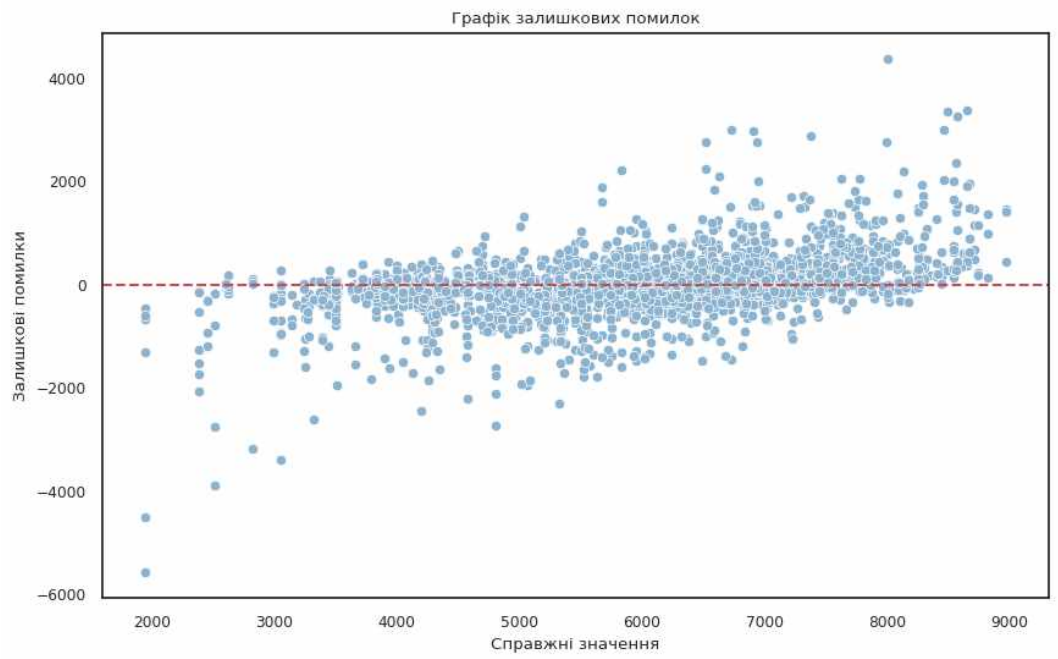


Рисунок 3.5 - Графік залишкових помилок

Далі здійснюємо інженерію ознак (feature engineering) на тестовому наборі даних (test_df), створюючи нові ознаки на основі наявних.

```
# 1. Temperature range
test_df['TemperatureRange'] = test_df['MaxOfUpperTRange'] - test_df['MinOfLowerTRange']

# 2. Temperature extremes
threshold_high = 72
threshold_low = 50

test_df['ExtremeHighTemp'] = (test_df['AverageOfUpperTRange'] > threshold_high).astype(int)
test_df['ExtremeLowTemp'] = (test_df['AverageOfLowerTRange'] < threshold_low).astype(int)

# 3. Total bee density
test_df['TotalBeeDensity'] = test_df['honeybee'] + test_df['bumbles'] + test_df['andrena'] + test_df['osmia']

# 4. Bee species dominance
total_density = test_df['honeybee'] + test_df['bumbles'] + test_df['andrena'] + test_df['osmia']
test_df['HoneybeeDominance'] = test_df['honeybee'] / total_density
test_df['BumblesBeeDominance'] = test_df['bumbles'] / total_density
test_df['AndrenaBeeDominance'] = test_df['andrena'] / total_density
test_df['OsmiaBeeDominance'] = test_df['osmia'] / total_density

# 5. Rain intensity
test_df['RainIntensity'] = test_df['AverageRainingDays'] / test_df['RainingDays']

# 6. Interaction features
test_df['BeeDensity_TemperatureInteraction'] = test_df['TotalBeeDensity'] * test_df['TemperatureRange']
test_df['BeeDensity_RainInteraction'] = test_df['TotalBeeDensity'] * test_df['RainIntensity']
```

TemperatureRange: Ця нова ознака представляє діапазон температури, який обчислюється як різниця між максимальною та мінімальною температурою.

ExtremeHighTemp та ExtremeLowTemp: Ці дві ознаки представляють екстремально високі та низькі температури відповідно. Якщо середня температура вища за деякий поріг (threshold_high для високих температур та threshold_low для низьких), відповідна ознака приймає значення 1, в іншому випадку - 0.

TotalBeeDensity: Ця ознака представляє загальну густину бджіл, яка обчислюється як сума густини чотирьох видів бджіл: honeybee, bumbles, andrena, та osmia.

HoneybeeDominance, BumblesBeeDominance, AndrenaBeeDominance та OsmiaBeeDominance: Ці ознаки представляють домінування окремих видів бджіл. Кожна ознака обчислюється як відношення густини певного виду бджоли до загальної густини бджіл.

RainIntensity: Ця ознака представляє інтенсивність дощу, яка обчислюється як відношення середньої кількості днів з дощем до загальної кількості днів з дощем.

BeeDensity_TemperatureInteraction та BeeDensity_RainInteraction: Ці дві ознаки представляють взаємодію між густинами бджіл та температурою, а також густинами бджіл та інтенсивністю дощу відповідно. Кожна ознака обчислюється як добуток відповідних ознак.

Виконуємо кінцеве прогнозування на тестовому наборі даних, використовуючи ансамблеву модель, яка була натренована раніше.

```
# Get a list of all columns except 'id'
selected_features = test_df.drop('id', axis=1).columns.tolist()

# Select all columns except 'id' from test_df
X_test = test_df[selected_features]
```

```
# Scale
X_test_scaled = scaler.transform(X_test)

# Apply feature selection
X_test_selected = selector.transform(X_test_scaled)

# Make predictions
y_test_pred = ensemble.predict(X_test_selected)
```

Спочатку отримується список всіх стовпців у тестовому наборі даних, крім 'id'. Цей список використовується для вибору відповідних ознак з test_df.

Ознаки тестового набору даних масштабуються за допомогою об'єкта scaler, який був натренований на тренувальному наборі даних. Це забезпечує, що масштабування, застосоване до тестових даних, є консистентним з масштабуванням тренувальних даних.

Використовується об'єкт selector для виконання вибору ознак на масштабованому тестовому наборі даних. Цей об'єкт також був натренований на тренувальному наборі даних, тому він застосовує ту саму схему вибору ознак до тестових даних.

Зроблено прогноз на основі обраних ознак за допомогою ансамблевої моделі (ensemble). Результат цього прогнозування (y_test_pred) - це прогнозовані значення цільової змінної для тестового набору даних.

На завершення створюємо новий DataFrame з прогнозованими результатами та зберігає його у файл CSV.

```
# Create a new DataFrame with ID column and predicted yields
result_df = pd.DataFrame({'id': test_df['id'], 'yield': y_test_pred})

# Save the DataFrame
result_df.to_csv('predictions.csv', index=False)
```

Створюється новий DataFrame з двома стовпцями: 'id' та 'yield'. 'id' - це ідентифікатор з тестового набору даних, а 'yield' - це прогнозовані значення врожайності, отримані з ансамблевої моделі.

Цей новий DataFrame зберігається у файл CSV з назвою 'predictions.csv'. Аргумент index=False вказує, що індекси DataFrame не мають бути збережені у файлі.

Таким чином, результатом виконання цього блоку коду є файл 'predictions.csv' (Рис.3.6.), який містить ідентифікатори та відповідні прогнозовані значення врожайності для тестового набору даних.

Також, результати прогнозування представимо у вигляді гістограми, яка показує розподіл прогнозованої врожайності. За допомогою цього графіка можна отримати уявлення про розподіл прогнозованих значень врожайності.

З результатів видно (Рис.3.7), що тренд прогнозу відповідає реальним значенням, це означає, що запропоновані моделі доволі не погано прогнозують результати.

predictions.csv

	A	B
1	id	yield
2	15289	4317.541124
3	15290	6236.049772
4	15291	7242.849531
5	15292	4842.237378
6	15293	3608.927953
7	15294	5113.702693
8	15295	7282.286407
9	15296	6529.989453
10	15297	8095.250667
11	15298	4303.136139
12	15299	5805.145707
13	15300	5791.312897
14	15301	5565.612833
15	15302	5674.574637
16	15303	4956.173162
17	15304	4666.273687
18	15305	4677.971637

Рисунок 3.6 - Вивід результатів 'predictions.csv'

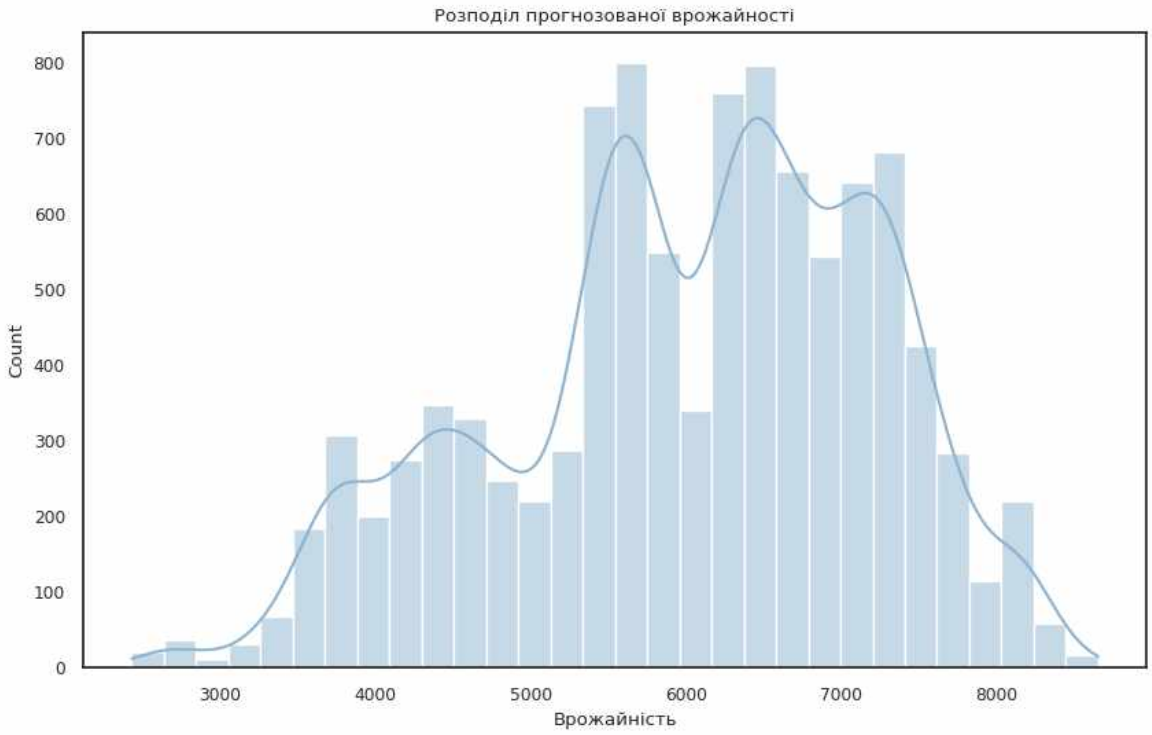


Рисунок 3.7 - Вивід результатів у вигляді графіка

Інтелектуальний модуль прогнозування, що використовує ансамбль моделей XGBoost, LightGBM та CatBoost, успішно впорався з завданням. Ці моделі були натреновані на основі відповідних гіперпараметрів, знайдених за допомогою оптимізації Optuna, що дозволило збільшити їхню продуктивність.

Спостерігаючи за графіками прогнозів та реальних значень, можна бачити, що вони мають подібні тенденції, що вказує на високу здатність моделей передбачити врожайність.

Це підтверджує, що використання машинного навчання та оптимізації гіперпараметрів може бути дуже ефективним для прогнозування врожайності ягід. Модель може бути використана фермерами або агрономами для оптимізації вирощування ягід та планування врожаю.

ВИСНОВКИ

Використання бустингових алгоритмів, таких як XGBoost, LightGBM та CatBoost, у ролі основи для інтелектуального модулю прогнозування врожайності ягід, демонструє значний потенціал. Ці алгоритми надають розробникам потужні інструменти для створення точних прогнозів.

Використання цих алгоритмів в якості ансамблю моделей дозволяє використовувати сильні сторони кожного з них, щоб компенсувати можливі слабкі місця. Це забезпечує найкращу можливу точність прогнозування, що стає важливим інструментом для прийняття обґрунтованих рішень в агробізнесі.

Розроблений алгоритм навчання моделей прогнозування врожайності ягід представляє собою важливий інструмент для створення інтелектуального модуля прогнозування. Використання комбінації трьох різних моделей машинного навчання забезпечує можливість максимально ефективного використання їх сильних сторін для отримання більш точних прогнозів. Налаштування гіперпараметрів за допомогою Optuna гарантує, що кожна модель оптимально адаптована до конкретного набору даних. Застосування методу SelectKBest для відбору ознак сприяє концентрації на найважливіших з них, що підвищує ефективність моделі та зменшує обсяг необхідних обчислень.

Розроблений алгоритм валідації та тестування виявив свою ефективність у визначенні роботоздатності інтелектуального модулю прогнозування врожайності ягід. Цей процес надав можливість оцінити точність моделі та її спроможність генералізувати відомості на нові дані. Імплементація крос-валідації та використання метрик як-от середньоквадратичне відхилення (MSE) та коефіцієнт детермінації (R-квадрат) сприяли вимірюванню продуктивності ансамблевої моделі. Додатково, аналіз залишків надав цінні відомості про прогнозні здібності моделі та її здатність адекватно відображати відносини між ознаками та цільовою змінною.

В ході роботи було проведено аналіз даних, згенерованих за допомогою моделі симуляції опилення дикої чорниці (Wild Blueberry Pollination Simulation Model). Ця модель була підтверджена та валідована на основі полевих спостережень і експериментальних даних, які були зібрані в штаті Мен (США) та Канадських Марітимах протягом останніх трьох десятиліть.

Подальший аналіз кореляції між різними характеристиками та врожайністю виявив, що деякі ознаки, такі як "fruitset", "seeds" та "fruitmass", мають високу позитивну кореляцію з врожайністю, тоді як інші, наприклад "AverageRainingDays" та "RainingDays", мають виражену негативну кореляцію.

Були виконані два важливі кроки: проведено вибір ключових ознак за допомогою SelectKBest і сформовано ансамбль моделей, базуючись на XGBoost, LightGBM та CatBoost. Вибір ознак спрямований на акцентування найбільш значущих змінних, що сприяє підвищенню якості прогнозування. Ансамбль моделей поєднує можливості трьох сильних алгоритмів машинного навчання, що сприяє покращенню точності прогнозів. У сукупності ці кроки формують робустну основу для інтелектуального модулю прогнозування врожайності ягід.

Інтелектуальний модуль прогнозування, що базується на ансамблі моделей XGBoost, LightGBM та CatBoost, ефективно виконав поставлене завдання. Ці моделі були навчені з використанням оптимальних гіперпараметрів, визначених через процес оптимізації Optuna, що сприяло підвищенню їхньої ефективності. Оглядаючи графіки прогнозованих та дійсних значень, з'ясовується, що вони відображають подібні тренди, свідчачи про високу спроможність моделей адекватно прогнозувати врожайність. Це підкреслює, що застосування машинного навчання та оптимізації гіперпараметрів може бути надзвичайно корисним для прогнозування врожайності ягід. Така модель може стати в нагоді фермерам та агрономам для оптимізації процесу вирощування ягід та планування урожаю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Asare, E., Hoshide, A. K., Drummond, F. A., Criner, G. K., & Chen, X. (2017). Economic risk of bee pollination in Maine wild blueberry, Vaccinium angustifolium. Journal of Economic Entomology, 110(5), 1980–1992. doi: 10.1093/jee/tox191
2. Popp, J., Pető, K., & Nagy, J. (2013). Pesticide productivity and food security: A review. Agronomy for Sustainable Development, 33(1), 243–255. doi: 10.1007/s13593-012-0105-x
3. Drummond, F. (2019). Reproductive biology of wild blueberry (Vaccinium angustifolium Aiton). Agriculture, 9(4), 69. doi: 10.3390/agriculture9040069
4. Tasnim, R., Drummond, F., & Zhang, Y.-J. (2021). Climate change patterns of wild blueberry fields in downeast, Maine over the past 40 years. Water, 13(5), 594. doi: 10.3390/w13050594
5. Tuell, J. K., & Isaacs, R. (2010). Weather during Bloom affects pollination and yield of highbush blueberry. Journal of Economic Entomology, 103, 557–562. doi: 10.1603/EC09387
6. Drummond, F. A., Lund, J., & Eitzer, B. (2021). Honey bee health in Maine wild blueberry production. Insects, 12(6), 523. doi: 10.3390/insects12060523
7. Crane-Droesch, A. (2018). Machine learning methods for crop yield prediction and climate change impact assessment in agriculture. Environmental Research Letters, 13(11), 114003. doi: 10.1088/1748-9326/aae159
8. Ji, B., Sun, Y., Yang, S., & Wan, J. (2007). Artificial neural networks for rice yield prediction in mountainous regions. Journal of Agricultural Science, 145(3), 249–261. doi: 10.1017/S0021859606006691
9. Matsumura, K., Gaitan, C. F., Sugimoto, K., Cannon, A. J., & Hsieh, W. W. (2015). Maize yield forecasting by linear regression and artificial neural networks in Jilin, China. Journal of Agricultural Science, 153(3), 399–410. doi: 10.1017/S0021859614000392

10. Pantazi, X. E., Moshou, D., Alexandridis, T., Whetton, R. L., & Mouazen, A. M. (2016). Wheat yield prediction using machine learning and advanced sensing techniques. Computers and Electronics in Agriculture, 121, 57–65. doi: 10.1016/j.compag.2015.11.018
11. Conţiu, Ş., & Groza, A. (2016). Improving remote sensing crop classification by argumentation-based conflict resolution in ensemble learning. Expert Systems with Applications, 64, 269–286. doi: 10.1016/j.eswa.2016.08.006
12. Zhang, Y., & Wang, X. (2017). Yield prediction of summer maize in China using remote sensing data and a process-based model. International Journal of Remote Sensing, 38(4), 1025–1041. doi: 10.1080/01431161.2016.1268963
13. Balafoutis, A., Beck, P. S. A., Bellon-Maurel, V., & Vangeyte, J. (2020). Machine learning in agriculture: A review. Sensors, 20(10), 2881. doi: 10.3390/s20102881
14. Jha, P., Joshi, D., & Singh, P. K. (2019). Machine learning techniques for crop yield prediction: A systematic review. Computers and Electronics in Agriculture, 161, 272–293. doi: 10.1016/j.compag.2019.03.014
15. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419. doi: 10.3389/fpls.2016.01419
16. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational Intelligence and Neuroscience, 2016, 3289801. doi: 10.1155/2016/3289801
17. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 147, 70–90. doi: 10.1016/j.compag.2018.02.016
18. Singh, A., Ganapathysubramanian, B., & Singh, A. K. (2016). Machine learning for high-throughput stress phenotyping in plants. Trends in Plant Science, 21(2), 110–124. doi: 10.1016/j.tplants.2015.10.014
19. Mohanty, S. P., Mohanty, S., Hughes, D. P., & Salathé, M. (2017). Using machine learning algorithms to analyze large-scale plant phenomics data generated by

high-throughput imaging. Methods in Molecular Biology, 1578, 391–402. doi: 10.1007/978-1-4939-6857-2_23

20. Minervini, M., Abdelsamea, M. M., & Tsaftaris, S. A. (2016). Image-based plant phenotyping with incremental learning and active contours. Ecological Informatics, 36, 82–95. doi: 10.1016/j.ecoinf.2016.09.005
21. Araus, J. L., Cairns, J. E., & Sanchez, C. (2014). Are we ready to breed maize for photosynthesis? Journal of Experimental Botany, 65(13), 3479–3490. doi: 10.1093/jxb/eru273
22. Prasanna, B. M., Vasal, S. K., & Kassahun, B. (2001). Quality protein maize. Current Science, 81(10), 1308–1319.
23. Lobell, D. B., Cassman, K. G., & Field, C. B. (2009). Crop yield gaps: Their importance, magnitudes, and causes. Annual Review of Environment and Resources, 34, 179–204. doi: 10.1146/annurev.environ.041008.093740
24. Ray, D. K., Mueller, N. D., West, P. C., & Foley, J. A. (2013). Yield trends are insufficient to double global crop production by 2050. PLoS ONE, 8(6), e66428. doi: 10.1371/journal.pone.0066428
25. Tilman, D., Balzer, C., Hill, J., & Befort, B. L. (2011). Global food demand and the sustainable intensification of agriculture. Proceedings of the National Academy of Sciences, 108(50), 20260–20264. doi: 10.1073/pnas.1116437108
26. Godfray, H. C. J., Beddington, J. R., Crute, I. R., Haddad, L., Lawrence, D., Muir, J. F., Pretty, J., Robinson, S., Thomas, S. M., & Toulmin, C. (2010). Food security: The challenge of feeding 9 billion people. Science, 327(5967), 812–818. doi: 10.1126/science.1185383
27. FAO. (2020). The future of food and agriculture - Trends and challenges. Food and Agriculture Organization of the United Nations. Retrieved from http://www.fao.org/3/ca9529en/ca9529en.pdf
28. Lipper, L., Thornton, P., Campbell, B. M., Baedeker, T., Braimoh, A., Bwalya, M., Caron, P., Cattaneo, A., Garrity, D., Henry, K., Hottle, R., Jackson, L., Jarvis, A., Kossam, F., Mann, W., McCarthy, N., Meybeck, A., Neufeldt, H., Remington,

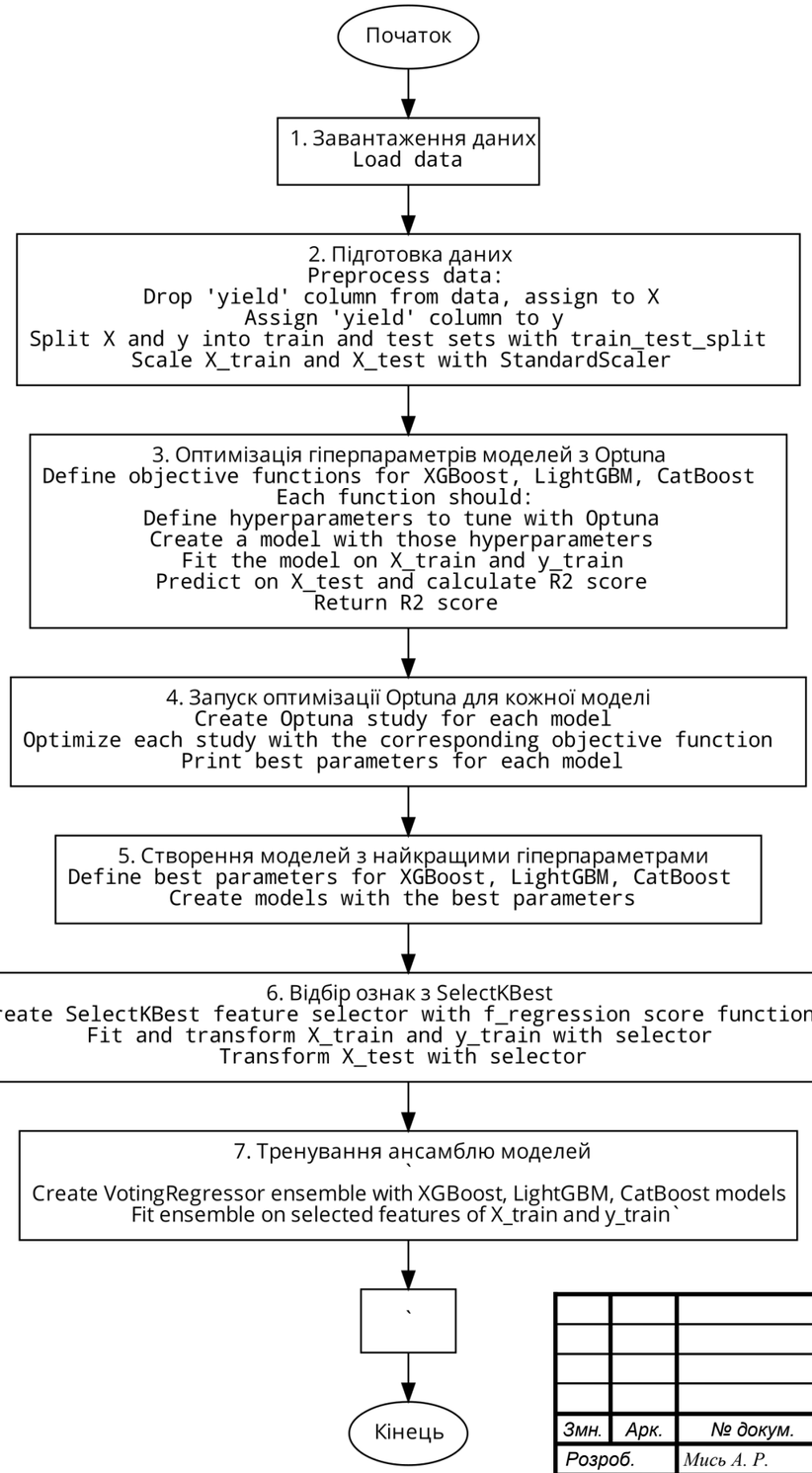
T., ... & Zougmore, R. (2014). Climate-smart agriculture for food security. Nature Climate Change, 4(12), 1068–1072. doi: 10.1038/nclimate2437

29. Pretty, J., Toulmin, C., & Williams, S. (2011). Sustainable intensification in African agriculture. International Journal of Agricultural Sustainability, 9(1), 5–24. doi: 10.3763/ijas.2010.0583
30. Giller, K. E., Witter, E., Corbeels, M., & Tittonell, P. (2009). Conservation agriculture and smallholder farming in Africa: The heretics' view. Field Crops Research, 114(1), 23–34. doi: 10.1016/j.fcr.2009.06.017
31. Rockström, J., Williams, J., Daily, G., Noble, A., Matthews, N., Gordon, L., Wetterstrand, H., DeClerck, F., Shah, M., Steduto, P., de Fraiture, C., Hatibu, N., Unver, O., Bird, J., & Sibanda, L. (2017). Sustainable intensification of agriculture for human prosperity and global sustainability. Ambio, 46(1), 4–17. doi: 10.1007/s13280-016-0793-6
32. Haddad, L., Hawkes, C., Webb, P., Thomas, S., Beddington, J., Waage, J., & Flynn, D. (2016). Food systems and diets: Facing the challenges of the 21st century. Global Food Security, 9, 1–9. doi: 10.1016/j.gfs.2016.06.001
33. FAO. (2019). The state of food and agriculture 2019. Moving forward on food loss and waste reduction. Food and Agriculture Organization of the United Nations. Retrieved from http://www.fao.org/3/ca6030en/ca6030en.pdf
34. HLPE. (2019). Agroecological and other innovative approaches for sustainable agriculture and food systems that enhance food security and nutrition. A report by the High Level Panel of Experts on Food Security and Nutrition of the Committee on World Food Security. Retrieved from http://www.fao.org/3/ca5602en/ca5602en.pdf
35. UN. (2015). Transforming our world: The 2030 Agenda for Sustainable Development. United Nations. Retrieved from https://sdgs.un.org/2030agenda
36. FAO. (2021). The state of food security and nutrition in the world 2021. Transforming food systems for food security, improved nutrition and affordable healthy diets for all. Food and Agriculture Organization of the United Nations. Retrieved from http://www.fao.org/3/cb4471en/cb4471en.pdf

37. HLPE. (2020). Food security and nutrition: Building a global narrative towards 2030. A report by the High Level Panel of Experts on Food Security and Nutrition of the Committee on World Food Security. Retrieved from http://www.fao.org/3/ca9950en/ca9950en.pdf
38. DeFries, R., Fanzo, J., Remans, R., Palm, C., & Wood, S. A. (2015). Metrics for land-scarce agriculture. Science, 349(6245), 238–240. doi: 10.1126/science.aaa5766
39. IPBES. (2019). Summary for policymakers of the global assessment report on biodiversity and ecosystem services of the Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services. IPBES Secretariat. Retrieved from https://ipbes.net/sites/default/files/2020-02/ipbes_global_assessment_report_summary_for_policymakers_en.pdf
40. Vermeulen, S. J., Campbell, B. M., & Ingram, J. S. (2012). Climate change and food systems. Annual Review of Environment and Resources, 37, 195–222. doi: 10.1146/annurev-environ-020411-130608
41. Benton, T. G., Bailey, R. M., & Fergusson, R. A. (2016). Biodiversity in agricultural landscapes: The role of ecosystem services. Cambridge University Press.
42. Vanlauwe, B., Descheemaeker, K., Giller, K. E., Huising, J., Merckx, R., Nziguheba, G., Wendt, J., Zingore, S., & Bielders, C. (2015). Integrated soil fertility management in sub-Saharan Africa: Unravelling local adaptation. Soil, 1(2), 491–508. doi: 10.5194/soil-1-491-2015
43. Lipyanina, H., Sachenko, S., Lendyuk, T., Brych, V., Yatskiv, V., & Osolinskiy, O. (2021). Method of detecting a fictitious company on the machine learning base. In *Advances in Computer Science for Engineering and Education IV* (pp. 138-146). Cham: Springer International Publishing.
44. Krysovatyy, A., Lipyanina-Goncharenko, H., Desyatnyuk, O., Sachenko, S., Lukasevych-Krutnyk, I., & Butrin-Boka, N. (2021, September). Classification Method of Fictitious Enterprises Based on Gaussian Naive Bayes. In *2021 IEEE*

16th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 2, pp. 210-216). IEEE.

45. Lipianina-Honcharenko, K., Lukasevych-Krutnyk, I., Butryn-Boka, N., Sachenko, A., & Grodskyi, S. (2021, October). Intelligent Method for Identifying the Fraudulent Online Stores. In *2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)* (pp. 218-222). IEEE.
46. Komar, M., Savenko, O., Sachenko, A., Lendiuk, T., Lipianina-Honcharenko, K., Hladiy, G., & Vasylkiv, N. (2022). Evaluation the Efficiency of Information Technology of Big Data Intelligence Analysis and Processing.
47. Qu, H. (2020, 12 вересня). Data for: Wild blueberry yield prediction using a combination of computer simulation and machine learning algorithms. Mendeley Data. https://data.mendeley.com/datasets/p5hvjzsvn8/1
48. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С. Методичні рекомендації до виконання кваліфікаційної роботи з освітньопрофесійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. – Тернопіль: ЗУНУ, 2021. – 56 с.

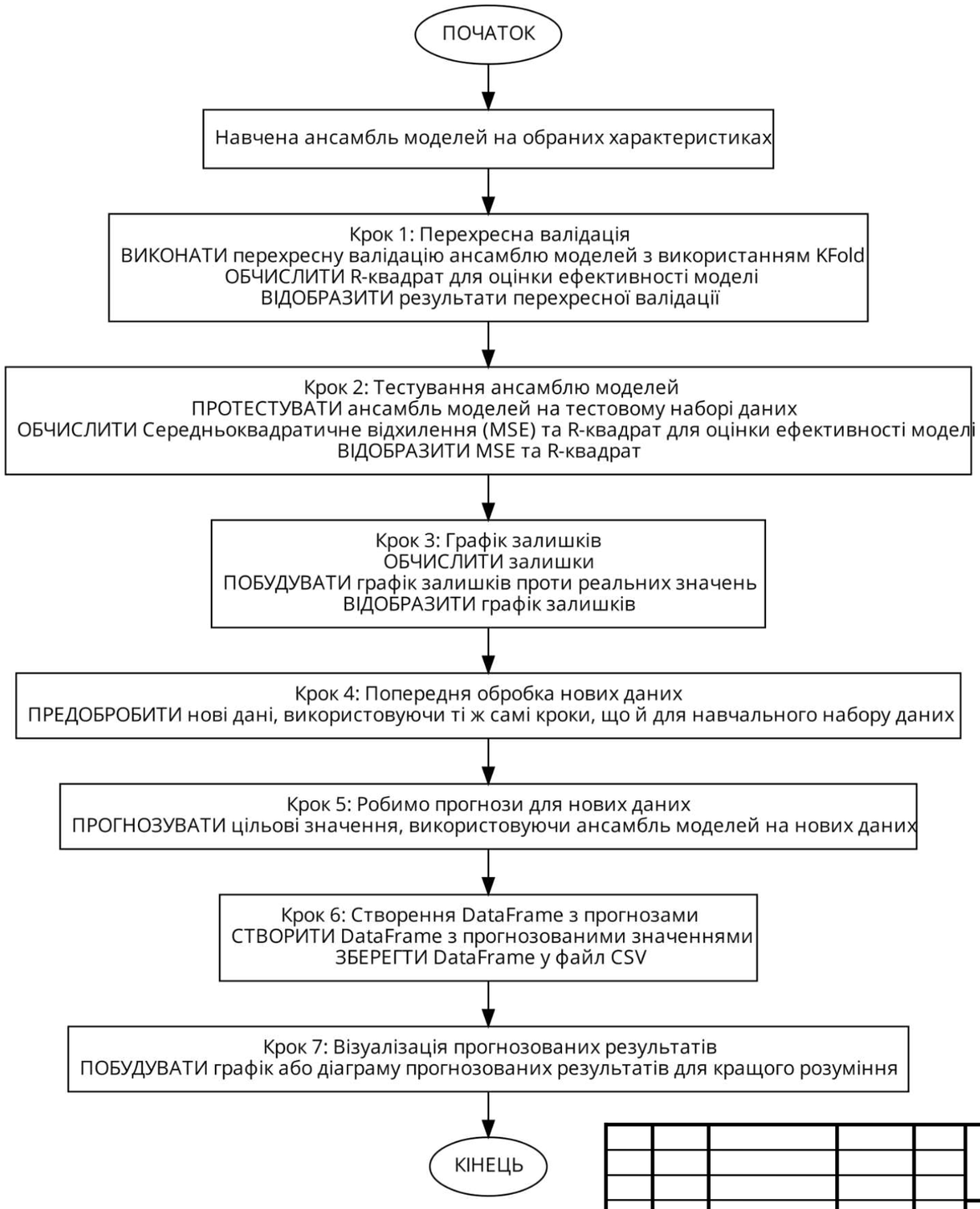


					ДП.КН. 8351441.001 А1
					Алгоритм навчання моделей прогнозування врожайності ягід
Змн.	Арк.	№ докум.	Підпис	Дата	
Розроб.		Мись А. Р.			
Перевір.		Ленджюк Т.В.			
Консультант					
Т. Контр.					
Н. Контр.		Ленджюк Т.В.			
Затверд.		Комар М.П.			

Літера	Маса	Масштаб

Аркуш 1	Аркушів 1

ЗУНУ.ФКІТ.КН-41



					ДП.КН. 8351441.001 А1	
Змн.	Арк.	№ докум.	Підпис	Дата		
Розроб.		Мись А. Р.			Алгоритм валідації моделей прогнозування врожайності ягід	Літера / Маса / Масштаб
Перевір.		Лендюк Т.В.				
Консультант						Аркуш 1 / Аркушів 1
Т. Контр.						
Н. Контр.		Лендюк Т.В.				ЗУНУ.ФКІТ.КН-41
Затверд.		Комар М.П.				