НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ

І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК 004.9:63-027.3

«ПОГОДЖЕНО»	**«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»**
Декан факультету	Завідувач кафедри комп'ютерних наук інформаційних технологій
Болбот І.М., д.т.н., професор	Голуб Б.Л., к.т.н., доцент
_____	_____
____ _____ 2024 р.	**____ _____ 2024 р.**

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «Система аналізу виробництва, продажу та запасів продукції сільського господарства та їх прогнозування»

Спеціальність _____ 122 "Комп'ютерні науки" _____

(код і назва)

Освітня програма _____Інформаційні управляючі системи та технології _____

(назва)

Орієнтація освітньої програми _____ освітньо-професійна _____

(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

_____к.т.н., доцент _____ _____ Голуб Б.Л. _____

(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи

_____д.т.н., професор_____ _____ _____Бушма О.В._____

(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ _____Кравченко О.В._____

(підпис) (ПІБ студента)

КИЇВ-2024

ЗМІСТ

ВСТУП

Актуальність. Сільське господарство є ключовим галуззю, яка забезпечує продовольчу безпеку та забезпечує постачання населенню продуктів харчування.

Через воєнні дії, зміни у світових тенденціях виробництва продукції та продовольчій кризі підприємства стикаються з викликами, які ускладнюють процеси обліку продукції, передбачення майбутнього попиту та виробництва, що не дозволяє виробництвам максимізувати як виробничі так і фінансові результати діяльності.

Рис.1 Зміна обсягів виробництва продукції через війну

Система аналізу виробництва, продажу та запасів продукції сільського господарства та їх прогнозування може стати важливим інструментом для виробництв в агропромисловому секторі. ЇЇ створення має на меті полегшити

аналіз виробленої продукції, обсяги її продажу та запасів. А також виявити закономірності та фактори впливу на ключові показники виробництва та автоматизувати та покращити процес прогнозування даних параметрів.

Мета полягає у дослідженні впровадження сучасних алгоритмів машинного навчання, статистичних та аналітичних методів, інформаційних технологій для автоматизації аналізу виробництва та прогнозування продукції в сільськогосподарському секторі та методів пошуку цінної інформації у великих обсягах даних. Це включає створення системи комплексного аналізу та прогнозування на основі великих даних, яка дозволить оптимізувати управлінські рішення.

Об'єкт дослідження: виробничі процеси підприємства, що включають управління виробництвом, запасами, попитом та прогнозування майбутніх виробничих потреб для забезпечення оптимального функціонування підприємства.

Предмет дослідження: Система аналізу виробництва, продажу та запасів продукції сільського господарства та їх прогнозування

Завдання дослідження :

- Дослідити предметну область для подальшого моделювання системи.
- Провести комплексне моделювання системи.
- Розробити архітектуру системи.
- Провести дослідження технологій, які будуть використовуватись в системі.
- Провести дослідження та опис основних алгоритмів та методів.
- Виконати апробацію різних алгоритмів прогнозування продуктивності та виявити найбільш ефективні для конкретних виробничих сценаріїв.
- Виділити основні апаратні та функціональні вимоги.

- Провести дослідження з використанням описаних технологій, методів та алгоритмів.
- Сформулювати висновки та пропозиції для подальшого розвитку системи та її адаптації до інших напрямків аграрного виробництва.

Методи дослідження включають наступні технології та методи:

1. Алгоритми машинного навчання були задіяні для побудови моделей прогнозування застосовуються методи машинного навчання з використанням бібліотек Scikit-Learn та Pytorch, що дозволяє підвищити точність передбачень.
2. Технологія OLAP була використана для багатовимірного аналізу даних з метою розрахунку ключових показників ефективності
3. Алгоритми Data Mining були використані для виявлення прихованих закономірностей та створення нових гіпотез використовуються методи кластеризації та дерев рішень,
4. Технології створення звітів такі як Power BI та Tableu були використані для створення інтерактивних дашбордів для узагальнення інформації.

В ході виконання дослідження було запропоновано вдосконалення алгоритмів обробки інформації для автоматизації процесів виявлення закономірностей у великих наборах даних шляхом інтеграції сучасних методів машинного навчання, зокрема, алгоритмів кластеризації та дерев рішень. Це вдосконалення дозволяє не тільки знаходити приховані патерни в даних, але й робить це більш ефективно за рахунок зменшення обчислювальних витрат і підвищення швидкості обробки великих масивів інформації.

Таким чином, удосконалені алгоритми обробки інформації забезпечують ефективніший аналіз великих наборів даних, що дозволяє виявляти приховані закономірності, підвищувати точність прогнозів і покращувати ключові показники ефективності виробництва.

Проблематика даної теми була досліджена та описана в тезах:

- Застосування цифрових технологій в процесах обліку якості продукції, аналізу продажів та запасів в агропідприємстві та приклади даних технологій

- Застосування методів машинного навчання у прогнозуванні виробництва сільськогосподарської продукції на прикладі системи аналізу виробленої продукції та її прогнозування

Так в даних тезах було досліджено застосування цифрових технологій в процесах обліку продажу продукції, прогнозування обсягів виробництва, застосування методів машинного навчання для реалізації прогнозування. Також було досліджено проблеми з якими стикаються виробники продукції через брак застосування цифрових технологій.

Структура дипломної роботи

Структура диплому Дана класифікаційна магістерська робота складається з наступних частин:

- Вступ
- Розділ 1– Аналіз предметної області
- Розділ 2 – Архітектура системи
- Розділ 3 – Основні технології та алгоритми
- Розділ 4 – Результати дослідження
- Висновки

У вступі описується актуальність обраної теми, її проблематику та мету розробки даної кваліфікаційної роботи.

У першому розділі більш широко розглядається предметна область, її основні процеси та функції системи. Також даний розділ містить основні моделі, які будуть використані в подальшому проектуванні.

У другому розділі розглянуто архітектуру системи, основні технології, які були використані у розробці системи та описано основні методи та алгоритми обробки даних.

У четвертому розділі було проведено дослідження з використанням досліджених та впроваджених технологій та методів.

У висновках було описано результати дослідження, доцільність впровадження системи та перспективи її використання

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Постановка завдання

Основне завданням магістерського дослідження полягає у розробці та впровадженні інформаційних технологій, технологій data science для підвищення якості та глибини аналізу та застосування методів машинного та глибокого навчання, статистичних моделей для прогнозування виробництва продукції. Це включає розробку системи, яка автоматизує процес обробки даних, а також реалізацію алгоритмів машинного навчання для оптимізації прогнозування обсягів продукції.

1.2 Розгляд основних процесів предметної області

Аналіз виробництва, продажу та запасів продукції сільського господарства є важливим аспектом управління, що забезпечує ефективність управління виробництвом від контролю виробленої продукції до її реалізації. Це включає аналіз структури виробленої продукції, яке залежить від багатьох змінних, аналіз кількості виробленої продукції, аналіз впливу багатьох факторів, аналіз доходів від проданої продукції.

Прогнозування є також важливою частиною контролю виробництва, оскільки воно дозволяє завчасно передбачати обсяг виробленої продукції, обсяги продажів на основі попередніх даних. Це може допомогти оптимізувати виробничі витрати, змінити акцент у напрямку виробництві. Завдяки застосуванню статистичних моделей і методів машинного навчання, аналітик може з більшою легкістю будувати гнучкі прогнози без потреби проводити аналіз часового ряду власноруч.

Система виконує дві основні функції, які будуть доступні користувачеві. Це аналіз даних виробленої продукції та показників її продажу. Під аналізом розуміється перегляд даних в консолідованому вигляді такому як графіки та метрики, які дозволять чіткіше зрозуміти об'єм та структуру виробленої

продукції. Також до цієї функції входить побудова моделей Data mining для виділення корисної інформації з даних. Другою функцією є побудова прогнозів на наступні періоди для кращого розуміння скільки підприємство зможе отримати продукції для збуту або зберігання.

1.3 Аналіз наявних рішень

Патент - Методи та системи для точного управління посівами - US12079874B2

Даний патент стосується способів і пов'язаних з ними систем для точного моделювання посівів і керування ними. Ці методи, як правило, використовують сезонну інформацію про погодні умови, які фактично спостерігаються на полі, для підготовки оновлених планів управління посівами в середині сезону.

Основними ж аналогами серед вже існуючих систем можна назвати Phocas - це платформа бізнес-аналітики та фінансового аналізу, яка допомагає організаціям приймати рішення на основі даних. Вона надає інструменти для аналізу даних, візуалізації та звітності.

Рис. 1-1-2 Phocas

Також схожою системою є SAP Analytics Cloud, що є частиною пакету SAP і пропонує економічно ефективні рішення для бізнес-аналітики (BI).

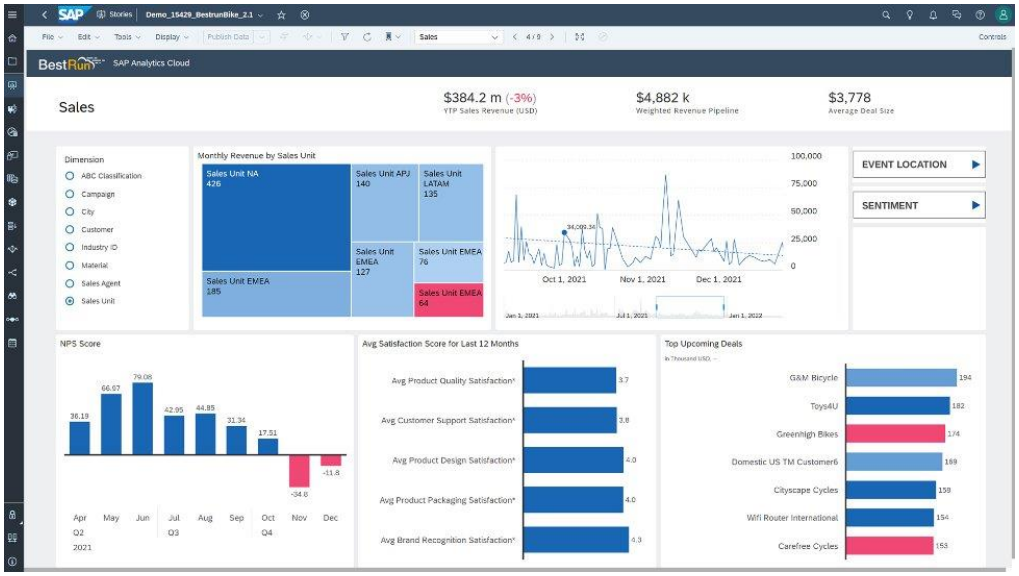


Рис. 1-3 SAP AC

Вона допомагає інтегрувати дані в режимі реального часу, будувати прогнози для прийняття більш обґрунтованих бізнес-рішень.

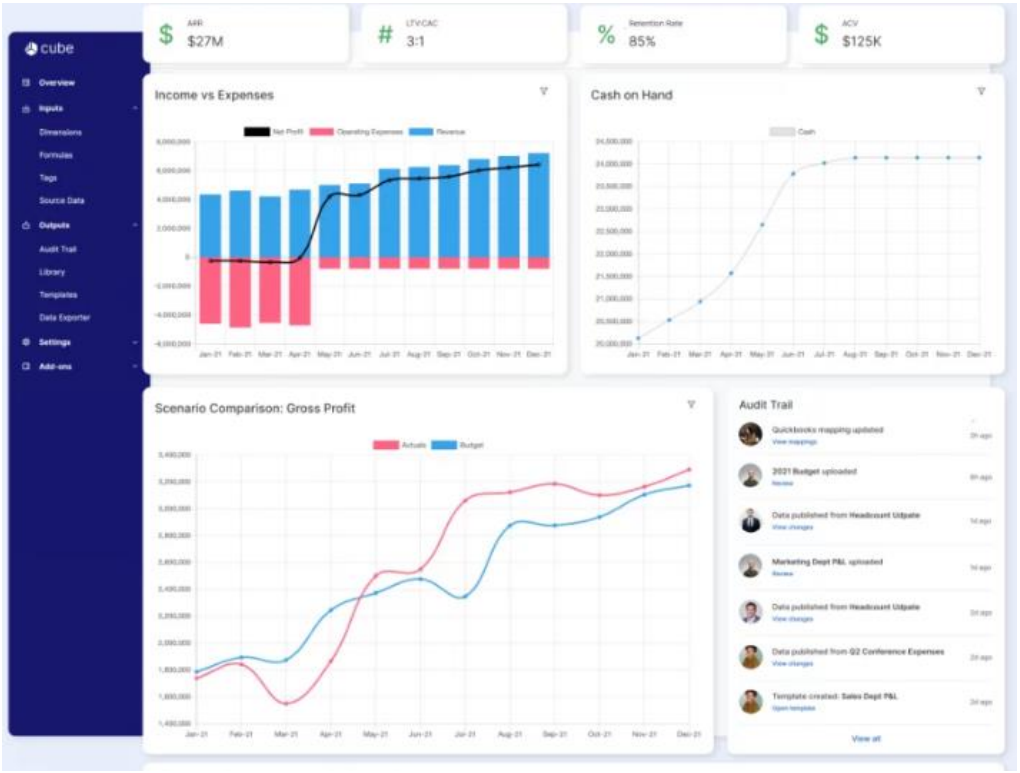


Рис. 1-3 Cube

Cube - це програмне забезпечення для фінансового планування та аналізу (FP&A), орієнтоване на роботу з електронними таблицями.

2. МОДЕЛЮВАННЯ СИСТЕМИ

Для побудови структури було обрано об’єктно орієнтований підхід побудови системи.

Було виділено такі основні класи та методи.

1. **Класи**:
    - **Product**: Представляє дані про кожен продукт
    - **DataCollector**: Отримує дані з різних джерел
    - **Analyzer**: Обробляє дані для виявлення трендів та закономірностей.
    - **Forecaster**: Створює прогноз, враховуючи історичні дані для прогнозування
    - **ReportGenerator**: Створює звіти.

2.1 Діаграма прецедентів

Діаграма прецедентів є однією з моделей, які використовуються в області аналізу предметної області для подальшого проектування системи.

Основними елементами діаграми є актори та прецеденти . Актори - це користувачі або зовнішні сутності, які взаємодіють з системою. Прецеденти - це конкретні дії або функції, які виконує система[4]. Взаємодія між акторами і прецедентами показує, які дії можуть виконувати актори в рамках системи Зробивши аналіз можна виділити 2 основних актора, перелічених нижче.

- Керівник Підприємства - Відповідає за розгляд звітності та прийняття рішення
- Аналітик – Відповідає за аналіз даних, формування звітності та складання прогнозів

В результаті на основі виділених прецедентів на основі предметної області можна побудувати діаграму прецедентів, яка розміщена на рис. 1

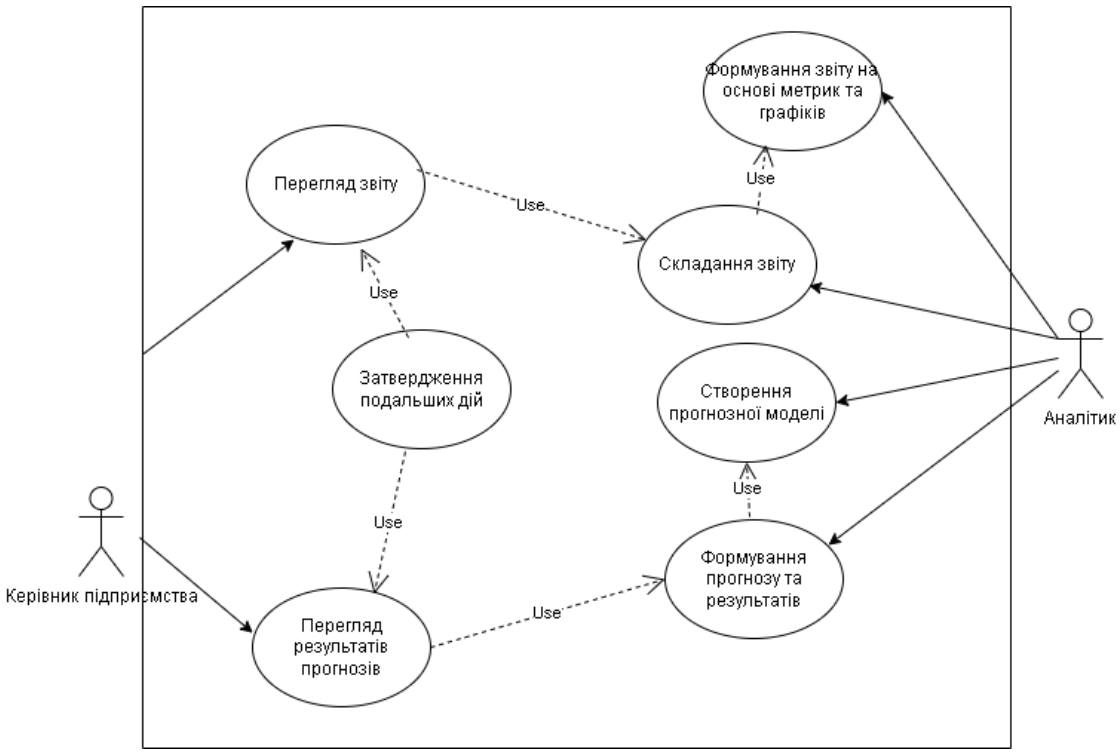


Рис. 2-4 Діаграма прецедентів

Наступною побудованою моделлю є діаграма активності.

Вона використовується для візуалізації послідовності дій або процесів, які відбуваються в системі. Діаграма діяльності дозволяє послідовно відобразити процеси і зв'язки між діями в системі. Вона допомагає зрозуміти порядок виконання дій, прийняття рішень та потоку даних.[6]

Дана діаграма відображає послідовність дій, які має виконати користувач для реалізації функціоналу системи. Так було виділено 2 основні сутності Аналітик та Керівник

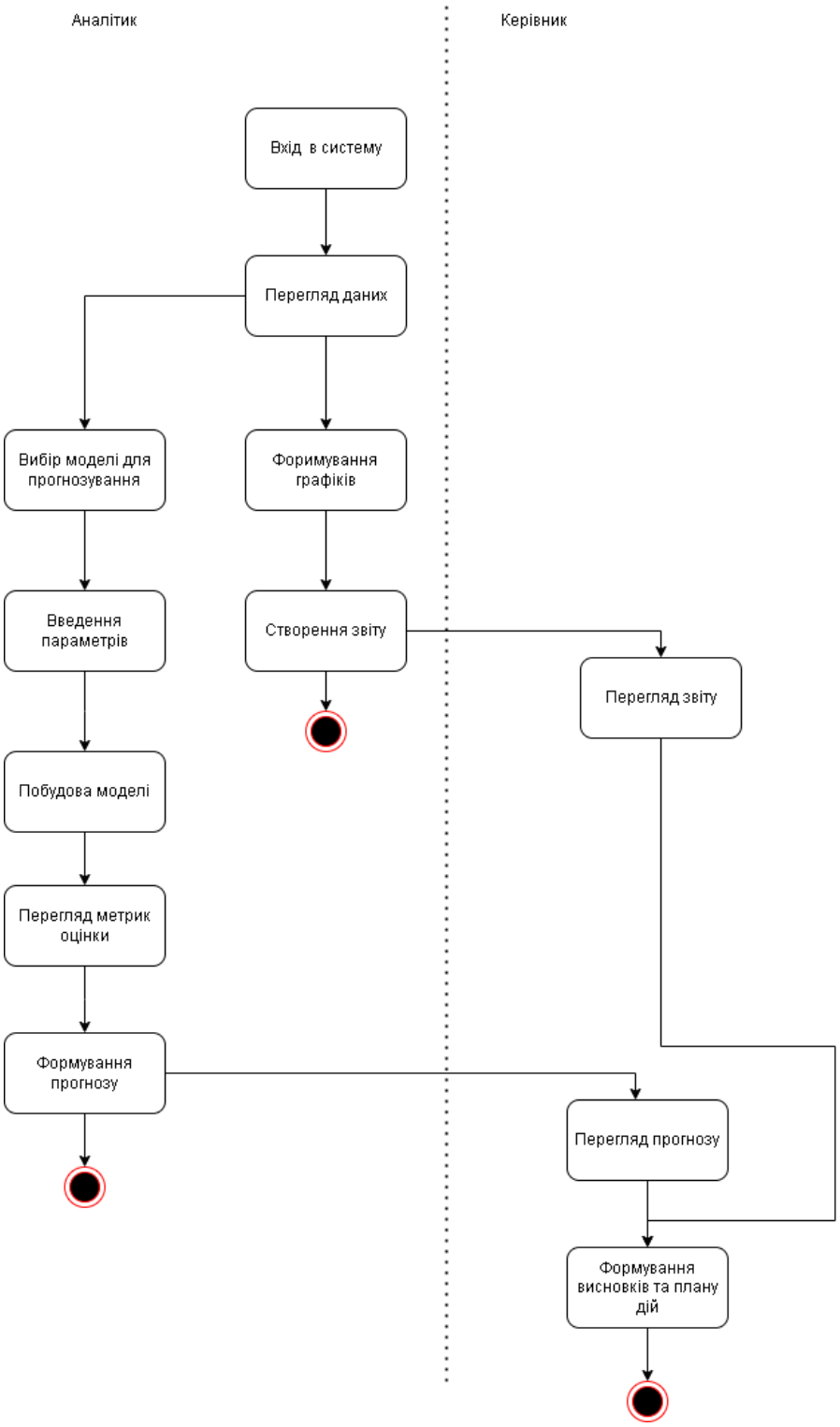


Рис. 2-5 Діаграма активності

Розглянувши всі аспекти предметної області, було складено відповідні діаграми, які дають розуміння щодо основних активностей, які можуть виконувати користувачі та які повинна реалізувати розроблювана система.

3. РОЗРОБКА СИСТЕМИ

3.1 Архітектура системи

Архітектура системи представляє собою модель, яка визначає основні вузли, їх взаємодію, модулі та зв'язки.

Так для розробки даної системи була обрана клієнт-серверна архітектура - система поділяється на клієнтську частину, яка взаємодіє з користувачем, і серверну, що зберігає дані та виконує основні обчислення.

На рис. 2 зображено архітектуру розроблюваної системи.

Основними фізичними вузлами є:

- Робоча станція працівника аналітика: Модуль аналізу даних
- Робоча станція контролера якості: Модуль введення та передачі інформації
- Сервер:
    - Оперативна база даних
    - Сховище даних

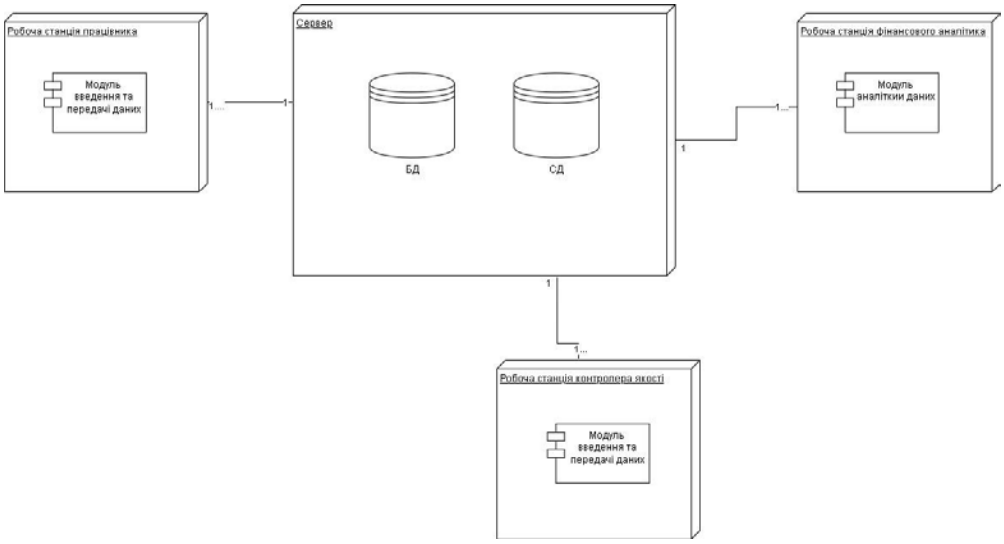


Рис. 3-6 Архітектура системи

3.2 Структура СД

Сховище даних представляє собою спеціалізовану систему управління даними, основним призначенням якого є забезпечення функціонування бізнес аналітики.

Сховище даних функціонує консолідована платформа для зберігання великих обсягів даних з різних джерел. Воно дозволяє отримувати цінні бізнес-інсайти з власних даних з метою удосконалення процесів прийняття рішень.

У ході розробки системи було створено сховища даних, спроектоване для проведення аналізу в різних аспектах.

Для збереження необхідної інформації у сховища даних були розроблені такі таблиці:

✦ Product_Category_dim

- Category_id – ідентифікатор категорії
- Category_name – назва категорії

✦ Product_dim

- Product_id – ідентифікатор продукта
- Product_name – назва продукту
- Category_id – ідентифікатор категорії
- Cost- собівартість продукції

✦ Date_dim

- Date_id – ідентифікатор дати
- Year – рік
- Month - місяць
- Day -день

✦ Fact_mes

- Product_id – ідентифікатор продукта
- Date_id – ідентифікатор дати
- Revenue - дохід

- Profit– прибуток
- Total Cost- повна собівартість продукції

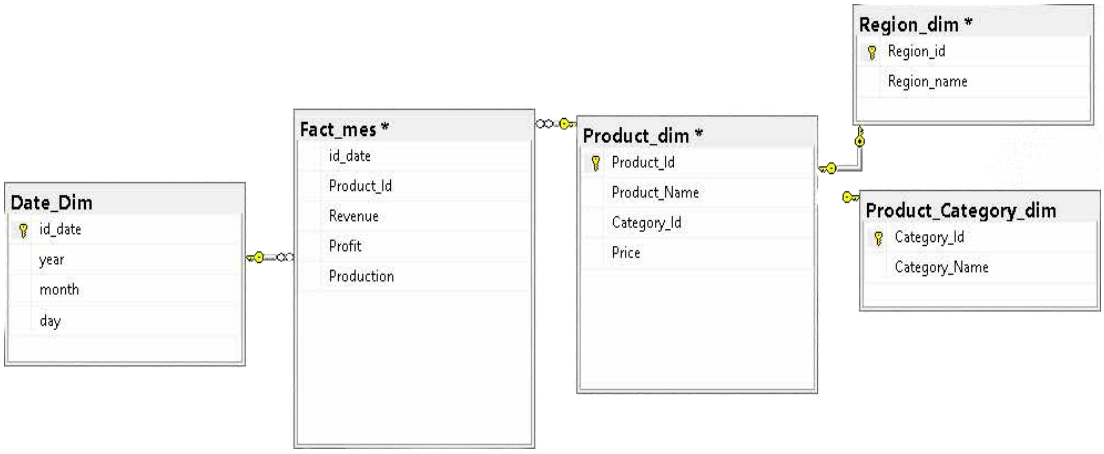


Рис. 3-7 Структура СД

3.3 OLAP

Служби аналізу SQL Server (SSAS) представляють собою OLAP сервер і механізм аналітики, який забезпечує розділення великих обсягів даних.

Для створення куба використовується середовище Visual Studio з розширенням SSAS. Перший етап передбачає визначення джерела даних - OLAP-бази даних або сховища даних. На основі обраного джерела дані імпортуються у систему. На рис. 8 показано процес встановлення з'єднання з джерелом даних за допомогою майстра Data Source, де вибирається раніше створене сховище даних. На заключному етапі у випадку розробки даної системи.

На рис. 9 Зображено створення джерела даних на основі, якого буде створено представлення даних та створено виміри та сам куб

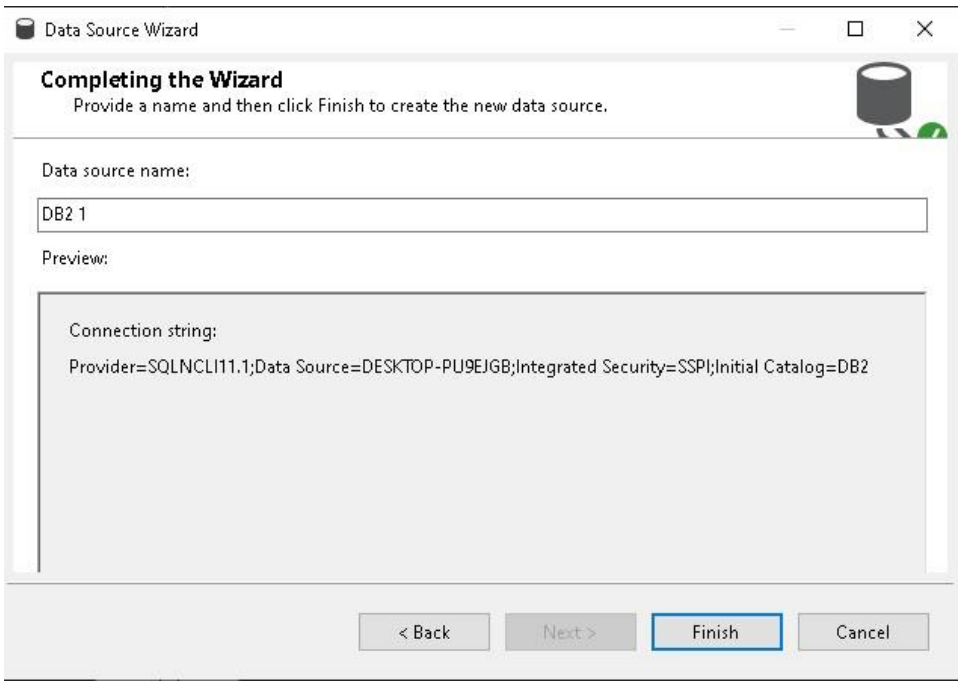


Рис. 3-8 Створення джерела даних

Далі було створене представлення даних, яке містить усі необхідні таблиці взяті з джерела даних Після цього потрібно створити виміри, на основі яких буде створено куб. На рисунках представлено скріншоти з поетапним створенням вимірів.

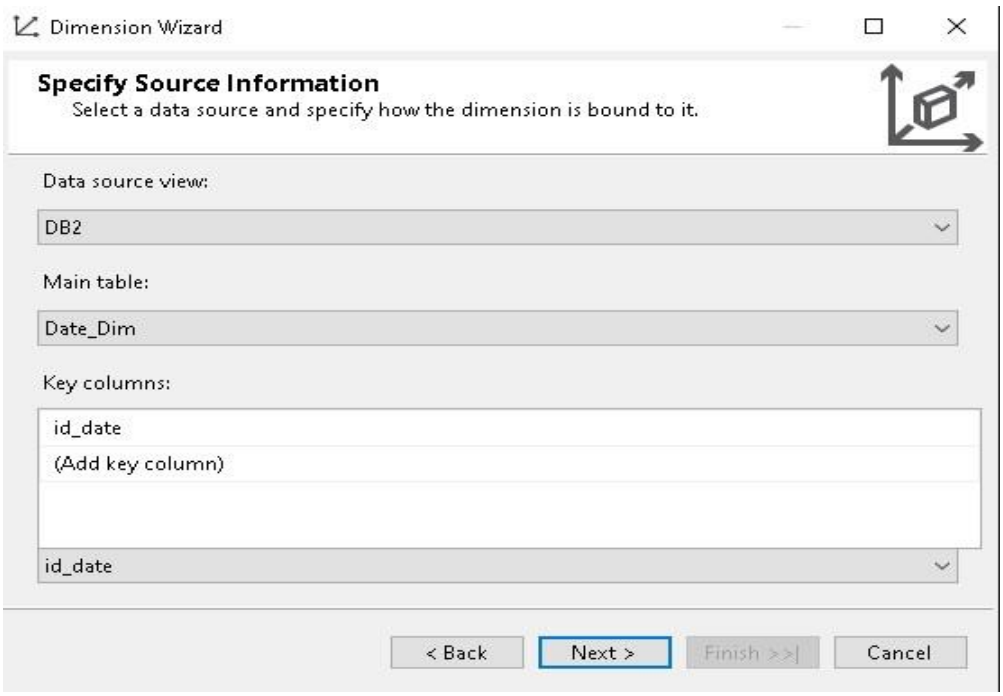


Рис. 3-9 Створення виміру

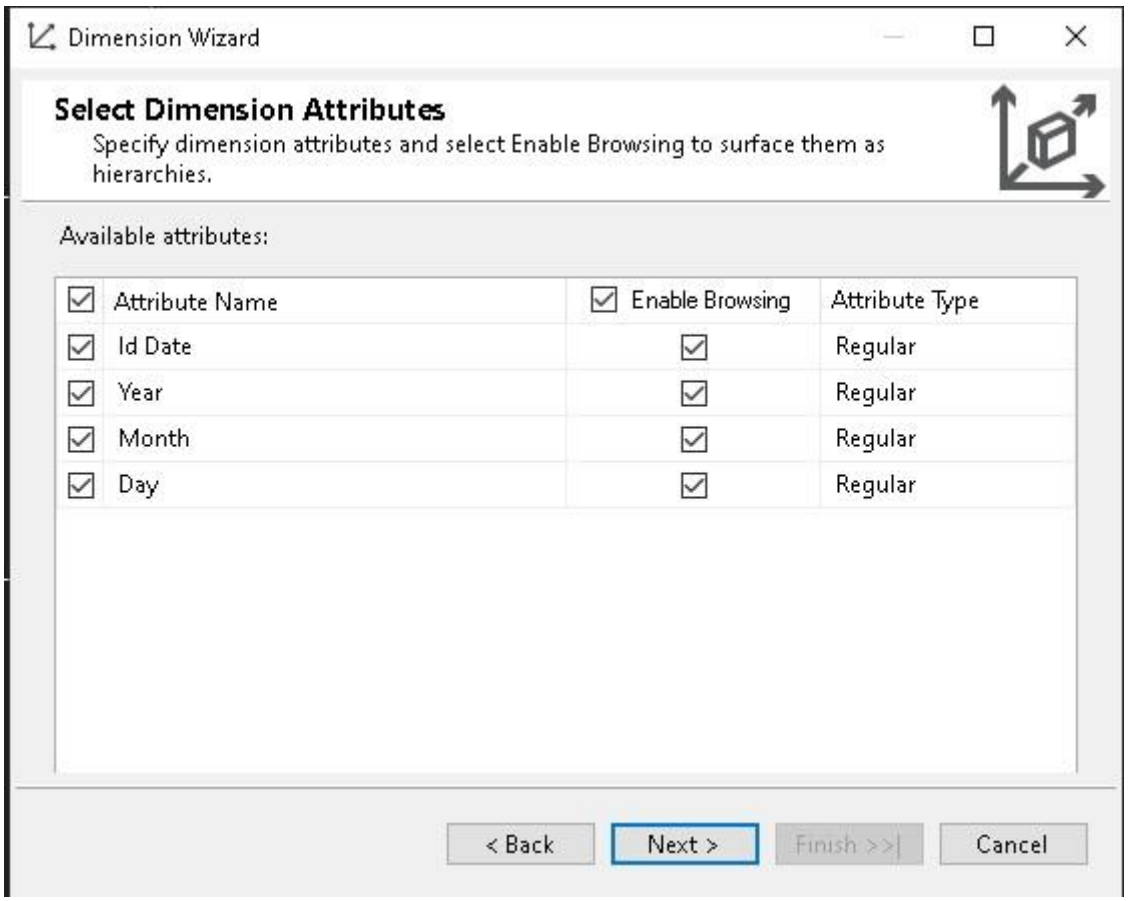


Рис. 3-10 Вибір атрибутів

Останнім етапом є створення куба за допомогою майстра створення кубів. На даному етапі ми обираємо таблицю фактів та виміри на основі яких буде створено куб

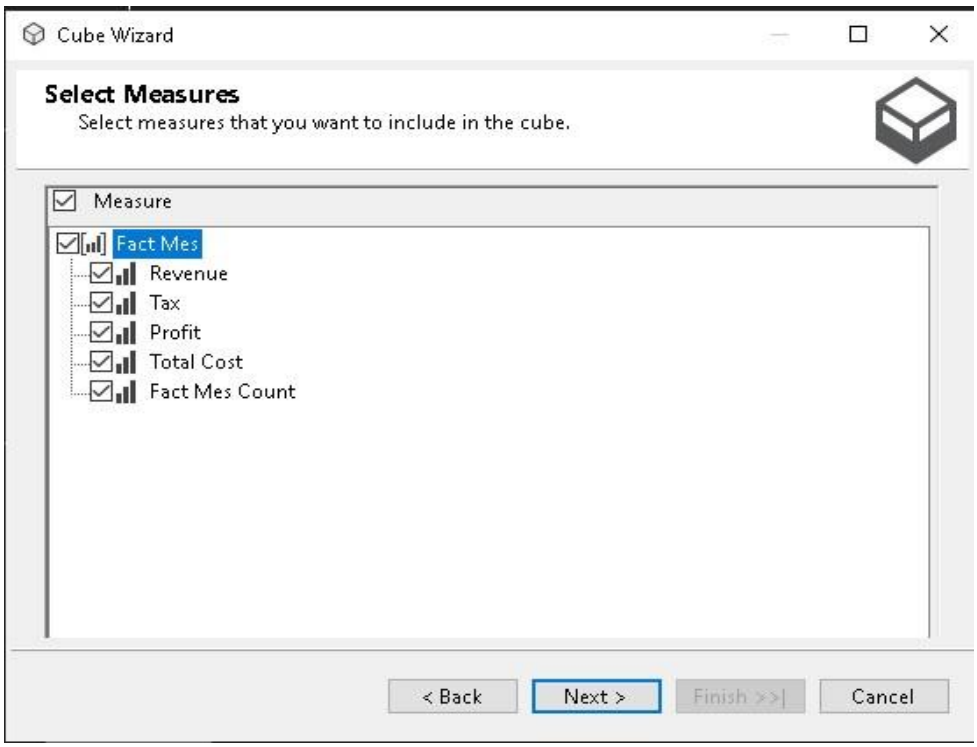


Рис. 3-11 Вибір мір

Структура куба представлена на рис. 3-12

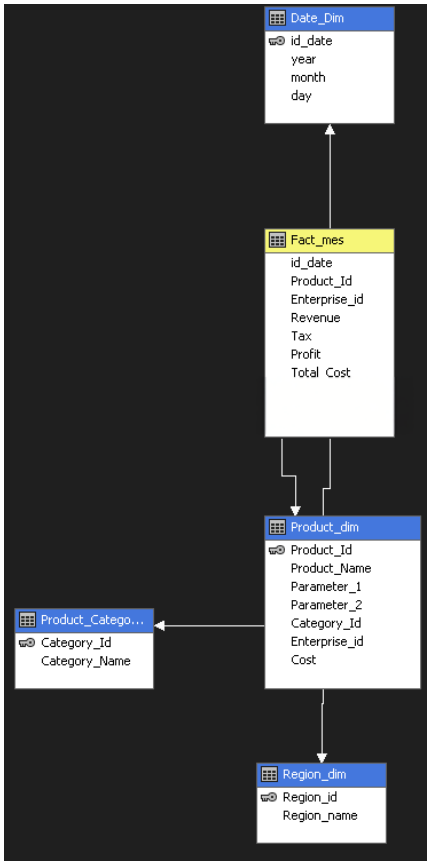


Рис. 3-12 Структура кубу

3.4 Заповнення кубу за допомогою SSIS

Передачу даних було виконано з використанням служби SQL Server Integration Services (SSIS). SSIS – це платформа для створення рішень для інтеграції та перетворення даних на рівні підприємства. Використовується для вирішення складних бізнес-задач. [8].

У інструменті SSIS існує служба Data Flow, яку використовували для наповнення таблиць вимірів та фактів.

Наповнення сховища даних (СД) базується на оперативній базі даних,)

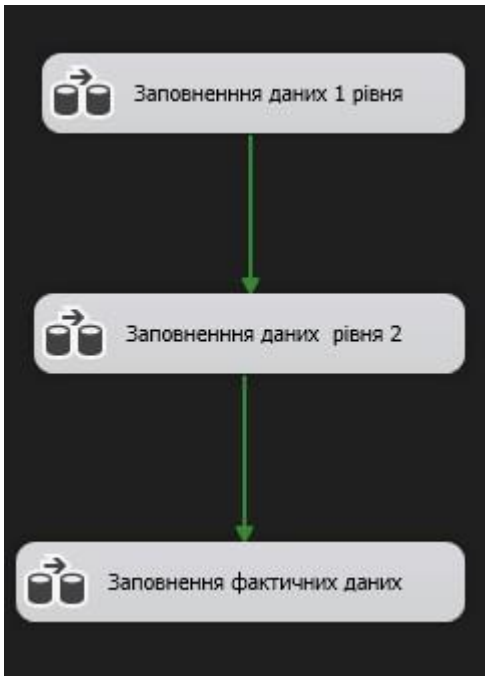


Рис. 3-13 Потоки даних

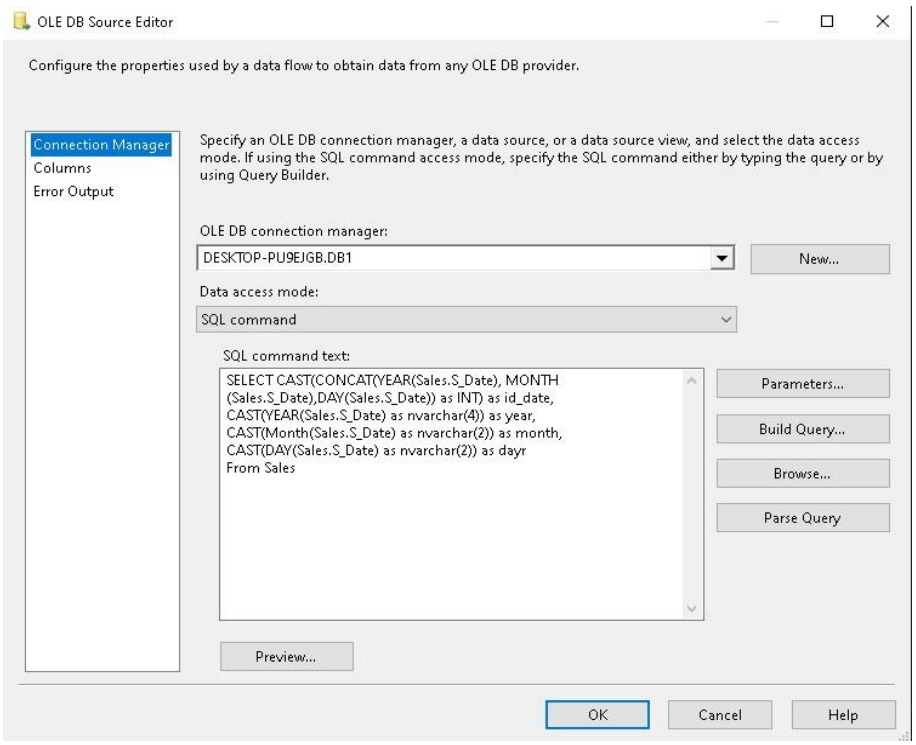


Рис. 3-14 Запит до оперативної бази даних

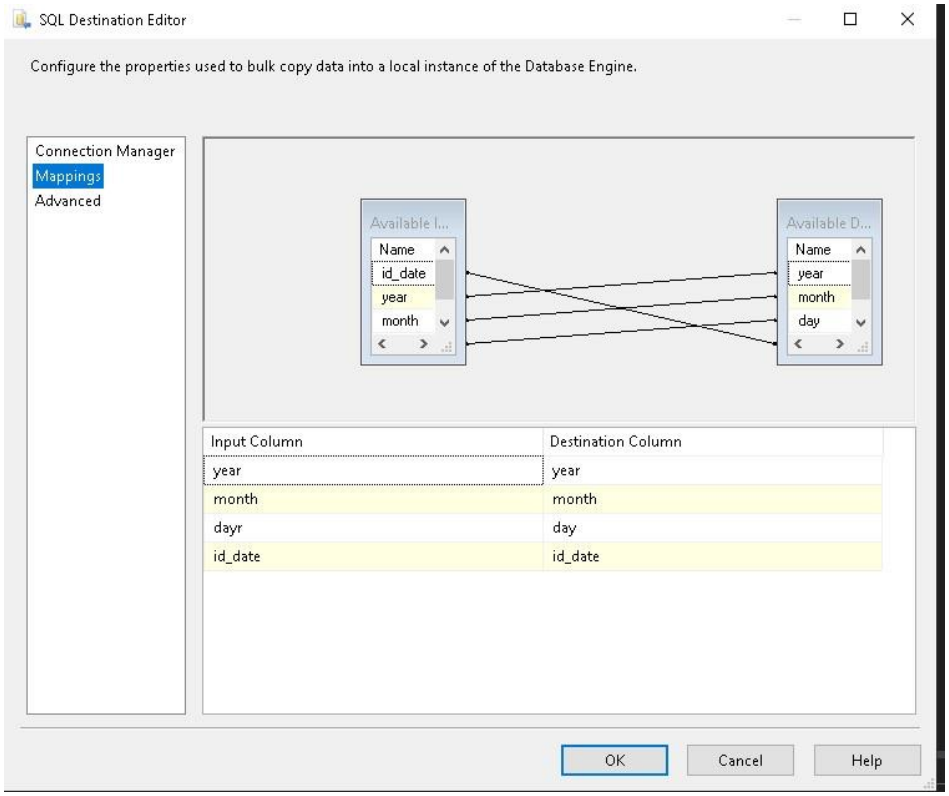


Рис. 3-15 Створення зв'язків

```
SELECT
DB1.dbo.Sales.Product_Id, DB2.dbo.Date_Dim.id_date, DB2.dbo.Product_dim.Enterprise_id,
(DB1.dbo.Sales.Unit_price*DB1.dbo.
Sales.Quantity) as Revenue ,
(((DB1.dbo.Sales.Unit_price* (100-DB1.dbo.Sales.Tax_Rate))/100-DB2.dbo.Product_dim.Cost)*DB1.dbo.
Sales.Quantity) as Profit,
DB1.dbo.Sales.Quantity* DB2.dbo.Product_dim.Cost as Total_cost,
((DB1.dbo.Sales.Unit_price* DB1.dbo.Sales.Tax_Rate)/100*DB1.dbo.
Sales.Quantity) as Tax
FROM DB1.dbo.Sales
JOIN DB2.dbo.Product_dim on DB2.dbo.Product_dim.Product_Id=DB1.dbo.Sales.Product_Id
JOIN DB2.dbo.Date_Dim ON Date_Dim.[year] =
CAST(YEAR(DB1.dbo.Sales.S_Date) as nvarchar(4))
AND Date_Dim.[month] = CAST(MONTH(DB1.dbo.Sales.S_Date) AS
NVARCHAR(2))
AND Date_Dim.[day] = CAST(DAY(DB1.dbo.Sales.S_Date) AS
NVARCHAR(2))
Group by DB1.dbo.Sales.Product_Id, DB1.dbo.Sales.Unit_price, DB1.dbo.Sales.Quantity, DB1.dbo.Sales.Tax_Rate,
DB2.dbo.Date_Dim.id_date,DB2.dbo.Product_dim.Enterprise_id,DB2.dbo.Product_dim.Cost
```

Рис. 3-16 Запит на дані

Для оцінки ефективності підприємства аналітики використовують коефіцієнт ефективності Key Performance Indicators. KPI представляє собою групу обчислень, пов'язаних із набором показників у багатовимірному кубі, що використовуються для оцінки успіху бізнесу. Зазвичай ці обчислення складаються з виразів мови багатовимірних виразів (MDX) або обчислюваних елементів. Крім того, KPI мають додаткові метадані, які надають інформацію про те, як слід відображати результати обчислень KPI [9].

Так для створення KPI необхідно обрати необхідні виміри з таблиці фактів та використовуючи MDX розрахувати необхідний показник. Так було розраховано три основних KPI, які узагальнювати успішність фінансової діяльності виробництва. Так на рис. 17-19 зображено код для створення мір, таких як Рентабельність – це кількість прибутку на одну гривню витрат

Кількість сплачених податків – показує скільки було сплачено податків на 1 гривню доходу , або ж скільки було сплачено податків на одну гривню витрат.

Дані показники показують наскільки виробництво є фінансово успішним та масштаби сплати податків.

Головною перевагою використання OLAP кубів є можливість зручного багатовимірного аналізу даних та KPI. Так аналітик може обрати необхідний період як, наприклад, місяць чи квартал, обрати вид продукції, або регіон і

побачити дані показники у розрізі цих вимірів, без використання коду у інтерактивному форматі

```
[Measures].[Tax]/ [Measures].[Profit]
```

Рис. 3-17 Створення міри

```
[Measures].[Profit]/ [Measures].[Total Cost]
```

Рис. 3-17 Створення міри ROI

```
[Measures].[Tax]/ [Measures].[Revenue]
```

Рис. 3-18 Створення міри TROI

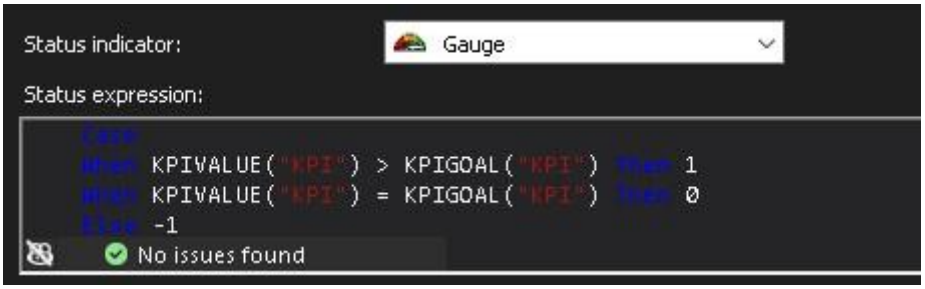


Рис. 3-19 Статус індиктора

Рис. 3-20 Відображення показників

3.5 Основні бібліотеки використані в розробці

У розробці системи буде використовувалась мова програмування Python.

Для побудови моделей, розрахунків та побудови інтерфейсу системи було використано такі бібліотеки:

Pandas - бібліотека для роботи з даними. Так основними структурами даних якими оперує дана бібліотека та які використовувались в системі є Dataframe та Series.

Наступною бібліотекою є NumPy. Це бібліотека, яка містить методи роботи з одновимірними та багатовимірними масивами, та основні методи роботи з ними.

Для роботи з моделями машинного навчання, data mining та обрахунку оціночних метрик була використана бібліотека Scikit-learn. Вона надає прості та ефективні інструменти для побудови моделей та їх оцінки. Вона містить широкий спектр алгоритмів для класифікації, регресії, кластеризації.

Для побудови різноманітних графіків було використано модулі matplotlib та plotly - ці бібліотеки використовуються для візуалізації даних. Вони містять функції для створення різноманітних графіків і діаграм, що допомагають більше зрозуміти дані.

Для побудови статистичних моделей було використано бібліотеку Statsmodel. Вона містить основні статистичні моделі, та методи для роботи з даними.

Для побудови моделей машинного навчання було використано бібліотеку PyTorch. Ця бібліотека має безліч методів для створення нейронних мереж та їх налаштування як для задач класифікації так і для регресійних проблем .

Для побудови основного графічного інтерфейсу для взаємодії з системою було використано. Streamlit.

3.6 Дослідження та застосування методів data mining

Data Mining— це процес виявлення прихованих закономірностей, тенденцій і корисної інформації в великих обсягах даних. В ньому використовуються методи статистики, машинного навчання та штучного інтелекту для пошуку та виділення цінної інформації, залежностей, які можуть допомогти керівнику у прийнятті управлінських рішень.

Так одним з основних методів data mining є найбільш поширений метод кластеризації - k-means. Кластеризація, в свою чергу, це процес створення кластерів(груп даних) на основі їх схожості за певними ознаками.

K-means - алгоритм кластеризації, який використовується для поділу набору даних на k окремих кластерів. Основна ідея алгоритму полягає в тому, щоб мінімізувати суму квадратів відстаней між точками даних і центроїдами кластерів.[18]

Основні кроки алгоритму K-means:

1. Вибирається кількість кластерів k, за оптимальним значенням оцінки силуету кластерів.
2. Випадково вибираються k точок даних як початкові центроїди кластерів або використовуються інші методи ініціалізації.
3. Кожна точка даних призначається до найближчого центроїда на основі обраної метрики відстані (зазвичай це евклідова відстань).
4. Обчислюється новий центроїд для кожного кластера як середнє значення всіх точок, що належать до цього кластера.
5. Повторюються кроки 3 і 4 до тих пір, поки центроїди не перестануть змінюватися або зміни будуть незначними (конвергенція).

Рис. 3-21 Приклад поділу на кластери

Для пошуку відповідної кількості кластерів був використаний алгоритм, який порівнює Silouette score, де він найвищий, таку кількості кластерів система обирає для кластеризації.

Так для виділення кластерів використовується метрика Silhoute score.

```
sil_scores = list()
for i in range(2, 13):
    kmeans = KMeans(n_clusters = i, random_state = 42)
    kmeans.fit(X)
    sil_scores.append(silhouette_score(X, kmeans.labels_))
plt.figure()
plt.plot(range(2, 13), sil_scores, color = 'salmon')
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette score')
plt.title('Silouette metrics')
plt.axvline(x = sil_scores.index(max(sil_scores))+2, linestyle = 'dotted', color = 'red')
plt.show()
```

Рис. 3-22. Алгоритм побудови графіку ліктя

Також в процесі data mining використовуються класифікаційні моделі для категоризації та прогнозування класу даних з використанням натренованої на минулих даних моделі. Такі моделі дають класифікувати на основі вже існуючих зразків нові дані, що може бути корисним у подальшому.

Random Forest Classifier - це ансамблевий метод машинного навчання, який використовується для вирішення задач класифікації. Він складається з великої кількості окремих дерев, кожне з яких є незалежним одне від одного[20].

```
scaled_df['Cluster']=kmeans6.labels_
X=scaled_df.drop('Cluster',axis=1)
y=scaled_df['Cluster']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.33, random_state=42)
rfc=RandomForestClassifier(max_depth=5,random_state=42)
rfc.fit(X_train,y_train)
results=rfc.predict(X_test)
print(accuracy_score(results,y_test))
scatter = plt.scatter(scaled_df['Yield'], scaled_df['Production'], c=scaled_df['Cluster'],
cmap='viridis')
plt.xlabel('Profit')
plt.ylabel('Total Cost')
plt.legend(handles=scatter.legend_elements()[0], labels=['High profitability', 'Low profitability'], title='Clusters')
plt.xlabel('Profit')
plt.ylabel('Total_Cost')
plt.title('Кластеризація')
plt.show()
```

Рис. 3-23 Код створення моделі та побудова графіку

Так для оцінки моделі використовується Confusion matrix. Ця матриця наочно показує кількість точно прогнозованих класів та розподіл помилок між ними.

```
def conf_matrix_plot(model, x_data, y_data):


    model_pred = model.predict(x_data)
    cm = confusion_matrix(y_data, model_pred, labels=model.classes_)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm,
                                  display_labels=model.classes_)

    disp.plot(values_format='')
    plt.show()
```

Рис.3-24 Код для побудови Confusion matrix

Вона дозволяє зрозуміти, наскільки добре модель справляється з класифікацією даних на основі фактичних та передбачених класів.

Матриця невідповідностей складається з чотирьох основних компонентів:

- TP (True Positive): Кількість правильно передбачених позитивних випадків
- TN (True Negative): Кількість правильно передбачених негативних випадків
- FP (False Positive): Кількість неправильно передбачених позитивних випадків

- FN (False Negative): Кількість неправильно передбачених негативних випадків

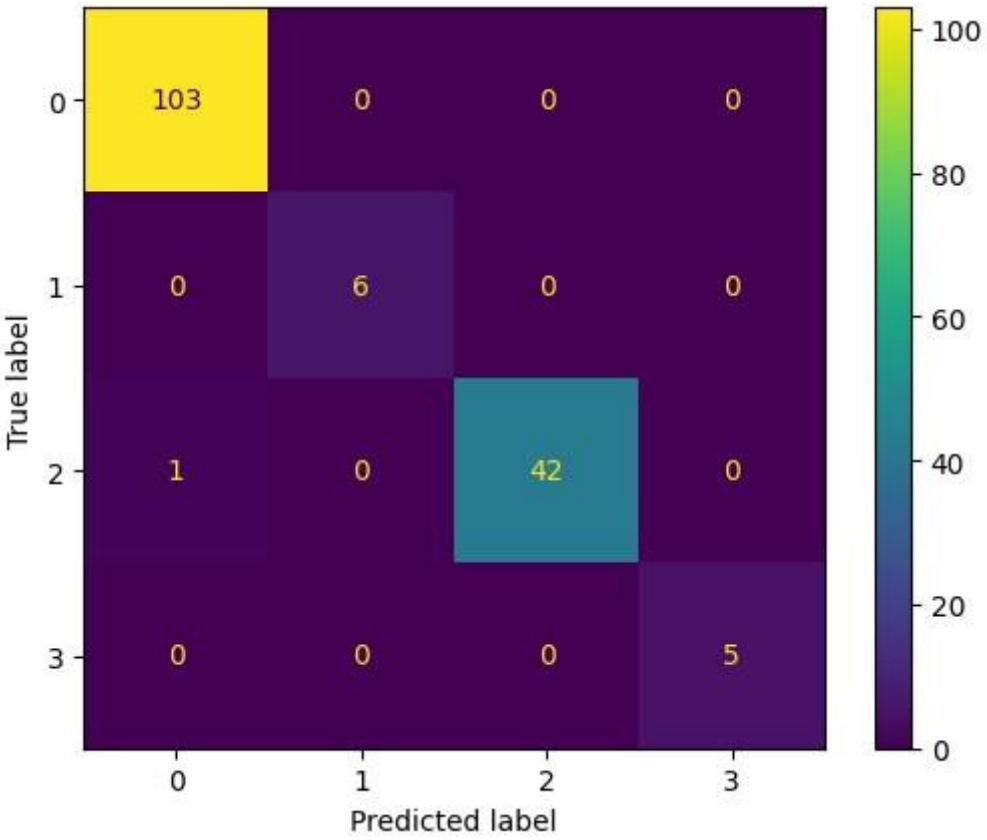


Рис. 3-25 Confusion matrix

3.7 Дослідження методів прогнозування

Прогнозування — це процес оцінки майбутніх подій на основі аналізу наявних даних. Даний процес широко використовується в сільському господарстві, для передбачення майбутніх значень кількості виробленої продукції, її продажу, для кращої підготовки підприємства або ж цілої країни до наступного сезону, що допоможе скоротити витрати та оптимізувати діяльність виробництва.

Прогнозування є важливим процесом в багатьох галузях, зокрема у виробничих, де своєчасне передбачення обсягів випуску продукції та необхідних ресурсів сприяє оптимізації виробничого циклу та підвищенню ефективності підприємства. Для досягнення цих цілей застосовуються різні моделі

прогнозування, які базуються на методах аналізу часових рядів, регресійному аналізі, а також алгоритмах машинного та глибокого навчання.

Процес прогнозування складається з кількох ключових етапів, які допомагають отримати точний і надійний прогноз на основі історичних даних. Так процес прогнозування можна поділити на такі етапи:

1. Визначення цілей прогнозування
2. Проведення сезоного декомпозування
3. Вибір моделі прогнозування
4. Вибір параметрів моделі
5. Тренування моделі
6. Оцінка моделі
7. Формування прогнозу

Для пошуку та виділення змістовних інсайтів про часовий ряд таких як тренд та сезонність використовують сезонне декомпозування.

Сезонне декомпозування – це метод аналізу часових рядів, який дозволяє розділити початковий часовий ряд на кілька складових частин.

Мета проведення декомпозування – виокремити трендову складову, сезонні коливання та шум. Це дасть можливість краще зрозуміти структуру часового ряду та підібрати кращу стратегію для побудови майбутнього прогнозу. Основними характеристиками часового ряду є:

- **Тренд** – відображає загальний напрямок зміни даних протягом тривалого часу.
- **Сезонна складова** – відображає повторювані коливання, які виникають через річні, квартальні чи місячні цикли.
- **Випадкова складова або залишок** – відображає випадкові коливання, що не піддаються прогнозуванню.

Сезонне декомпозування дозволяє зробити більш точні припущення щодо даних, що дасть змогу підібрати більш необхідну модель та налаштувати її параметри

Його можна провести за допомогою використання бібліотеки statmodels, а саме за допомогою методу seasonal_decompose(). Так звернувшись до даного методу можна отримати необхідну інформацію щодо часового ряду та його властивості.

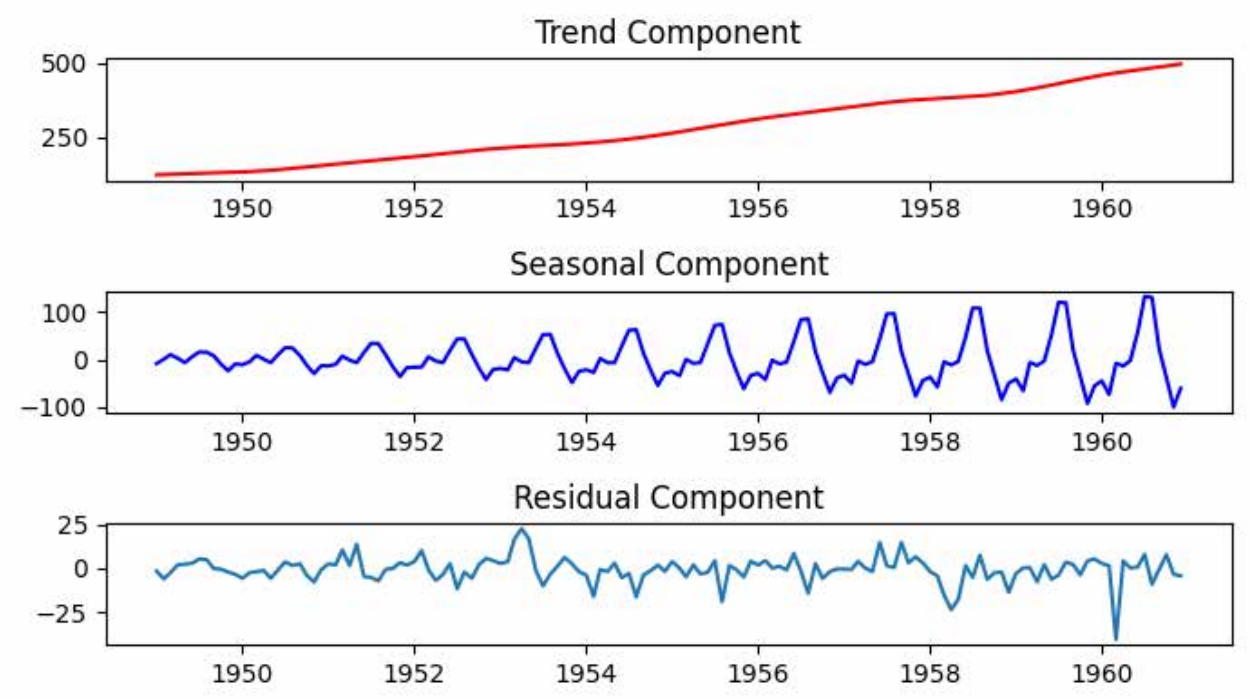


Рис. 3-26 Графік сезонного декомпозування

Наступним етапом є вибір моделі для прогнозування

Так можна виділити 3 основні категорії методів, які можна використати в прогнозуванні, кожен з яких використовується залежно від наявних даних, їх кількості та складності:

1. Класичні статистичні методи
2. Машинне навчання
3. Моделі глибокого навчання

Правильний вибір моделі прогнозування дозволяє підвищити точність прогнозів

Для прогнозування в виробництва в сільському господарстві можна використати багато методів як машинного навчання так і регресійних методів. Оскільки обсяги виробництва невпинно ростуть та залежать від, то на кількість виробленої продукції впливає тренд - це закономірність підйому або падіння показника в динаміці. Оскільки сільське господарство є сезонним видом діяльності то на майбутні значення також впливає сезонність. Сезонність у прогнозуванні — це повторюваний, передбачуваний патерн, який виникає у даних з певною регулярністю протягом фіксованих періодів часу.

Тому для вирішення цих складностей може підійти SARIMA модель, яка враховує дані параметри при прогнозування майбутніх значень.

SARIMA (Seasonal ARIMA) — це розширення моделі ARIMA, яке враховує сезонність даних. SARIMA додає до базової ARIMA-моделі сезонні компоненти для покращення точності прогнозів, якщо в ряді є повторювані патерни або циклічні зміни[13].

Основні параметри

1. **(P, D, Q, m)**, де:
    - P — порядок авторегресії,
    - D — порядок сезонного інтегрування
    - Q — порядок сезонної ковзної середньої
    - m — довжина сезонного циклу

```
def sarima_fit(stationary_df,order):
    model = smapi.tsa.sarima.SARIMA(stationary_df, order=order)
    results_SARIMA = model.fit()
    plt.plot(stationary_df)
    plt.plot(results_SARIMA.fittedvalues, color='red')
    plt.title('RSS: %.4f'%sum((results_SARIMA.fittedvalues - stationary_df[column[0]])**2))
    print('Plotting ARIMA model')
    return results_SARIMA
```

Рис. 3-27 Код створення моделі

Переваги SARIMA

- Ефективно працюює з часовими рядами де є сезонність та тренд

- Враховує сезонні коливання, що дозволяє моделювати циклічні патерни в часових рядах.

Наступним методом, який буде розглянуто, є метод LSTM.

LSTM — це тип нейронних мереж, які добре підходять для обробки послідовних даних, таких як часові ряди, текст або відео. Часто використовуються в задачах прогнозування, класифікації, обробки тексту [11].

LSTM є вдосконаленою версією RNN, розробленою для збереження довготривалих залежностей у послідовності даних. Вона є однією з найточніших моделей для задач прогнозування та обробки часових рядів, оскільки ефективно зберігає в своїй «пам’яті» дані за попередні періоди, які необхідні для кращої продуктивності моделі на часовому ряді.

Причиною обрання даного саме методу є те, що LSTM здатна утримувати більше історичних даних і враховувати їх при прогнозуванні, тоді як RNN може стикатися з проблемою "згасання градієнтів", що обмежує її здатність працювати з тривалими часовими рядами. Завдяки здатності до збереження інформації на довший період, LSTM краще обробляє дані з сезонними або довготривалими тенденціями, забезпечуючи краще навчання моделі та точність прогнозів навіть з використанням більш старіших історичних даних. Нижче на рис.3-28 зображена схема роботи LSTM

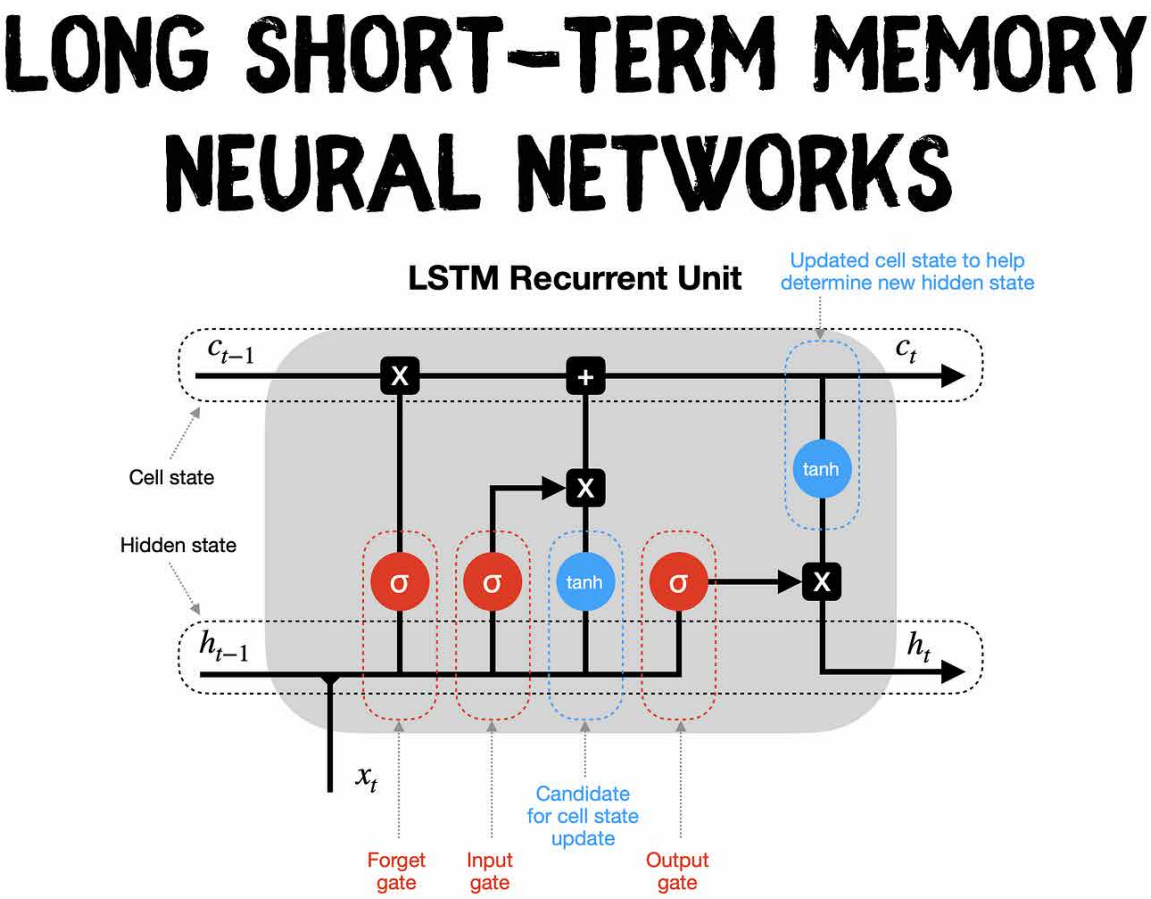


Рис. 3-28 Архітектура LSTM

Основні компоненти LSTM:

1. Комірка пам'яті
2. Вхідні ворота
3. Забуваючі ворота.
4. Вихідні ворота.

Принцип роботи LSTM:

- Вхідні ворота контролюють, яку інформацію з поточного стану потрібно зберегти.
- Забуваючі ворота видаляють непотрібну інформацію.
- Вихідні ворота використовують оновлену інформацію для обчислення виходу.

На рис.29 зображено код моделі

```python
class LSTMModel(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, dropout=0.2):
        super().__init__()
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers, batch_first=True, dropout=dropout)
        self.linear = nn.Linear(hidden_size, 1)

    def forward(self, x):
        out, _ = self.lstm(x)
        out = self.linear(out[:, -1, :])
        return out
```

Рис. 3-29 Клас моделі

Також був розроблений алгоритм тренування моделі. Так користувач вводить кількість циклів тренування , після кожного кроку модуль оптимізує функцію втрати за допомогою градієнтного спуску.

```python
def epoch_train(train_loader,test_loader,model,loss_fn,optimizer,device):
    # Set training parameters
    num_epochs = 1000  # Increased number of epochs
    train_hist = []
    test_hist = []
    for epoch in range(num_epochs):
        total_loss = 0.0
        model.train()
        for batch_X, batch_y in train_loader:
            batch_X, batch_y = batch_X.to(device), batch_y.to(device)
            predictions = model(batch_X)
            loss = loss_fn(predictions, batch_y)

            optimizer.zero_grad()
            loss.backward()
            optimizer.step()

            total_loss += loss.item()

        average_loss = total_loss / len(train_loader)
        train_hist.append(average_loss)

        model.eval()
        with torch.no_grad():
            total_test_loss = 0.0

            for batch_X_test, batch_y_test in test_loader:
                batch_X_test, batch_y_test = batch_X_test.to(device), batch_y_test.to(device)
                predictions_test = model(batch_X_test)
                test_loss = loss_fn(predictions_test, batch_y_test)

                total_test_loss += test_loss.item()

            average_test_loss = total_test_loss / len(test_loader)
            test_hist.append(average_test_loss)

        if (epoch + 1) % 10 == 0:
            st.write(f'Epoch [{epoch + 1}/{num_epochs}] - Training Loss: {average_loss:.4f}, Test Loss: {average_test_loss:.4f}')
    return train_hist,test_hist,model
```

Рис. 3-30 Алгоритм тренування моделі

Після виконання даного процесу користувач побачить криву навчання на якій можна побачити навчання моделі.

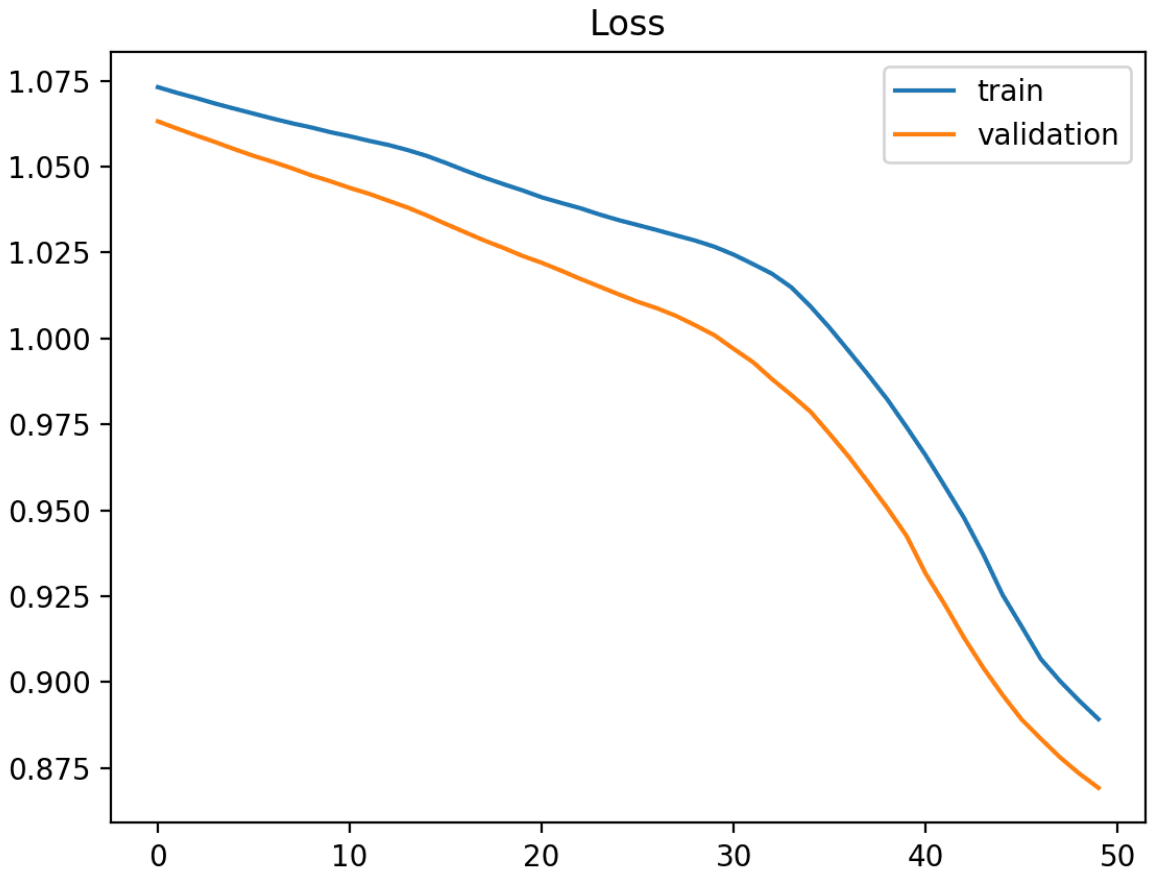


Рис. 3-31 Приклад кривої навчання

Ще одним методом прогнозування є використання моделей машинного навчання з використанням алгоритмом градієнтного бустингу. Одним з таких методів є XGB

XGBoost є одним із найпотужніших алгоритмів машинного навчання, який базується на техніці градієнтного бустингу. Він часто використовується для вирішення завдань класифікації та регресії, особливо в задачах з великими обсягами даних і складними залежностями.[21].

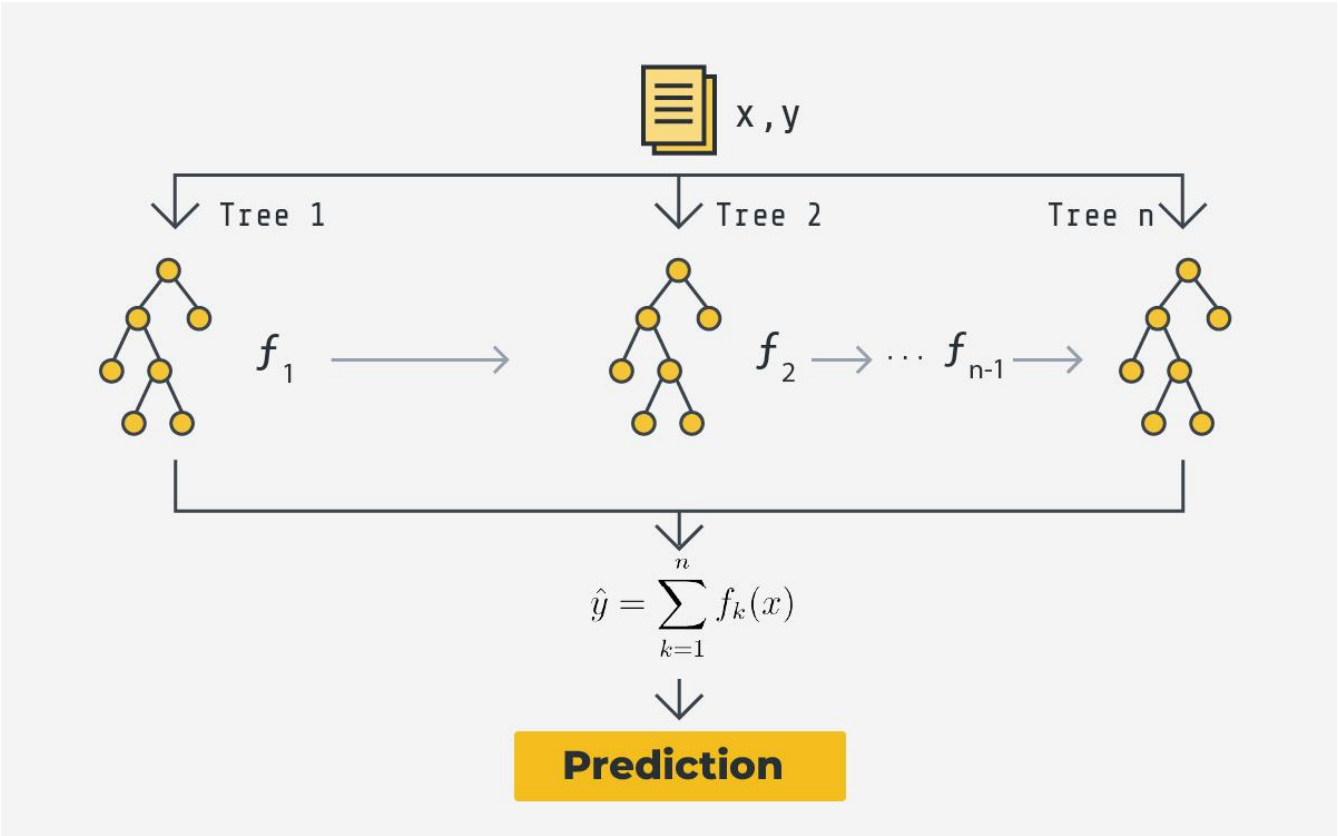


Рис. 3-31 Архітектура XGB

Основні гіперпараметри XGBoost:

1. Learning Rate.
2. n_estimators
3. max_depth
4. Subsample
5. colsample_bytreeregularization_alpha (L1 регуляризація) і regularization_lambda (L2 регуляризація)

На рис. 32 та 33 зображено код створення ознак часового ряду та код створення моделі та побудови порівняльного графіку

```python
def create_features(df):
    df = df.copy()
    df['quarter'] = df.index.quarter
    df['month'] = df.index.month
    df['year'] = df.index.year
    return df
```

Рис. 3-32 Створення ознак для тренування модулі

```python
def forecast_xgb(model,steps,df_forecast):
    dates=pd.date_range(start=df_forecast.index[-1],periods=steps,freq='MS')
    xgb_future_pred=pd.DataFrame({'quarter':dates.quarter,'month':dates.month,'year':dates.year},index=dates)
    xgb_future_pred['predictions']=model.predict(xgb_future_pred)
    fig = go.Figure()
    fig.add_trace(go.Scatter(x=xgb_future_pred.index, y=xgb_future_pred['predictions'], line=dict(color='blue')))
    st.plotly_chart(fig,use_container_width=True)
```

Рис. 3-33 Код моделі

Якщо ж минулі спостереження мають вплив на фактичні значення часового ряду, то аналітик має справу з автокореляцією часового ряду, і тому для побудови моделі необхідно замислитись над створенням послідовності лагів з якими корелює фактичне значення для побудови більш точної та сильнішої моделі.

Лаги в аналізі часових рядів - це термін часу відставання одного спостереження від іншого, пов'язаного з ним.

У часових рядах лаги представляють собою зсув значень на фіксований інтервал часу.

Використання лагів у часових рядах має як переваги, так і недоліки. Лаги допомагають моделі враховувати автокореляцію. Це дозволяє моделі краще ловити динаміку часового ряду і покращувати точність прогнозів, оскільки модель отримує більше інформації про часовий ряд.

Також однією з переваг є простота реалізації, адже використання лагів є доволі інтуїтивним способом обробки часових рядів, що не потребує значних обчислювальних ресурсів.

Однак використання лагів має недоліки. Під час створення лагів на початку ряду виникають пропущені значення, які потрібно або видаляти, або обробляти, що зменшує обсяг даних для аналізу, особливо якщо ряд є коротким.

Так для визначення доцільності використання лагів можна використати графік pacf, який відображає автокореляцію між попередніми вимірюваннями. Приклад даного графіку можна побачити на рис. 3-33.

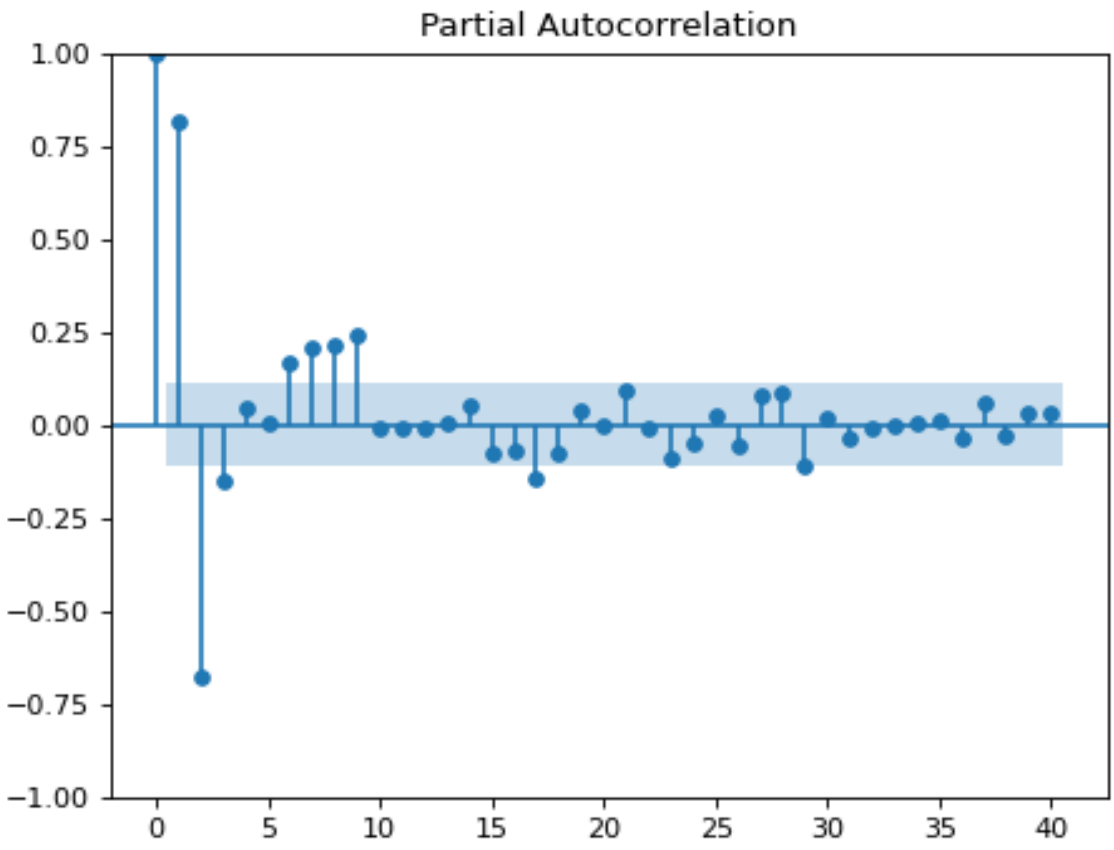


Рис. 3-33 Графік автокореляції

Так для створення лагів була створена функція зображена на рис. 3-34.

```python
def create_lag_features(df, lags=2):
  df_w_lags=df.copy()
  y = df_w_lags.loc[:, 0]
  for lag in range(lags):
    df_w_lags[f"lag_{lag + 1}"] = y.shift(lag + 1)
  return df_w_lags[lags:]

df_w_lags = create_lag_features(df, lags=12)
```

Рис. 3-34 Створення лагів

Нижче наведено результат створених лагів

2017-09-01	98.6154	108.9312	112.1538	102.1532	92.0805	88.3530	101.0396	99.4901	114.8505	112.7694	92.8900	91.4867	102.7637
2017-10-01	93.6137	98.6154	108.9312	112.1538	102.1532	92.0805	88.3530	101.0396	99.4901	114.8505	112.7694	92.8900	91.4867
2017-11-01	97.3359	93.6137	98.6154	108.9312	112.1538	102.1532	92.0805	88.3530	101.0396	99.4901	114.8505	112.7694	92.8900
2017-12-01	114.7212	97.3359	93.6137	98.6154	108.9312	112.1538	102.1532	92.0805	88.3530	101.0396	99.4901	114.8505	112.7694
2018-01-01	129.4048	114.7212	97.3359	93.6137	98.6154	108.9312	112.1538	102.1532	92.0805	88.3530	101.0396	99.4901	114.8505

Рис. 3-35 Вигляд таблиці лагів

3.8 Основні метрики для оцінки прогнозів та їх принципи розрахунку

Оцінка якості прогнозу — це важливий етап в процесі моделювання, який дозволяє оцінити точність та надійність моделі. Нижче наведені основні метрики, що використовуються для оцінки якості прогнозування, та принципи їх розрахунку. Для порівняння якості моделей було обрано такі метрики:

1. Середня абсолютна похибка (MAE — Mean Absolute Error)

MAE показує середнє значення абсолютної різниці між прогнозованими та фактичними значеннями. Ця метрика надає інформацію про середню величину похибки, незалежно від того чи прогнозовані дані вищи чи нижчі за фактичні, тому добре підходить для інтерпретації помилки в тих самих одиницях, що і значення прогнозу.

2. Середньоквадратична похибка (MSE — Mean Squared Error)

MSE показує середнє значення квадратів різниць між реальними та прогнозованими значеннями. Через зведення до квадрата, великі помилки мають більший вплив на значення MSE.

4. Середня абсолютна відносна похибка (MAPE — Mean Absolute Percentage Error)

MAPE показує середню абсолютну похибку у відсотках. Вона обраховується у відсотках, що робить її дуже легкою для інтерпретування, навіть для тих хто не має знань про досліджувані дані.

3.9 Використання технології PowerBi для побудови звітів

Power BI є одним із найпотужніших інструментів для візуалізації та аналізу даних, який ідеально підходить для побудови звітів у системах аналізу виробництва продукції та її прогнозування. Ця платформа дозволяє працювати з великими обсягами даних, перетворюючи їх на інтуїтивно зрозумілі візуалізації, які допомагають приймати обґрунтовані управлінські рішення.

Power BI підтримує підключення до різних джерел, таких як корпоративні системи управління підприємством (ERP), бази даних, Excel-файли або хмарні сервіси та OLAP куби. Це дає можливість з легкістю використовувати різні джерела даних, що дозволяє консолідувати більше інформації для прийняття рішень аналітиком або керівником виробництва

Так отримані дані можна обробляти за допомогою Power Query. За допомогою даного інструмента можна здійснити додаткове очищення даних від зайвих або неповних записів, а також перетворити їх у вигляд, придатний для аналізу за допомогою використання функцій півотування або ж групування.

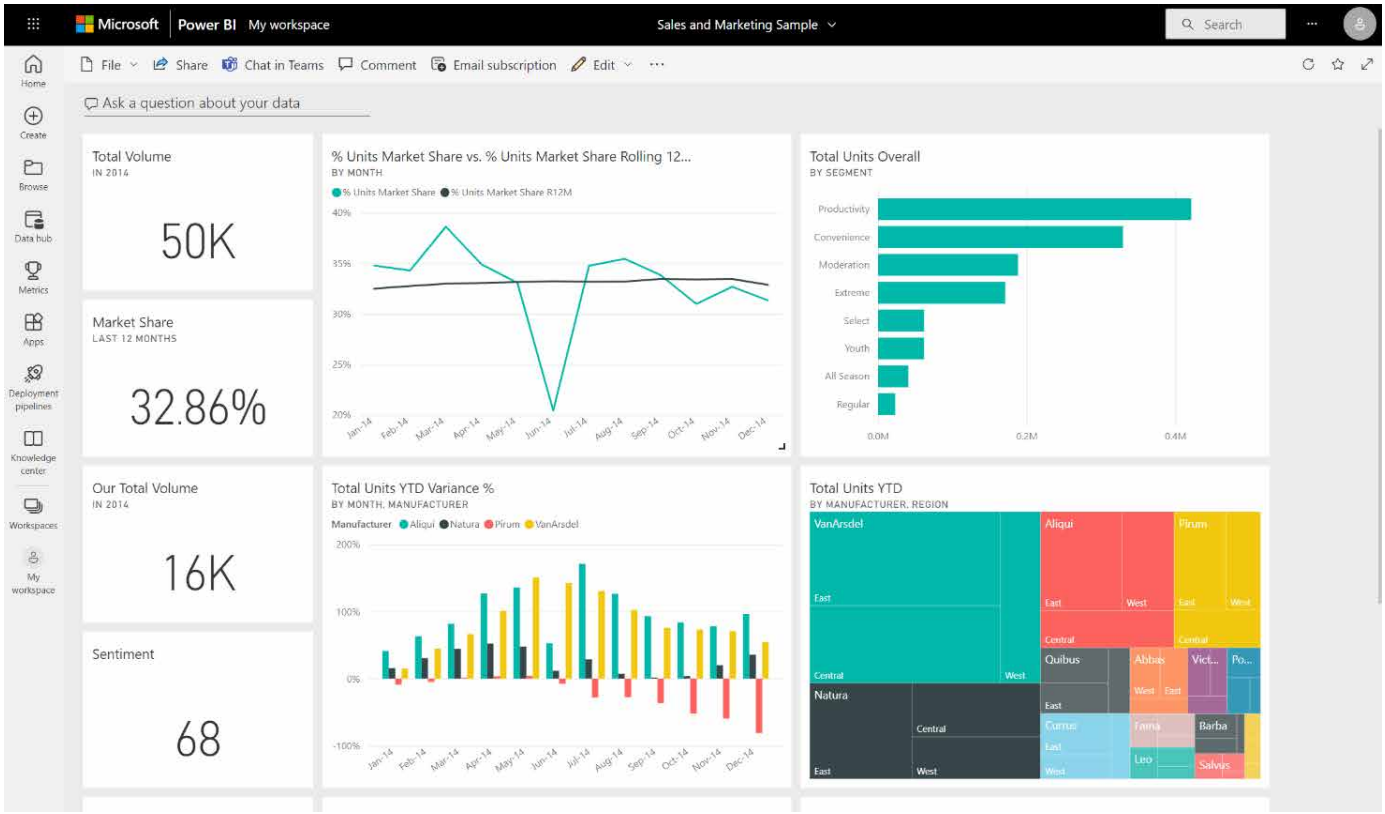
Також за допомогою Power BI можна побудувати інтерактивні графіки та таблиці, які відображають ключові показники ефективності, такі як обсяги виробництва, динаміка витрат, рентабельність продукції. Користувачі можуть легко налаштовувати фільтри для перегляду даних за різними параметрами, наприклад, за регіонами, продуктами або періодами.

Також Power Bi має, хоч і обмежений але доволі простий у використанні функціонал прогнозування. Завдяки цьому в Power BI можна будувати прогнози щодо майбутніх обсягів виробництва, попиту на продукцію. Основними моделями, які можна використовувати є моделі експоненціальне згладжування та SARIMA.

Інтерактивність Power BI є його головною перевагою. Користувачі можуть взаємодіяти із звітами, змінюючи фільтри або параметри аналізу.

Для забезпечення актуальності даних у звітах Power BI підтримує автоматичне оновлення через хмарні сервіси. Це значить, що після кожного оновлення бази даних чи інших джерел інформація у звітах, дані будуть автоматично оновлюватися без втручання користувачів, що полегшить використання звітності іншим користувачам, які не мають прямого доступу до звітності.

На рисунку нижче представлено приклад створеного дашборду за допомогою інструментів Power Bi.



Як висновок можна сказати, що використання функціоналу Power Bi може мати позитивний вплив на трактування результатів дослідження, швидке створення інтерактивних звітів та їх розповсюдження.

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Впровадження системи

Апартні вимоги:

• Процесор: Мінімум 4-ядерний процесор з частотою 2.9 ГГц або вище.

• Оперативна пам'ять: Мінімум 6 ГБ ОЗУ або вище.

• Карта мережевого адаптера: система повинна бути підключена до Інтернету через карту мережевого адаптера.

Програмні вимоги:

- На системі повинна бути встановлена операційна система Windows 7

• На системі повинна бути встновлена версія браузера Chrome 61.0.3163.98 або новіше, або відповідні версії браузерів FireFox, Opera, Microsoft Edge.

Апаратні вимоги до серверної частини:

- Процесор (CPU):
  - Мінімум: 8-ядерний процесор з тактовою частотою 4,0 ГГц.
  - Рекомендація: 16-ядерний процесор з тактовою частотою 2,5 ГГц або вище.
- Оперативна пам'ять (RAM):
  - Мінімум: 32 ГБ оперативної пам'яті.
  - Рекомендація: 64 ГБ оперативної пам'яті або більше.
- Жорсткий диск або твердотілий накопичувач (HDD або SSD):
  - Мінімум: 500 ГБ доступного дискового простору.
  - Рекомендація: 1000 ГБ або більше для забезпечення достатнього місця для зберігання даних.
- Мережевий адаптер:
  - Мінімум: 1 Гбітний Ethernet-адаптер.
  - Рекомендація: 10 Гбітний Ethernet-адаптер для високопродуктивних серверів.

- Резервне копіювання та зберігання даних:
    - Мінімум: Зовнішній жорсткий диск або мережеве забезпечення резервного копіювання для зберігання резервних копій даних.
    - Рекомендація: Рейд-система або мережеві пристрої забезпечення резервного копіювання для забезпечення надійності та доступності резервних копій.

4.2 Впровадження системи

Оскільки дана система передбачає собою веб-орієнтований додаток, то вона не потребує інсталяційного пакету, для користування сайтом, користувачу достатньо відкрити посилання на сайт, тому нижче описані етапи, які потрібно пройти, щоб розгорнути застосунок на сервер. Сервісом для створення серверу було обрано Heroku[23].

Етапи розміщення веб-додатку на сервер:[23]

- Завантажити додаток на GitHub або інший репозиторій контролю версій.
- Створити новий додаток на платформі Heroku та встановити Heroku CLI.
- З локального терміналу, виконати наступні команди для налаштування додатку Heroku, як зображено на рис.52:

```
heroku login
heroku create <app-name>
```

Рис. 4-1 Налаштування додатку

4. Створити файл **requirements.txt** в кореневій директорії проекту і додати необхідні залежності:
5. Створити файл **Procfile** в кореневій директорії проекту і визначити команду запуску сервера:

```
web: gunicorn <your-project-name>.wsgi
```

Рис. 4-2 Команда створення файлу

6. Налаштувати файли конфігурації settings.py та database.py проекту, щоб використовувати базу даних MySQL:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'mydb',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306'
    }
}
```

Рис. 4-3Налаштування серверу

7. Застосувати міграції до бази даних:

```
heroku run python manage.py migrate
```

Рис. 4-4 Команда міграції бд

8. Запустити додаток на Heroku:

```
heroku ps:scale web=1
```

Рис. 4-5 Команда запуску додатку

9. Відкрити ваш Django додаток у веб-браузері за допомогою команди:

```
heroku open
```

Рис. 4-6 Відкриття додатку

4.3 Проведення аналізу виробництва

Так в ході дослідження з допомогою використаних вище технологій було зроблено аналіз виробництва продукції в Україні, було розглянуто її структуру та зміни в ній протягом років. Також було проведено

Для побудови кругового графіку для відображення структури, було проведено аналіз кругового графіку, який відображає структуру продукції. Так було порівняно зміну в структурі довоєнного та післявоєнного періодів.

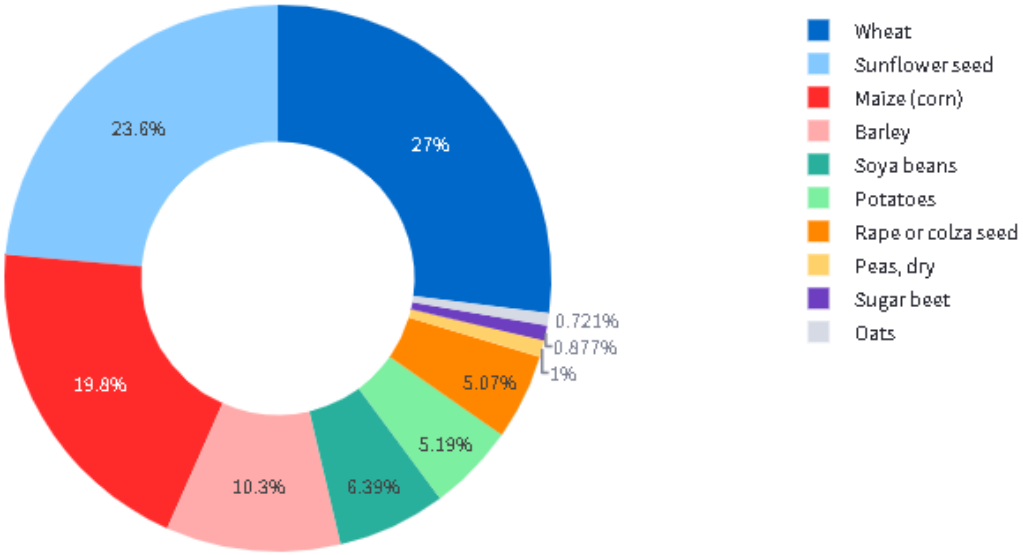


Рис. 4-36 Круговий графік структури продукції за 2021 рік

Як можна побачити з графіку найбільшу питому вагу у виробництві сільськогосподарської продукції займає пшениця 27%, потім йде насіння соняшнику – 23.6% та кукурудза – 19.8%

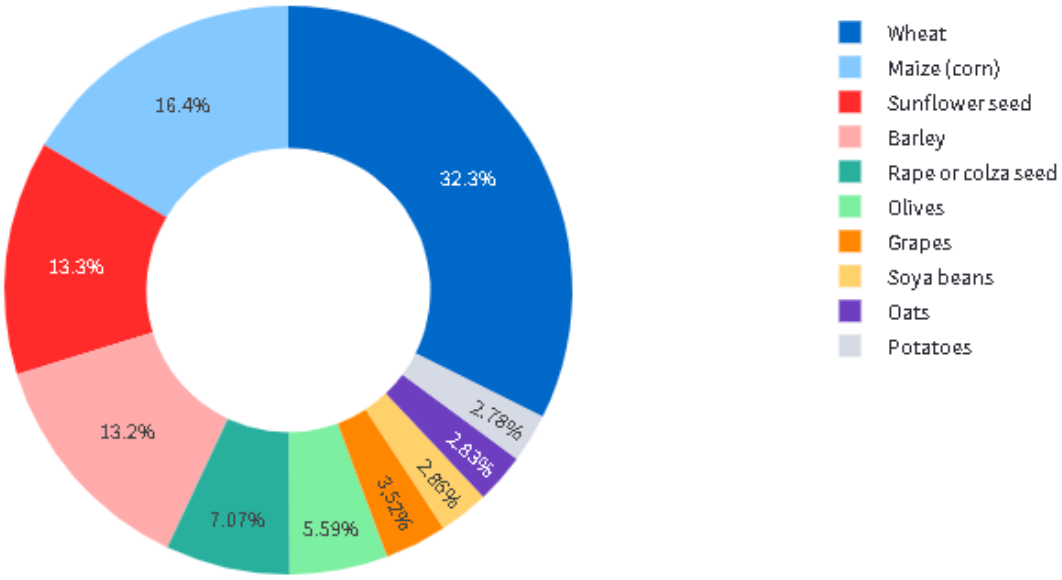


Рис. 4-37 Круговий графік структури продукції за 2023 рік

Для післявоєнного періоду можна побачити зміну в струкутурі продукції так питома вага площі соняшнику зменшилась на 10%, питома вага площі кукурудзи на 3%. Це могло бути спричинене окупацією південних територій де ці культури є одними з найбільш поширеними.

Для оцінки кількості виробленої продукції було розглянуто barplot та відсортовано його за спаданням. Так ми можемо побачити сильне падіння у виробленій продукції для основних культур.

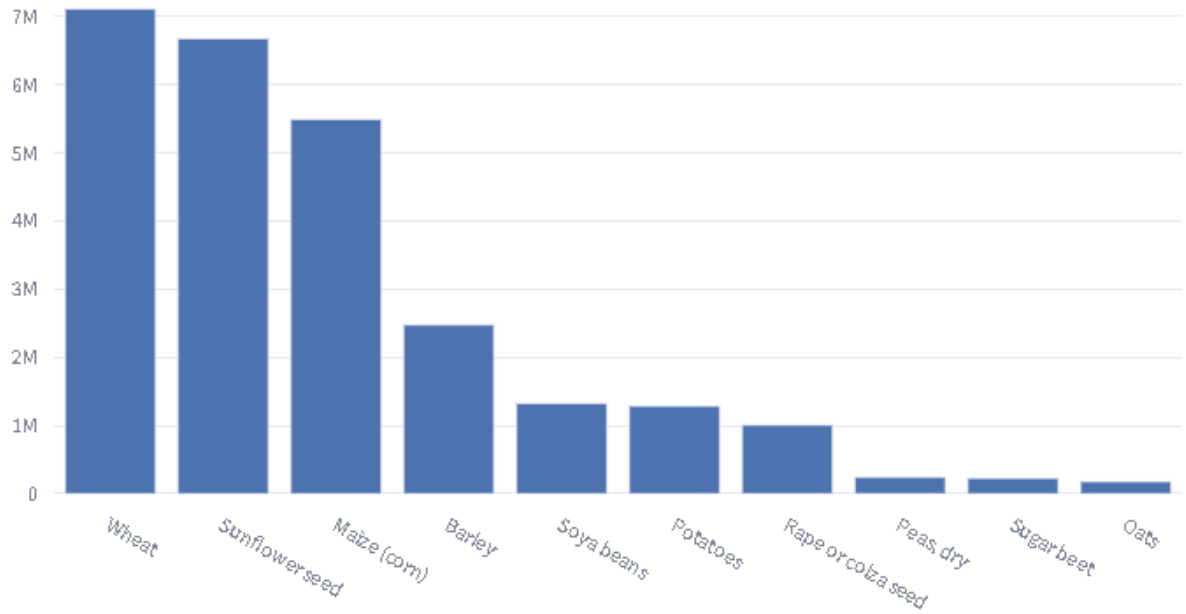


Рис. 4-38 Стовпчиковий графік виробництва продукції за 2021 рік

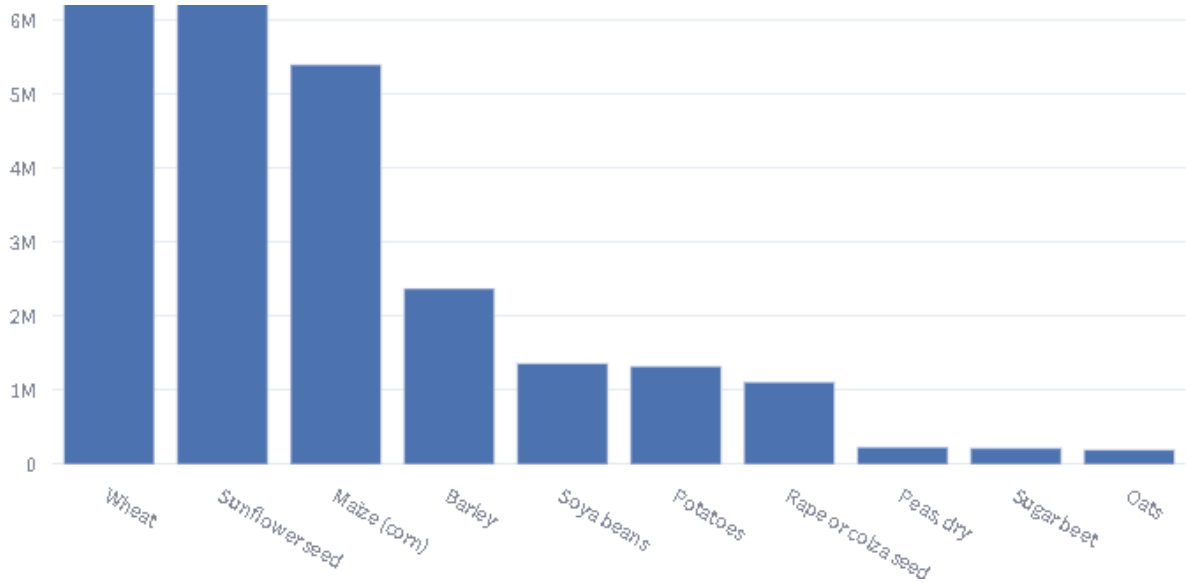


Рис. 4-39 Ствопчиковий графік виробництва продукції за 2023 рік

Порівнявши графіки, можна зробити висновок, що кількість виробленої продукції по всім видам культур впали на 20-25 відсотків в порівнянні з довоєнним періодом, що було спричинене окупацією територій та забрудненістю небезпечними вибуховими предметами не окупованих територій.

Наступним етапом проведення дослідження є аналіз впливу факторів на додану вартість.

Оскільки, кількість виробленої продукції зменшилась на 20-25 відсотків, найвпливовішими факторами стали зменшення посівних площ. Щоб замістити втрачений врожай необхідно зрозуміти, як підвищити продуктивність виробництва. Для цього побудуємо кореляційну матрицю яка міститиме такі фактори: середню кількість внесених органічних та неорганічних добрив, використання ІКТ, використання посадкового матеріалу.

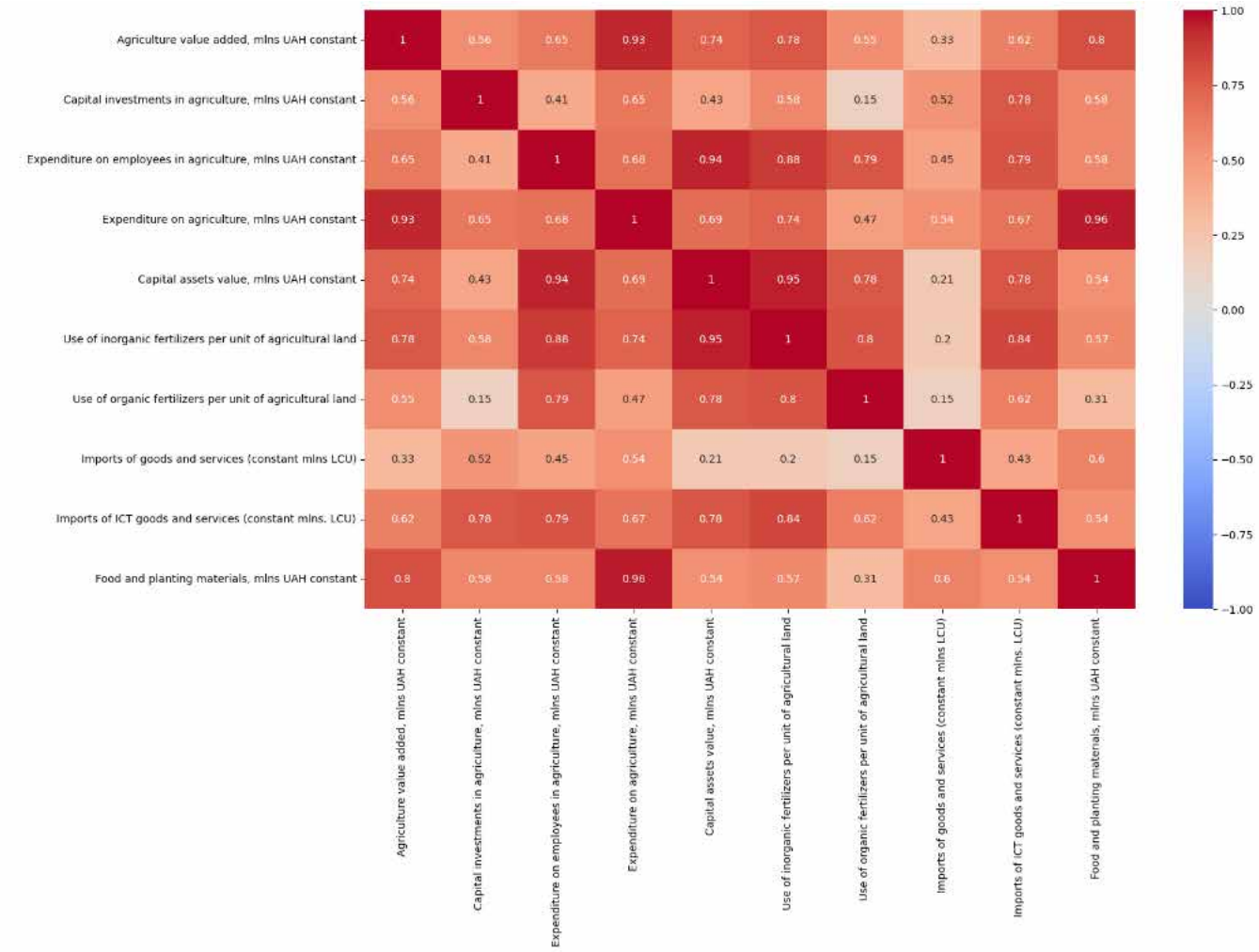


Рис. 4-40 Кореляційна матриця

Згідно з представленою кореляційною матрицею, ми можемо зробити висновки про вплив різних факторів на продуктивність виробництва:

- Витрати на сільське господарство (Expenditure on agriculture) мають дуже високу кореляцію (0.93) з продуктивність виробництва. Це означає, що збільшення витрат на сільське господарство позитивно впливає продуктивність виробництва.

- Витрати на працівників у сільському господарстві (Expenditure on employees in agriculture) також має високий рівень кореляції (0.65), що свідчить про важливість інвестицій в людський ресурс у розвитку сільського господарства.
- Інвестиції в основні засоби (Capital assets value) також значно корелюють із продуктивністю виробництва (0.74), що вказує на важливість капітальних інвестицій для підвищення продуктивності.
- Використання неорганічних добрив на одиницю сільськогосподарської землі (Use of inorganic fertilizers) має високу кореляцію (0.78), що показує позитивний вплив хімічних добрив на продуктивність виробництва .
- Інвестиції в ІКТ товари та послуги (Imports of ICT goods and services) мають кореляцію 0.62, що свідчить про певний позитивний вплив цифрових технологій продуктивність виробництва.
- Харчові продукти та посадковий матеріал (Food and planting materials) мають високу кореляцію (0.8), що вказує на важливість якісних матеріалів для покращення результатів у сільському господарстві.

Загалом, найбільший вплив на продуктивність мають витрати на сільське господарство, використання добрив, інвестиції в основні засоби та харчові продукти і посадковий матеріал. Ці фактори можуть бути розглянуті в першу чергу для подальшої розробки ефективної стратегії розвитку сільськогосподарського сектору.

4.4 Проведення кластерного аналізу

Кластерний аналіз використовується, коли необхідно сегментувати або класифікувати набір даних на групи на основі схожості, але користувач не знає, якими мають бути ці групи.

Так в даному дослідженні можна використати багато поєднань факторів для виділення кластерів, що дасть краще зрозуміти структуру продукції та її схожість. Для формування кластерів було використано дві ознаки - площа територій та витрати на виробництво рослинництва. Це дасть нам виділити основні групи, які мають схожі значення даних ознак, що, наприклад може допомогти у виділенні схожої продукції для подальшого її порівняння, виділення її прибутковості та доцільності її виробництва, що допоможе у збільшенні прибутковості підприємства.

Так по побудованому нижче графіку можна сказати, що оптимальною кількістю кластерів буде величина яка лежить на проміжку 2-5.

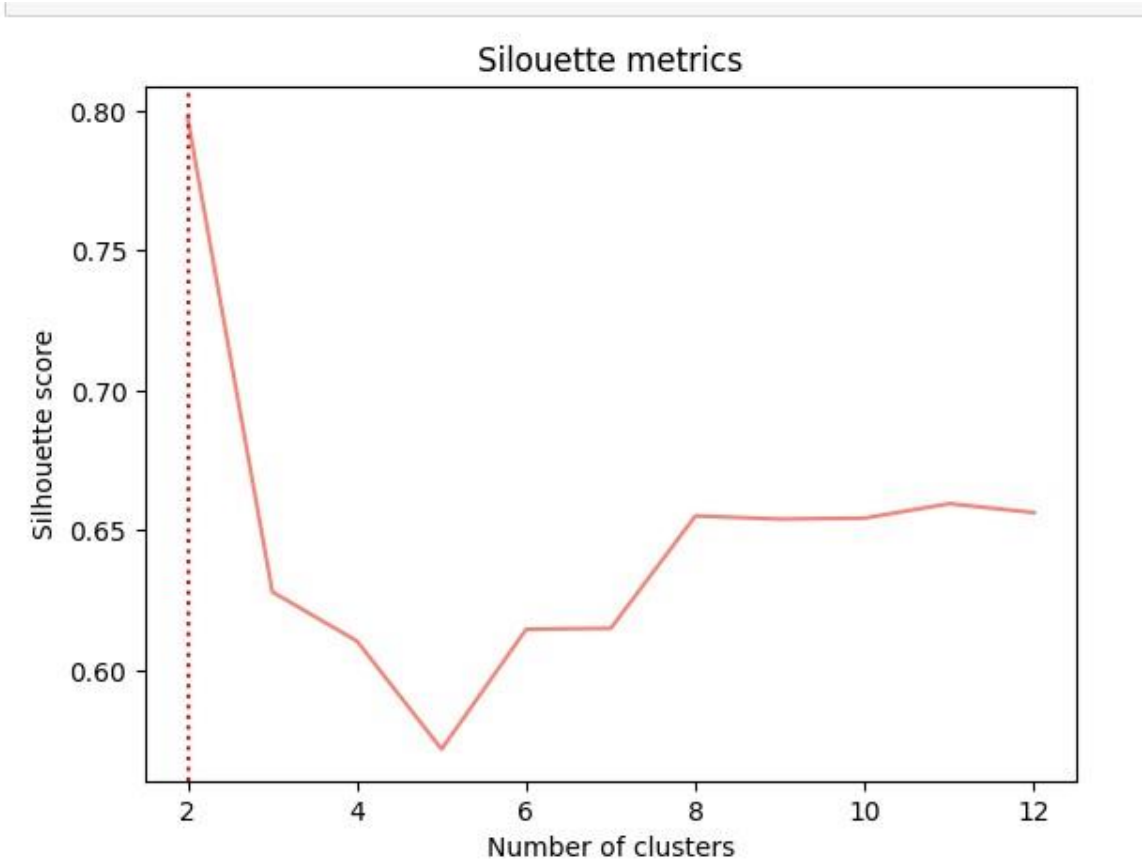


Рис. 4-41 Графік ліктя

В подальшому сформовані кластери можуть бути використані для групування продукції за певним критерієм.

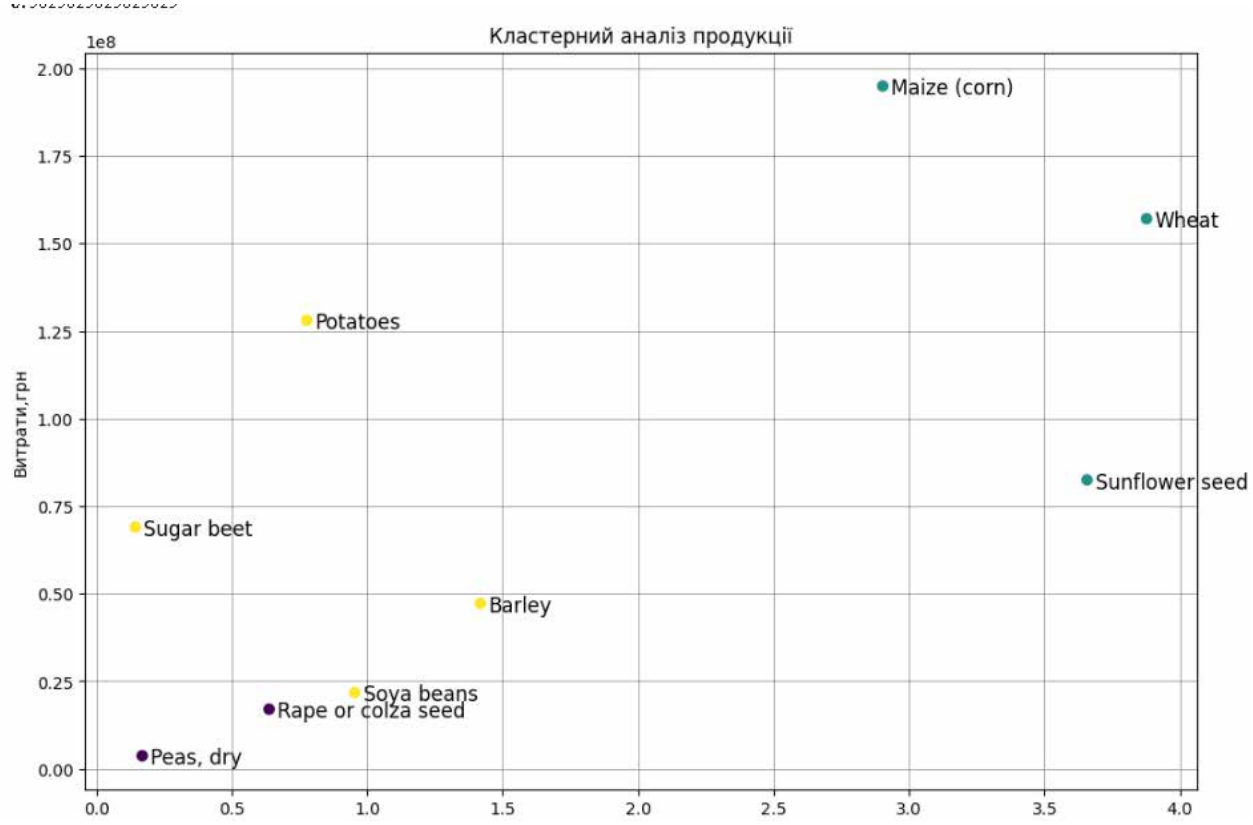


Рис. 4-42 Поділ продукції на кластери

Графік кластерного аналізу демонструє розподіл різних сільськогосподарських культур за площами посівів та витратами у гривнях. На горизонтальній осі відкладено площу посівів у гектарах, а на вертикальній – витрати у гривнях. Такий підхід дозволяє побачити, які культури потребують більше ресурсів, а які – менше.

У верхній правій частині графіка розташовані такі культури, як кукурудза, пшениця та соняшник. Вони мають і великі площі посівів, і значні витрати, що свідчить про їхню високу витрату ресурсів. Ці культури займають значну частину земельних площ та вимагають значних фінансових вкладень, що, ймовірно, пов'язано з їхньою важливістю для сільського господарства чи ринковою вартістю.

Ближче до середини графіка розташовані картопля та буряк. Хоча ці культури не займають надто великих площ, їх вирощування все одно пов’язане з досить високими витратами, порівняно з іншими культурами на подібних

площах. Це може бути результатом специфічних умов для їхнього вирощування або більш дорогих технологій обробки.

У лівій нижній частині графіка можна побачити культури, які потребують як невеликих площ, так і низьких витрат. Це, зокрема, соя, ріпак та сухий горох. Їхні позиції на графіку вказують на те, що ці культури є менш затратними і не потребують значних площ для вирощування. Можливо, їхня агротехніка є менш витратною, або ж вони мають меншу продуктивність, що також впливає на вибір площ для їхнього посіву.

Культури на графіку чітко розділені на три умовні групи: найбільш ресурсомісткі (кукурудза, пшениця, соняшник), середньоресурсомісткі (картопля, буряк) та малозатратні (соя, ріпак, горох).

Побудова класифікаційної моделі з використанням RandomForestClassifier.

Так для проведення даного етапу дослідження було виділено два класи продукції: Продукція яка належить до низького рівня виробітку та до високого.

Дерево рішень класифікує дані на дві категорії: Low (Низький) та High (Високий), використовуючи два основні показники: Area harvested (Площа збору врожаю) та Yield (Врожайність). Спочатку, в кореневому вузлі, перевіряється умова, чи є значення Area harvested меншим або рівним 605900.0. Якщо це так, то зразки класифікуються як Low, і ми продовжуємо перевірку в лівій гілці дерева. Якщо ж значення Area harvested більше 605900.0, зразки класифікуються як High, і ми йдемо по правій гілці дерева.

У лівій гілці дерева перевіряється наступна умова: якщо Yield менший або рівний 282224.0, зразки продовжують рухатися далі по лівій гілці, де переважно зустрічаються зразки класу Low. Якщо значення Yield більше за 282224.0, то зразки класифікуються як High, і переходимо до правої гілки. У лівій гілці, де

Yield менше або рівне 282224.0, є ще одна умова для перевірки значення Yield: якщо воно менше або рівне 253587.0, то всі зразки класифікуються як Low, і дерево досягає листового вузла з чистим результатом (усі зразки цього вузла належать до класу Low). Якщо ж Yield більше 253587.0, зразки продовжують перевірку по правій гілці цього вузла.

У правій гілці від цього вузла перевіряється умова, чи Area harvested менший або рівний 66200.0. Якщо так, зразки класифікуються як Low. Якщо ж значення Area harvested більше 66200.0, зразки класифікуються як High.

У правій гілці кореневого вузла, де Area harvested більше 605900.0, зразки одразу класифікуються як High, оскільки всі значення в цій гілці належать до цього класу.

Таким чином, дерево рішень показує, що більшість зразків належать до класу Low, а лише невелика частина — до класу High. Ключові критерії класифікації — це значення Area harvested та Yield, причому високі значення цих показників зазвичай свідчать про клас High.

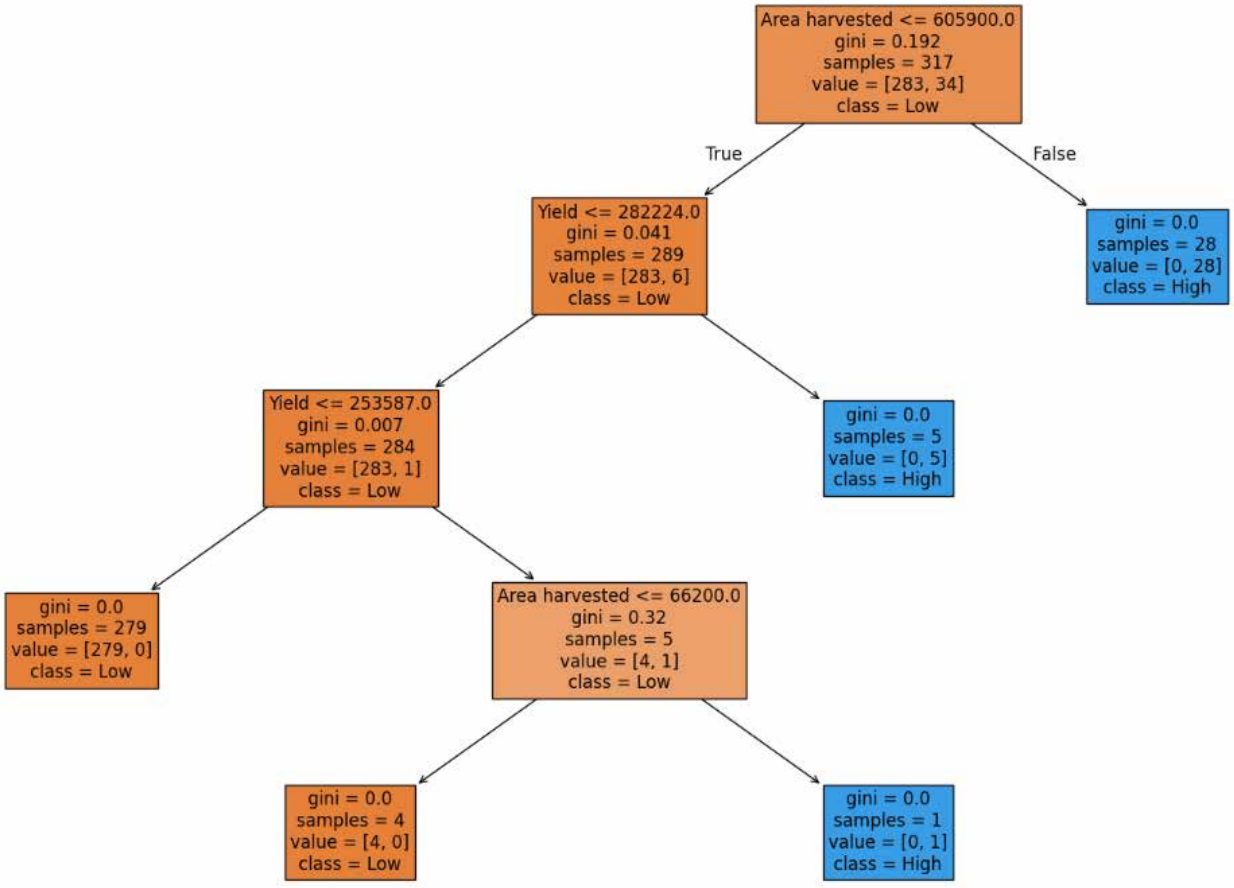


Рис. 4-43 Дерево рішень

Точність побудованої моделі 99.6%.

4.5 Проведення прогнозування

Першим етапом було проведення сезонного декомпозування часового ряду, на основі якого буде проведено прогнозування.

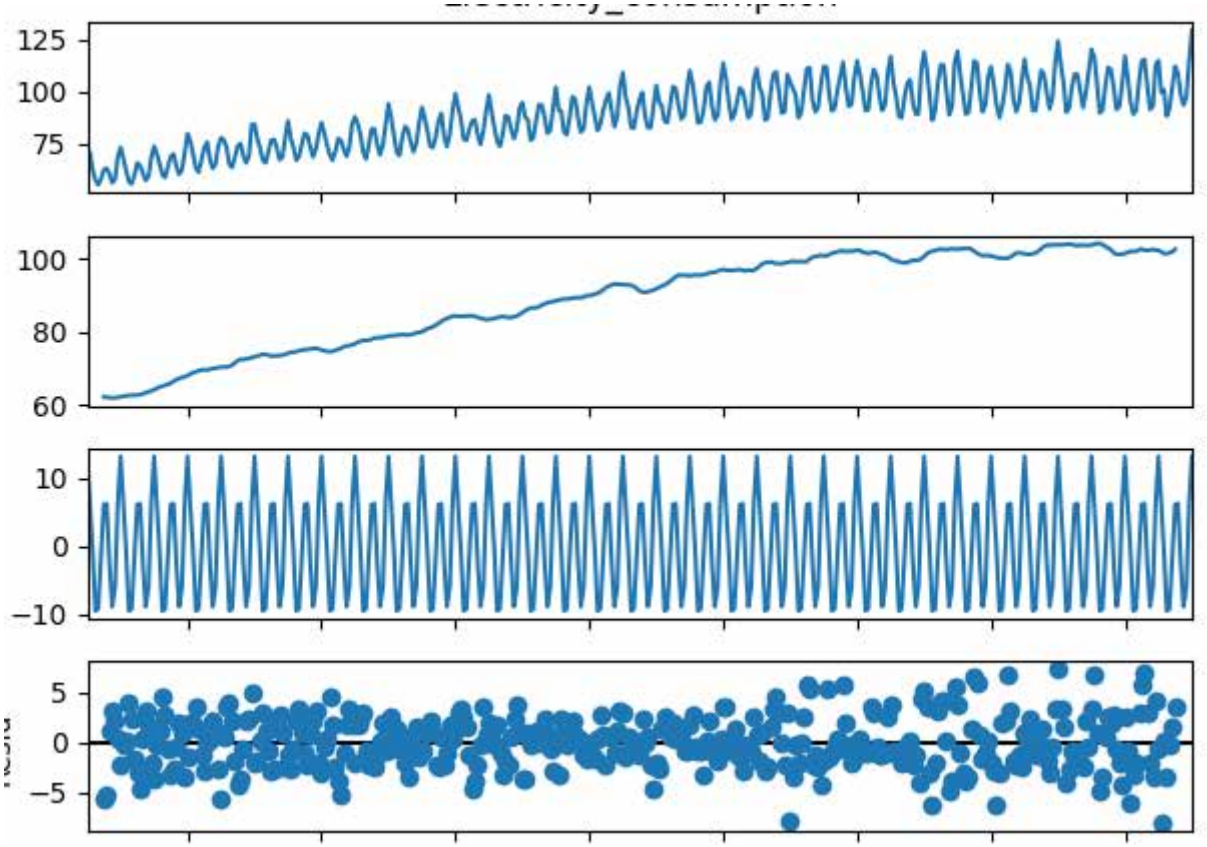


Рис. 4-44 Проведене сезонне декомпозування

Так по побудованому графіку системою можна сказати, що даний часовий ряд має сталий позитивний тренд, та чітко виражено річну сезонність. Дана стабільність часового ряду дасть більшу точність для прогнозу. Після цього кроку Необхідно розуміти чи є автокореляція між попередніми періодами і наскільки глибокою вона є. Для цього система будує графік автокореляції за допомогою функції acf_plot.

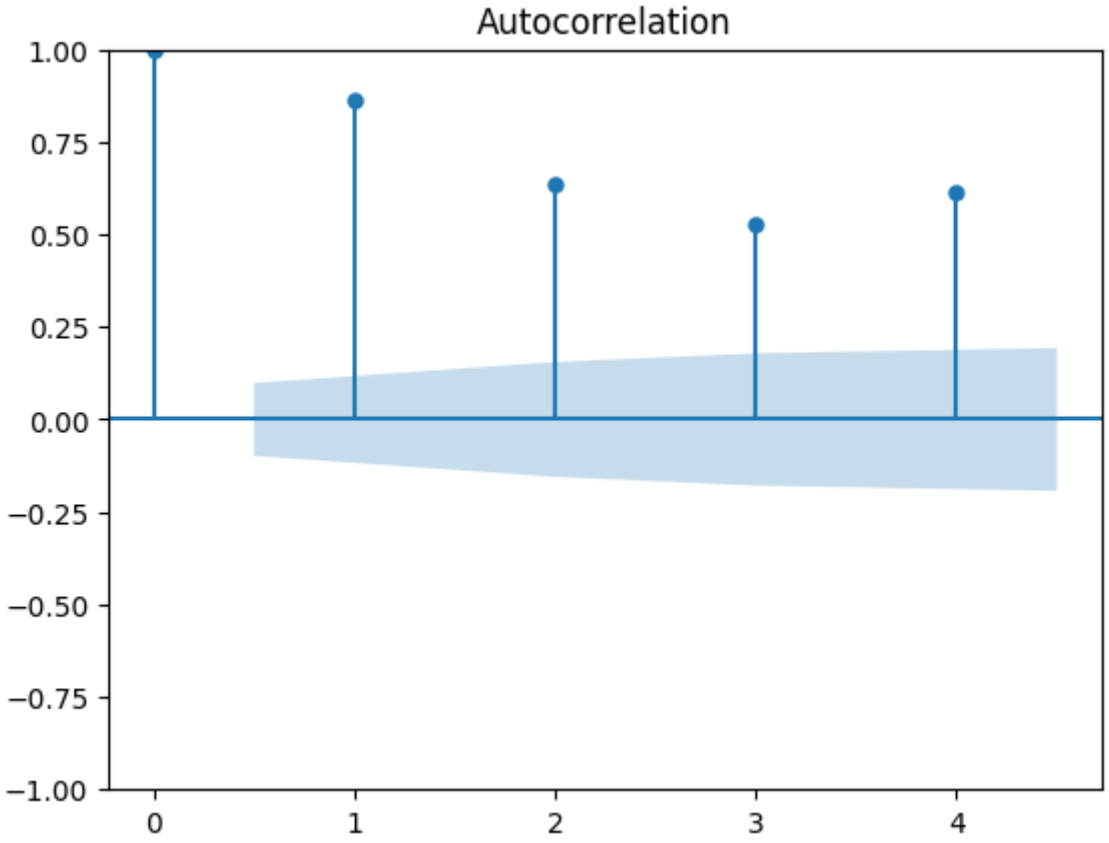


Рис. 4-45 Автокореляція

Розглянувши графік можна сказати, що часовий ряд має доволі сильну автокореляцію і що застосування лагів у прогнозуванні може значно підвищити якість прогнозу це буде досліджено далі.

метод XGB з використанням лагів з кроком 4

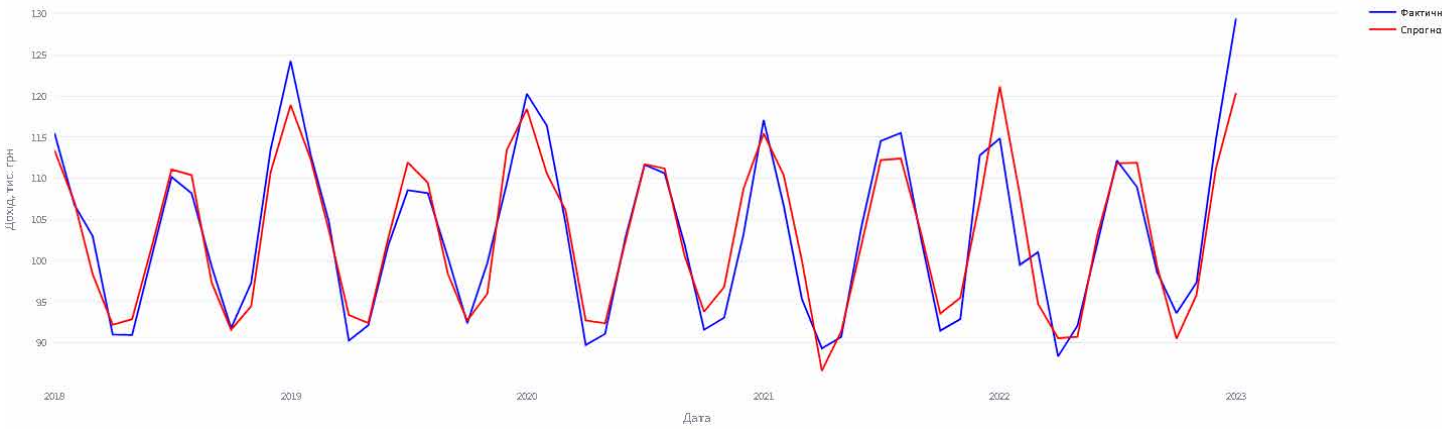


Рис. 4-46 Графік порівняння фактичних та прогнозованих моделлю значень

Рис. 4-47 Метрики оцінки моделі

Розглянувши графік та метрики оцінки можна сказати, що модель є доволі точною та ловить сезонні коливання та тренд. MAPE дорівнює 5 відсоткам, що є дуже гарним показником.

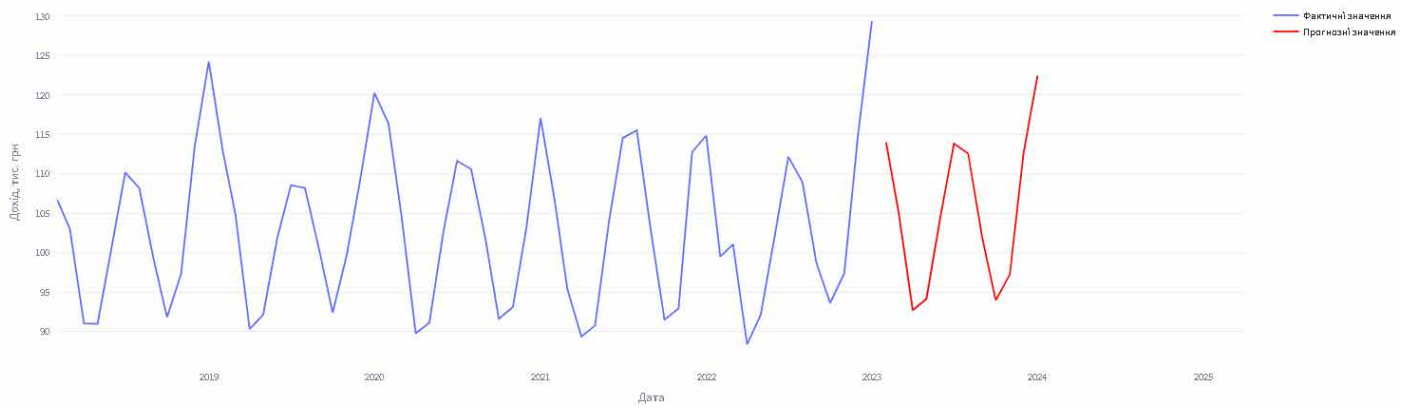


Рис. 4-48 Прогноз

Так побудувавши прогноз на наступні 12 місяців можна зробити висновок, що виробництво продукції не збільшиться, а залишиться майже на тому самому рівні. Тренд відсутній.

Так для побудови даної моделі користувач має ввести сезонність, а параметри p d q система підбере самотужки. Для дослідження було підібрано параметри сезонності -12 p-2,d-0,q-1. За побудованою системо графіками та розрахованими метриками можна зробити висновок, що даний метод є доволі

точним, але все ж таки через свій доволі простий алгоритм, можливі помилки при входженні аномальних значень – викидів.

Рис. 4-49 36 Графік порівняння фактичних та прогнозованих моделлю значень

Рис. 4-50 Метрики оцінки моделі

Роглянувши метрики та показання графіку можна сказати, що хоча відсоток помилки не є високим і складає 6.6, але з графіку видно, що модель не ловить повністю сезонні коливання, що може стати проблемою, тому використання даної моделі не є доцільним.

Наступним досліджуваним методом є метод з використанням глибокого навчання, а саме LTMS. Його архітектуру було описано вище. Так для побудови моделі необхідно вказати кількість циклів навчання, кількість шарів моделі та кількість нейронів і частоту виключення. Оптимальними параметрами були.

Після побудови моделі можна побачити динаміку зміни функції втрати як на тренувальних даних так і тестових, на основі яких системою будується графік, по якому можна визначити чи є модель перенавченою або недонавченою.

```
Epoch [10/100] - Training Loss: 0.0475, Test Loss: 0.0922
Epoch [20/100] - Training Loss: 0.0169, Test Loss: 0.1320
Epoch [30/100] - Training Loss: 0.0122, Test Loss: 0.0961
Epoch [40/100] - Training Loss: 0.0096, Test Loss: 0.0790
Epoch [50/100] - Training Loss: 0.0081, Test Loss: 0.0686
Epoch [60/100] - Training Loss: 0.0061, Test Loss: 0.0634
Epoch [70/100] - Training Loss: 0.0048, Test Loss: 0.0421
Epoch [80/100] - Training Loss: 0.0038, Test Loss: 0.0410
Epoch [90/100] - Training Loss: 0.0037, Test Loss: 0.0340
Epoch [100/100] - Training Loss: 0.0031, Test Loss: 0.0276
```

Рис. 4-51 Train-test loss

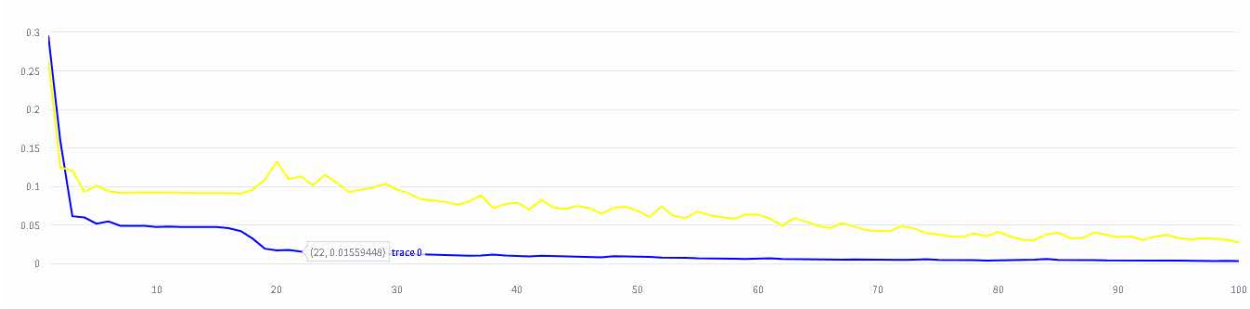


Рис. 4-52 Крива навчання

Рис. 4-53 Метрики оцінки моделі

Так з кривої навчання можна зробити висновок, що модель не є недонавчено або перенавченою.

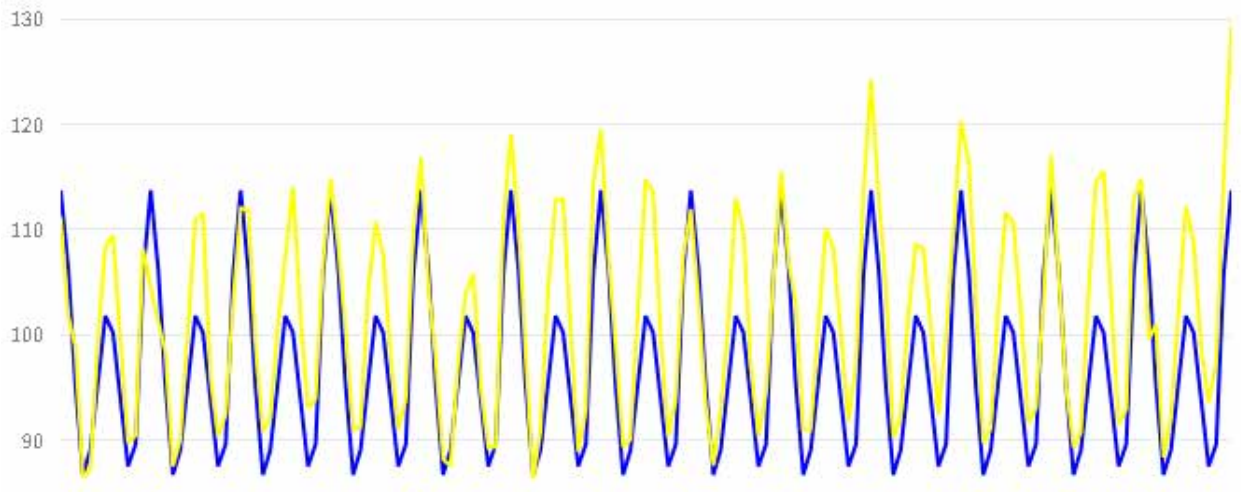


Рис. 4-54 Графік порівняння фактичних та прогнозованих моделлю значень

Так з графіку та оціночних метрик можна сказати, що модель доволі точно підібрала значення і врахувала сезонні та трендові коливання.

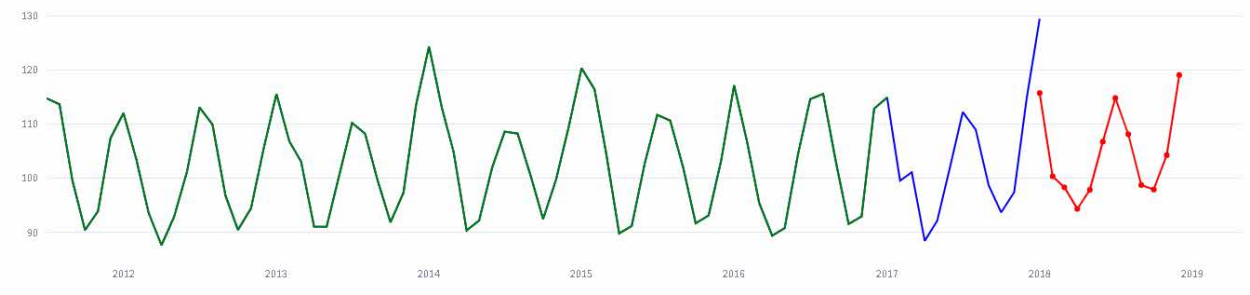


Рис. 4-55 Прогноз

Провівши прогнозування за допомогою даної моделі можна сказати, що виробництво продукції не збільшиться, а залишиться майже на тому самому рівні. Тренд відсутній.

Розробивши моделі та провівши їх тренування, порівнявши метрики оцінки якості моделі, та порівнявши прогнозні дані, можна зробити висновок, що найбільш оптимальною моделлю для подальшого використання є модель з використанням машинного навчання XGB. Модель на основі глибокого навчання також є непоганою альтернативою, якщо прогнозні значення попередньої моделі якимось чином не задовольняють аналітика.

ВИСНОВКИ

В ході виконання магістерського дослідження було досліджено виробництво сільськогосподарської продукції в Україні. Так було зроблено аналіз основних процесів даної предметної області а також здійснено пошук патентів та систем, які реалізують схожий функціонал, що й система. Також було побудовано основні моделі предметної області, які описують взаємодію користувачів з системою.

На етапі більш детального аналізу предметної області вивчено ключові процеси, що впливають на ефективність управління виробництвом, проведено огляд існуючих рішень та визначено вимоги до розробки системи. Постановка завдання дозволила сформувати чіткий план побудови системи, орієнтованої на глибокий аналіз даних та їх подальше використання для стратегічного планування.

На етапі моделювання створено діаграми прецедентів і активності, що відображають функціональні можливості системи та логіку її роботи.

Розробка системи передбачала створення багаторівневої архітектури, розробку структури сховища даних та інтеграцію OLAP-куба для виконання багатовимірного аналізу з використанням KPI. Інструмент SSIS забезпечив автоматизацію процесів імпорту даних для заповнення кубу.

Особливу увагу було приділено застосуванню методів data mining для кластеризації та виявлення закономірностей у даних, побудови дерев рішень для класифікації даних, а також використанню методів прогнозування. Для оцінки якості прогнозів було обрано ключові метрики, що дозволили обґрунтовано оцінити результати моделювання.

Завдяки інтеграції з Power BI було реалізовано сучасний підхід до візуалізації даних і побудови інтерактивних звітів. Це дозволило створити інтуїтивно зрозумілі дашборди.

Результати дослідження підтвердили ефективність впровадженої системи. Проведення кластерного аналізу дозволило сегментувати дані за ключовими

ознаками, а прогнозування забезпечило формування обґрунтованих рекомендацій для прийняття управлінських рішень.

Впровадження розробленої системи дозволить виробництвам автоматизувати аналіз виробничих процесів, підвищити точність прогнозів, покращити оперативність прийняття рішень і забезпечити конкурентоспроможність на ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Держстат - [Електронний ресурс] - Режим доступу - https://www.ukrstat.gov.ua/
2. Fao.org [Електронний ресурс] - Режим доступу - https://www.fao.org/faostat/en/#data/FBS
3. Tni.org [Електронний ресурс] - Режим доступу - https://www.tni.org/uk/article/ukrainian-agriculture-in-wartime
4. visual-paradigm.com [Електронний ресурс] - Режим доступу - https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/
5. Обидєннова, Т., & Васильєв, В. (2023). Цифрові технології в управлінні підприємством: теоретичний аспект. Адаптивне управління: теорія і практика. Серія Економіка, 15(30). https://doi.org/10.33296/2707-0654-15(30)-12
6. visual-paradigm.com [Електронний ресурс] - Режим доступу - https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/
7. TechTarget.com [Електронний ресурс] - Режим доступу - https://www.techtarget.com/searchdatamanagement/definition/OLAP
8. atscale.com [Електронний ресурс] - Режим доступу - https://learn.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver16
9. Investpedia..com [Електронний ресурс] - Режим доступу - https://www.investopedia.com/terms/k/kpi.asp
10. omparitech.com [Електронний ресурс] - Режим доступу - https://www.comparitech.com/net-admin/what-is-microsoft-ssis/
11. turing.com [Електронний ресурс] - Режим доступу - https://www.turing.com/kb/comprehensive-guide-to-lstm-rnn

12. medium.com [Електронний ресурс] - Режим доступу - https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c
13. neptune.ai [Електронний ресурс] - Режим доступу - https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide
14. statistics.com [Електронний ресурс] - Режим доступу - https://www.statistics.com/glossary/seasonal-decomposition/
15. corporatefinanceinstitute.com [Електронний ресурс] - Режим доступу - https://corporatefinanceinstitute.com/resources/data-science/autocorrelation/
16. machinelearningmastery.com [Електронний ресурс] - Режим доступу - https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/
17. jedox.com [Електронний ресурс] - Режим доступу - https://www.jedox.com/en/blog/error-metrics-how-to-evaluate-forecasts/
18. Lopez, Christian, et al. "An unsupervised machine learning method for discovering patient clusters based on genetic signatures." Journal of biomedical informatics 85 (2018): 30-39.
19. ibm.com [Електронний ресурс] - Режим доступу - https://www.ibm.com/topics/gradient-descent
20. ibm.com [Електронний ресурс] - Режим доступу - https://www.ibm.com/topics/random-forest
21. neptune.ai [Електронний ресурс] - Режим доступу - https://neptune.ai/blog/xgboost-everything-you-need-to-know
22. atlassian.com [Електронний ресурс] - Режим доступу - https://www.atlassian.com/data/charts/how-to-choose-data-visualization
23. Кравченко, О. В. Веб-платформа з моніторингу організації проведення студентських олімпіад: дипломний проект … бакалавра : 122 Комп'ютерні науки / Кравченко Олександр Вікторович. – Київ, 2023. – 70 с.