

НУБІП України

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

15.03 – КМР.1862 – “С” 2021.11.01.05 ПЗ
НУБІП України

ЖИГНІСА ДАНИЛА МИКОЛАЙОВИЧА

2022 р.
НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

НУБІП України

Факультет інформаційних технологій

НУБІП України

УДК
«ПОГОДЖЕНО»
Декан факультету

«ДОНУСКАЄТЬСЯ ДО ЗАХИСТУ»
Завідувач кафедри комп'ютерних наук

інформаційних технологій

НУБІП України

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

202_р.

202_р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

НУБІП України

на тему Експертна система на основі маркетингових досліджень
ринку ресторанних послуг

Спеціальність 122 Комп'ютерні науки

НУБІП України

Освітня програма Інформаційно-управляючі системи та технології

(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

НУБІП України

(науковий ступінь та вчене звання)

(підпис)

(ПІБ)

Керівник магістерської кваліфікаційної роботи

К. Т. Н., доцент

(науковий ступінь та вчене звання)

Дудник Алла Олексіївна

(підпис)

(ПІБ)

Виконав

НУБІП України

(підпис)

Жигінас Даниїл Миколайович

(ПІБ студента)

КІВ-2022

ЗМІСТ

ВСТУП	4
1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Постановка завдання	6
1.2 Огляд інформаційних джерел та існуючих рішень	7
2. МОДЕЛЮВАННЯ СИСТЕМИ	11
2.1 Діаграма прецедентів	11
2.2 Діаграма послідовності	13
2.3 Діаграма активності	13
2.4 Структура джерела інформації для проведення інтелектуального аналізу	15
3. РОЗРОБКА СИСТЕМИ	19
3.1 Архітектура системи	19
3.2 Інформаційне забезпечення	20
3.3 Тестування системи	27
4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ	31
4.1 Нечітка експертна система	31
4.2 Дослідження використання задач класифікації	42
4.3 Дослідження використання методу асоціативних правил	52
4.4 Дослідження використання алгоритмів кластеризації	57
ВИСНОВКИ	62
СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ	63
ДОДАТОК А	65
ДОДАТОК Б	69

НУБІП України

ВСТУП

Світ змінюється щохвилини. Те, що раніше нам здавалося неможливим, зараз має дуже великий попит і розвиток. Якщо звернути увагу на ресторанний бізнес то можна побачити, що в наш час, наявність лише закладу, не є достатнім, щоб бути конкурентоздатним в цій сфері бізнесу. Велика кількість закладів харчування зараз мають особисті ІТ рішення або ж підключені до єдиних сервісів які дозволяють користувачам робити замовлення знаходячись поза межами цих закладів. І це лише один критерій який значно покращує прибутковість.

Ще два роки назад, в ті часи які нам здавалися і так непростими, була всесвітня пандемія COVID-19 яка, на деякий, час повністю “заборонила” можливість відвідування закладів харчування. І ті хто вже мав свої інтернет-рішення для прийому замовлень і ті хто швидко зміг зорієнтуватися і створити або підключитися до них, змогли продовжувати вести ресторанний бізнес. А для когось, працювати тільки через доставку стало основною частиною роботи. Тому, власники які вже ввели частинку ІТ світу до свого бізнесу готові і для майбутніх непередбачених ситуацій. І знову, це ще один критерій, але далеко не останній.

Говорячи про особисту систему закладу харчування, можна виділити безліч додаткових корисних моментів які можуть значно покращувати деякі чинники.

Окрім цього, цей спосіб містить ще одну значну перевагу, яка відкриває нові можливості в сфері продажі та розумінні, що саме подобається клієнтам

Система надає можливості збору аналітичних даних на основі взаємодії

користувача з системою, а саме можна отримати такі дані:

- особисту інформацію користувача,
- кількість замовлень певної страви у певний проміжок часу;

• вподобання користувача;
• популярні пошукові запити на сайті,
• діяльність користувача на площадці (придбання, оцінювання та тощо).

Завдяки отриманим даним можна значно покращувати конверсію продажів у будь-якому ресторані. Даний проект може використовуватися як частина більшого проекту, наприклад для розробки повноцінної системи бухгалтерського обліку.

Всі дані які можна отримати з взаємодії користувача з веб-системи, мають бути збереженими та доступними для подальшого аналізу. Всі дані такого плану найкраще всього збирати та каталогізувати з допомогою різноманітних інструментів. Наприклад це може бути база даних побудована на мові SQL.

Об'єкт дослідження: Дані користувачів які взаємодіють із системою
Предмет дослідження: Експертна система на основі маркетингових досліджень ринку ресторанних послуг

Мета роботи: Дослідити та проаналізувати ефективність зібраних даних вподобань користувачів на основі взаємодії з системою замовлень у ресторані.

Структура роботи. Дипломний проект складається зі вступу, чотирьох розділів (аналізу предметної області, моделювання системи, розробки системи аналізу та результати досліджень), висновків, списку використаних джерел та додатків.

Записка складається з 70-ти сторінок. У роботі використано 9 джерел.

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Постановка завдання

Основною ціллю даної магістерської роботи є розроблення системи аналізу, яка має накопичувати дані веб-системи по продажу страв з ціллю подальшого аналізу для виявлення вподобань користувача. Тому що, на відміну від локальних закладів, система більш автоматична та здатна збирати усі необхідні дані які в майбутньому можуть покращувати взаємодію користувачів із системою, дати можливість проводити певні маркетингові дослідження і на їх висновках робити пропозиції, припущення або прогнози для користувачів. А це, може призвести для покращення конверсії, збільшення кількості клієнтів, розширення мережі закладів та збільшення прибутку, тощо.

Дані мають описувати такі аспекти системи

- інформація про страви:
 - вартість;
 - вага;
 - рейтинг страви;
- інша службова інформація;
- пошукові запити користувачів;

інформація про замовлення:

- дата та час о загальна вартість замовлення о товари в замовленні
- інформація про користувача:
 - вік;
 - країна проживання; о стать,
 - дані про оцінювання страви з боку користувачів.

В рамках цієї системи розглядається варіант з оцінкою страви по шкалі від 0 до 5 балів.

Має бути налаштований процес збору даних в операційну базу даних та подальшою передаючою цих даних в сховище даних, для подальшого аналізу за допомогою технології OLAP.

Проведення аналізу має давати відповідь на наступні питання:

Яка кількість замовлень на акційні пропозиції?

Які пошукові запити найпопулярніші?

Чи збільшується кількість замовлень на певну страву коли вона по знижці?

Чи роблять клієнти більше замовлень коли придбання страви надає додаткові бали до програми лояльності?

Відповідно аналіз має проводитися за певний часовий проміжок, який в рамках системи буде становить термін в 4 місяці.

1.2 Огляд інформаційних джерел та існуючих рішень

Кожен проєкт унікальний. Навіть незважаючи на те, що він може належати одній сфері діяльності та мати одну тематику, проєкт все одно чимось відрізнятиметься. Але для покращення розуміння ведення проєкту потрібно знайомитися з його референціями, щоб розуміти сильні та слабкі сторони конкурентів та правильно застосувати це у своєму проєкті.

Виходячи з цього, у сфері ресторанного бізнесу можна виділити три типи проєктів, які безпосередньо допомагають ресторанному бізнесу здійснювати продажі в інтернеті.

Перший з них сервіси типу кур'єрських доставок, такі як Raketa, Glovo, Bolt Food та інші.

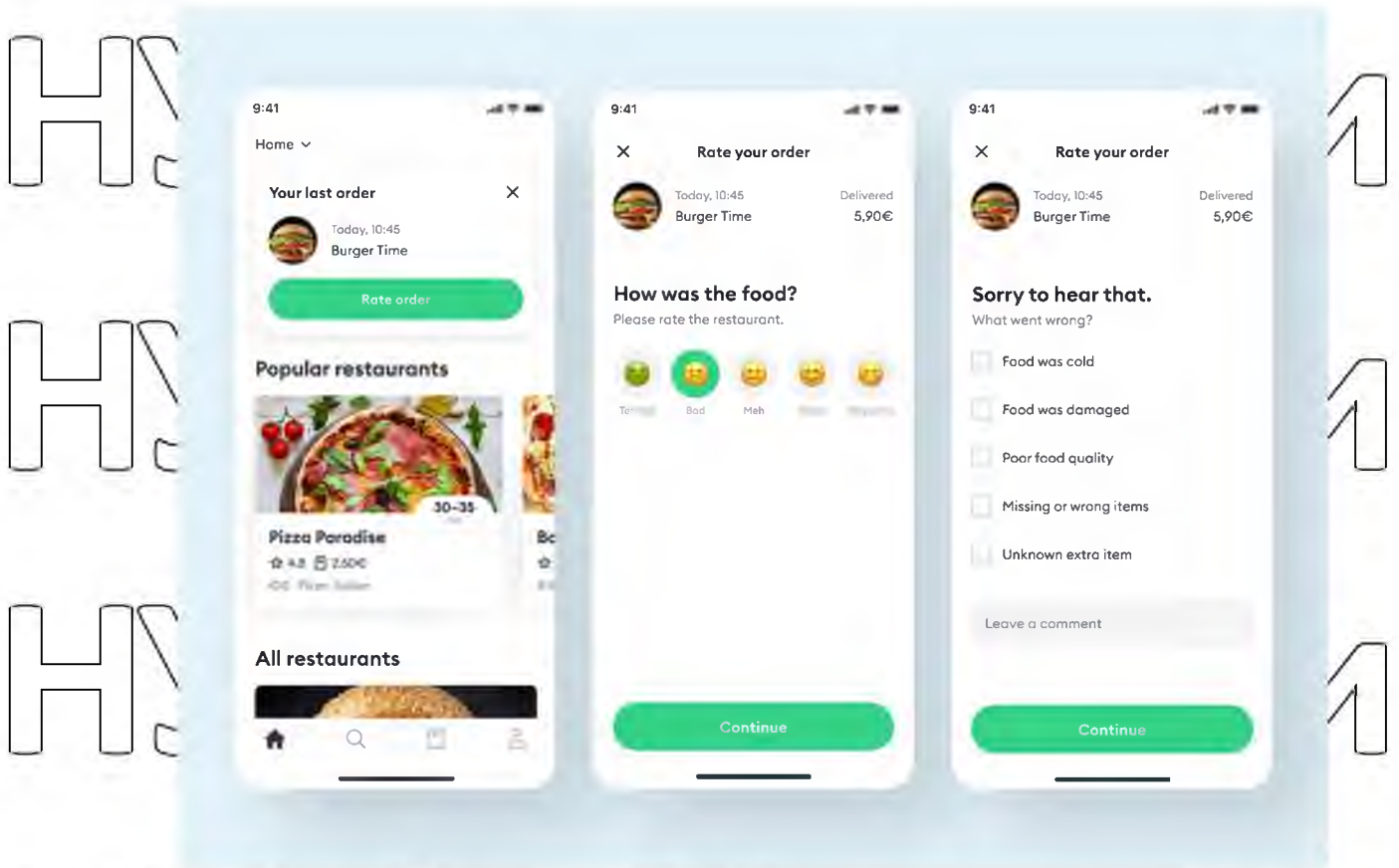


Рис. 1 Приклад сервісу доставки Bolt Food

Такі сервіси мають вигляд додатку, куди різні заклади харчування додають свої позиції з меню, а користувач може з ними ознайомлюватися і замовляти.

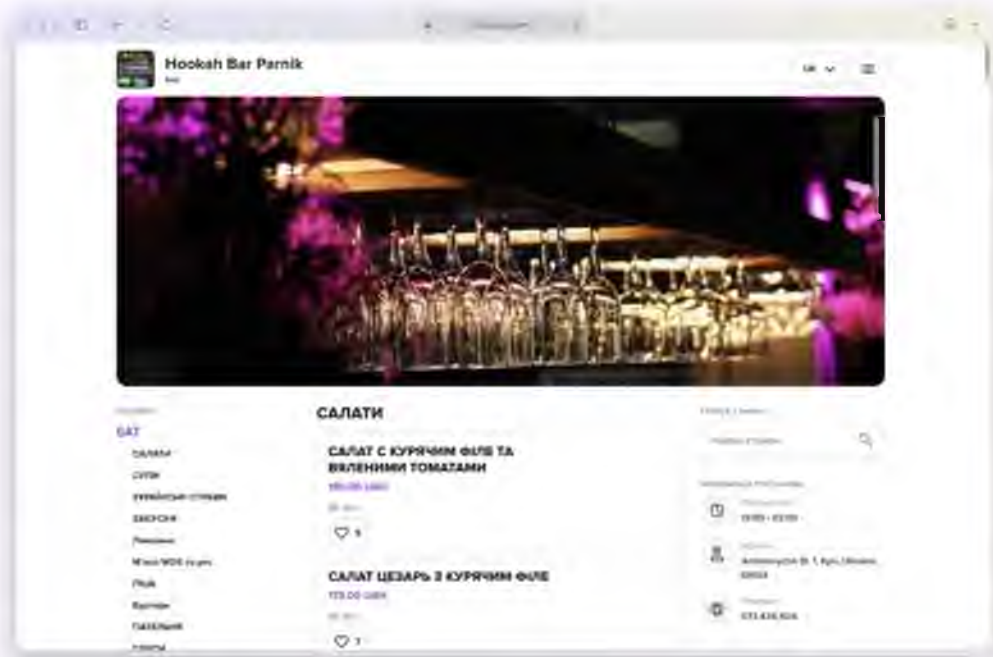
Такий проект простий в обслуговуванні, але він не надає аналітичних даних і не завжди є можливість укласти договір з цими кур'єрськими службами доставки.

Другий тип скоріше інформаційний. Є загальні системи, наприклад ChoiceQR, які дозволяють закладам харчування розміщувати у себе меню і в деяких випадках замовляти та оплачувати страви.

Н

Н

Н



И

И

И

Рис. 2 Приклад сервісу ChoiceQR

НУБІП України

Але як і попередній тип, у цьому немає збору аналітичної інформації. Цей вид проекту радше підходить під невеликі заклади. С технічної сторони, він

виглядає як окремий сайт під кожен заклад харчування, що виключає

необхідність створення звичайних сайт-візиток тощо.

НУБІП України

І третій тип це власна система, яка будується під кожен ресторан, кафе, бари тощо з урахуванням усіх потреб, які там потрібні. Цей тип є найбільш

універсальним оскільки крім простого відображення меню і простих доставок

можна будувати різні маркетингової діяльності та збирати аналітичні дані які

надали можуть підвищувати продажі, впізнаваність. Також можна влаштувати різні взаємодії з користувачами, щоб покращити взаємодію і так далі.

Для магістерської роботи розглядається саме третій тип таких проектів. Це

НУБІП України

не щось інноваційне, тому такі системи не потребують використання складних мов програмування. Як правило, за основу береться створення веб-системи і подальшої її реалізації у вигляді Progressive Web Application для

мобільних пристроїв.

Але також варто відзначити, що виходячи з вище перелічених типів проектів можна сказати, що третій тип є найбільш складним і дорогим, оскільки окрім створення проекту, який буде розроблятися під конкретний заклад, потрібно мати підтримку у вигляді системних адміністраторів, які

будуть займатися наповненням веб-системи, програмістів здатних вирішувати поставлені задачі в виді доопрацювання, поліпшення інтерфейсу та функціоналу і вирішення різних технічних помилок у випадках їх виникнення,

а також маркетологів, які можуть виходячи з аналітичних даних пропонувати

нові ідеї. Також, в окремих випадках можуть знадобитися UI/UX та графічні дизайнери.

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

2. МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Діаграма прецедентів

Важливим етапом роботи над системою є її попереднє дослідження та аналіз з метою виявлення її особливостей, задач, обмежень, сферу застосування та тощо. Одним з найбільш кращих способів проведення цієї роботи є моделювання системи.

Для моделювання системи було обрано використання об'єктно орієнтованого підходу до моделювання системи, а якщо точніше побудова моделі за схемою діаграми прецедентів.

Цей спосіб передбачає створення схеми, яка буде включати UML діаграму. Уніфікована мова моделювання (Unified Modeling Language, UML) є графічною мовою для візуалізації, специфікації, конструювання та документування систем, в яких велика роль належить програмному забезпеченню [1].

За допомогою UML можна детально описати систему, починаючи розробку з концептуальної моделі з її бізнес-функціями і процесами, а також описати особливості реалізації системи, такі як класи програмного забезпечення системи, схему бази даних. Використовуючи UML, також можна розробляти складні системи швидко і якісно.

На рис. 3 представлена діаграма прецедентів, що дозволяє ознайомитись з головними акторами в предметній області системи аналізу, а саме:

Клієнт - будь-яка особа, що може:

- переглянути асортимент товарів, що надає магазин;
- замовити потрібні їй товари;
- за необхідності, отримати консультацію від працівника закладу харчування.

Адміністратор - співробітник ресторану, на якого покладена велика кількість обов'язків, а саме:

- додавання, оновлення та видалення даних про страви у веб-системі;

- створення та заповнення звітів, щодо різних напрямків діяльності закладу харчування;
- обробка замовлень, що надходять із веб-системи;
- регулярне поповнення страв на основі меню закладу харчування.

Офіціант – співробітник ресторану, який може допомагати адміністратору, а саме:

- створення замовлень у системи що надходять із ресторану;
- обробка замовлень, що надходять із веб-системи;

Кухар – співробітник ресторану, який виконує головну роль у виконанні замовлень, а саме:

- передача замовлень клієнтам;
- обробка та виконання замовлення;

Аналітик – співробітник, що займається однією із складових діяльності

будь-якого підприємства – аналіз інформаційних потоків, виконуючи такі функції:

- аналіз попиту споживачів на страву ресторану, базуючись на даних про замовлення;
- аналіз пошукових запитів у конкретний проміжок часу;
- створення та ведення звітів на основі досліджень;

Перевагою використання цього способу моделювання системи є те, що цей

підхід дозволяє розкрити групи користувачів системи, їх вимоги до функціоналу та чітко задати предметну область системи та виокремити головні процеси в ній, які мають бути реалізовані для задоволення потреб користувачів.

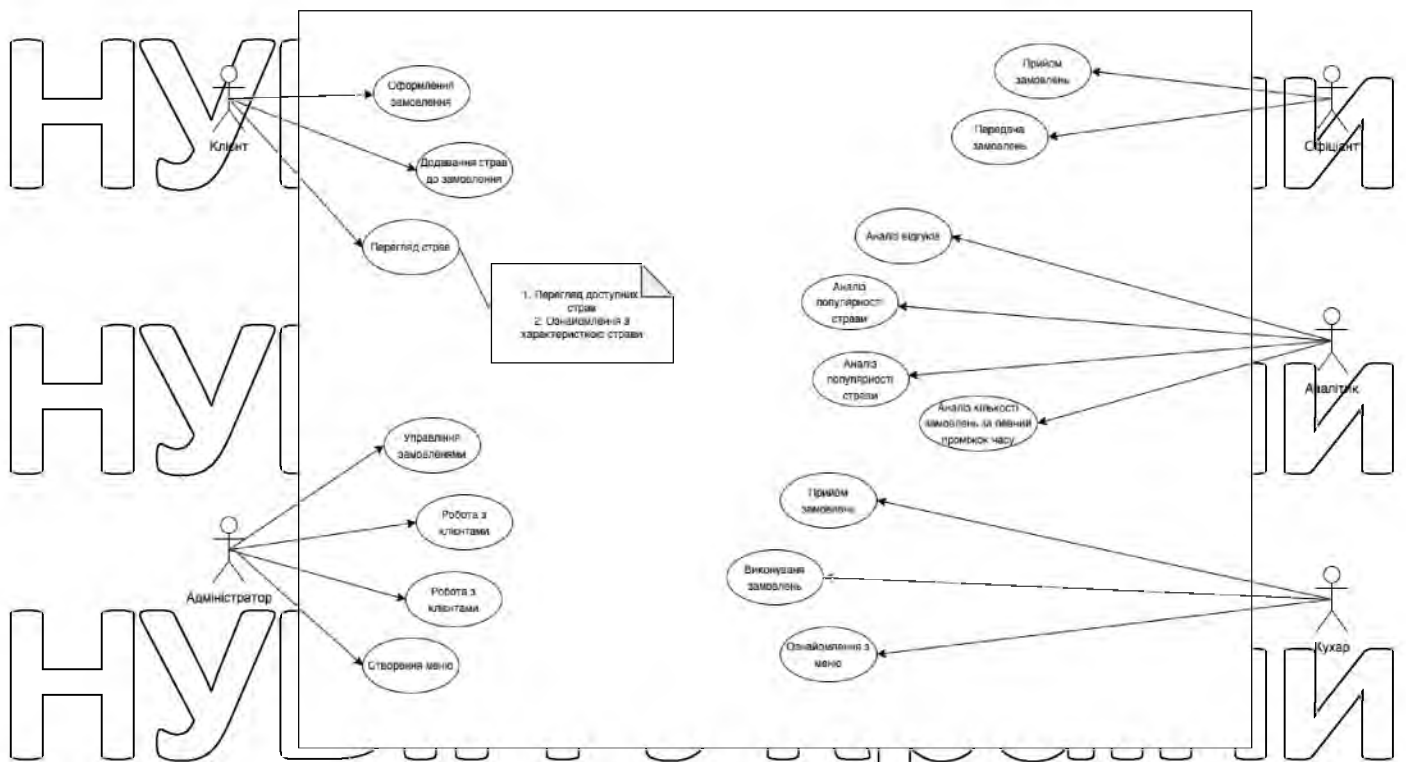


Рис. 3 Діаграма прецедентів

2.2 Діаграма послідовності

Зазвичай діаграма послідовності використовується для відображення реалізації варіантів використання бізнес-процесу, тобто відображення взаємодії об'єктів, відповідної частини або всього варіанту використання.

Взаємодія об'єктів, яке втілює варіант використання, ілюструється однією або декількома діаграмами послідовності.

Діаграми послідовності, важливі при проектуванні, тому що вони допомагають усвідомити ролі об'єктів в потоці Т, отже, надають початкові дані для визначення інтерфейсів і призначень класів. Діаграму послідовності прецедента «Оформлення замовлення» для експертної системи на основі маркетингових досліджень ринку ресторанних послуг зображено в додатку А на сторінці 1.

2.3 Діаграма активності

Діаграма діяльності — в UML та SysML, візуальне представлення графу діяльності. Граф діяльності є різновидом графу станів скінченного автомату,

вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Діаграма діяльності до контрольного класу "Оформлення замовлення" зображена в додатку А на сторінці 2.

Передумови.

Клієнт повинен обрати страви для замовлення.

Головний потік.

Прецедент починає виконуватися, коли клієнт потрапляє на сторінку оформлення замовлення. Система перевіряє наявність страв у замовленні (E-1) і

виводить можливі варіанти дій оформлення: замовити у ресторані і замовити

доставку. Якщо вибрана операція замовити у ресторані, S-1: виконується потік

перехід до оплати. Якщо вибрана операція замовити доставку, S-2: виконується потік реєстрації клієнта (S-3)

Під-потоки.

S-1: Замовити у ресторані. Система відображає діалогове вікно, що містить поле, в якому клієнт повинен ввести номер столику клієнта і заповнити всі обов'язкові поля. Після чого йде перехід до оплати. Потім прецедент починається спочатку.

S-2: Замовити доставку. Система відображає діалогове вікно, яке містить форму для оформлення, в якій клієнт повинен вказати адресу та приблизний час коли він хотів би отримати доставку, а також заповнити всі обов'язкові поля. Якщо клієнт новий, то система зберігає про нього дані.

S-2-1: Авторизація клієнта. Система відображає діалогове вікно, де клієнт може ввести своє ім'я та пароль. (E-2)

Альтернативні потоки

E-1: Ні одна страв не обрана. Користувач повинен обрати та додати товари до замовлення.

E-2: введено неправильне ім'я або пароль. Користувач повинен повторити введення або завершити прецедент

2.4 Структура джерела інформації для проведення інтелектуального аналізу

Для роботи з Analysis Services скористаємось середовищем Visual Studio.

Створюємо новий проєкт та підключаємо джерело даних – сховище даних, як показано на рисунку 4.

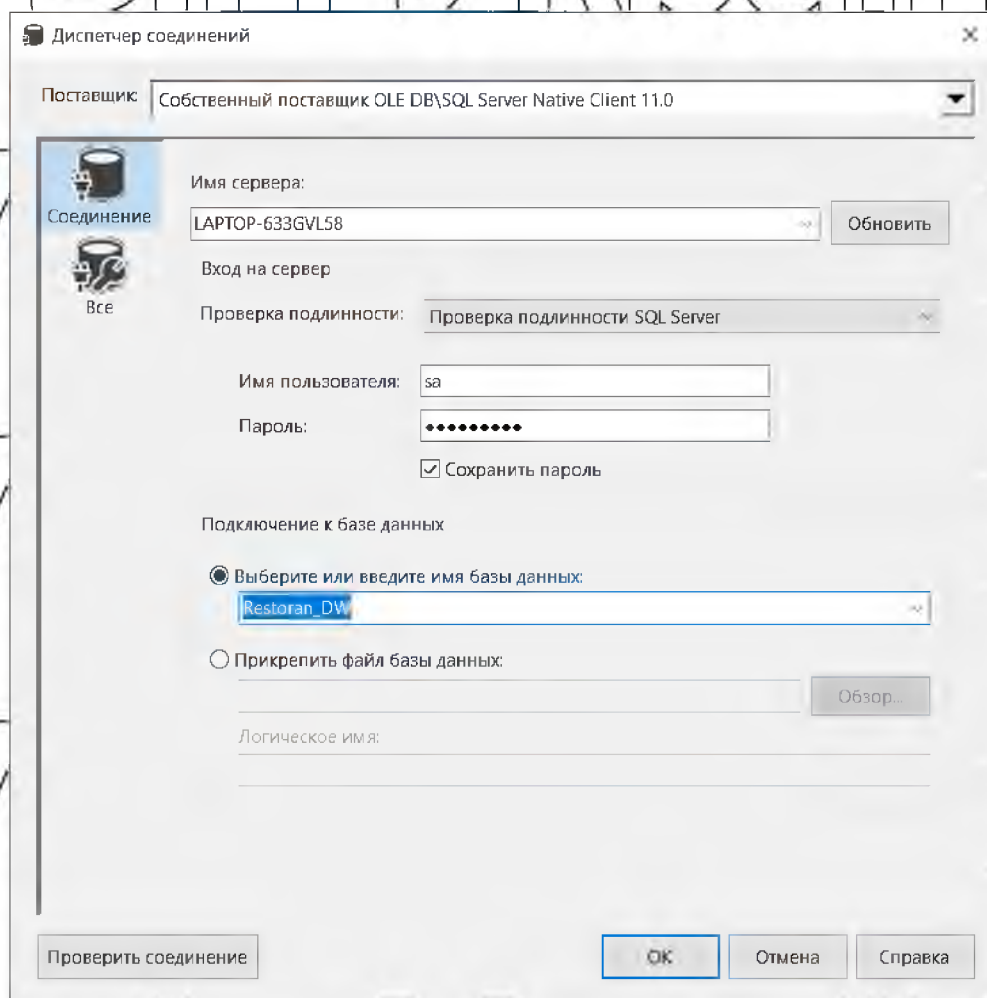


Рис. 4 Додавання джерел даних

Наступним кроком в розгортанні куба є створення уявлення джерела даних. Структура уявлення представлена на рисунку 5.

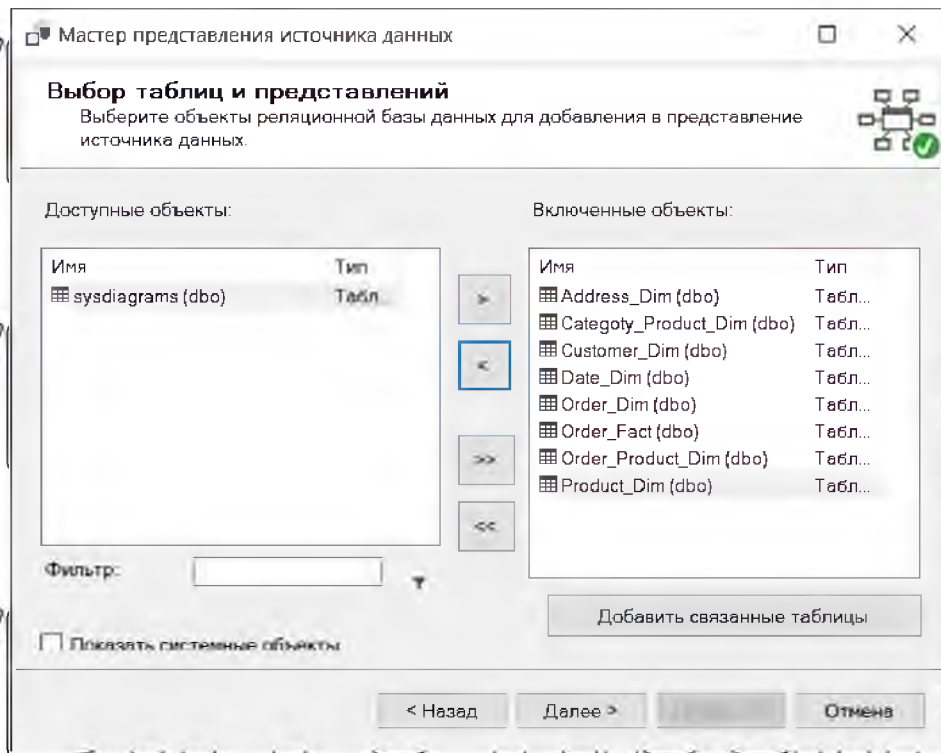


Рис. 5 Уявлення джерела даних.

Важливим аспектом розроблення є правильно створені виміри. На рисунку 6 представлено перший вимір – Date_Dim, що містить дату четверті. Він дає можливість проводити аналіз у розрізі різних часових періодів.

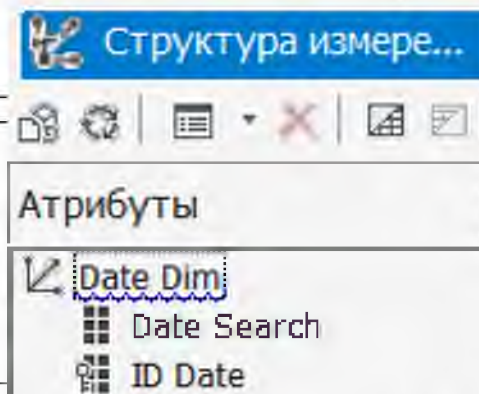


Рис. 6 Вимір Date_Dim.

Наступним виміром створюємо Product_Dim. Структура виміру представлено на рисунку 7

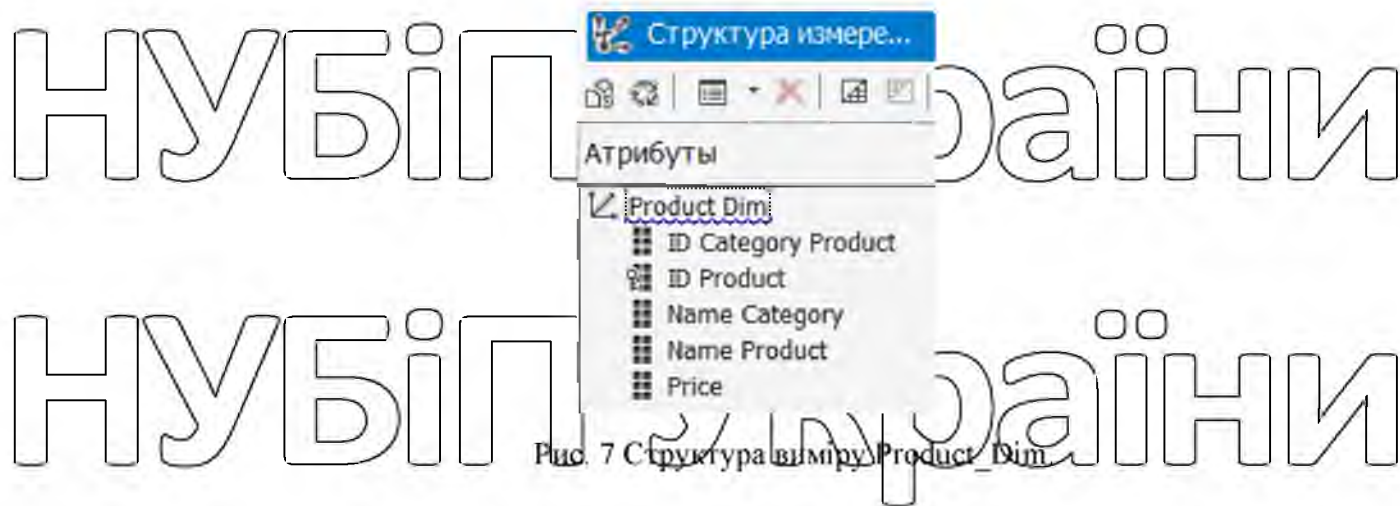


Рис. 7 Структура виміру Product_Dim

На рисунку 8 представлено структуру Order_Dim.

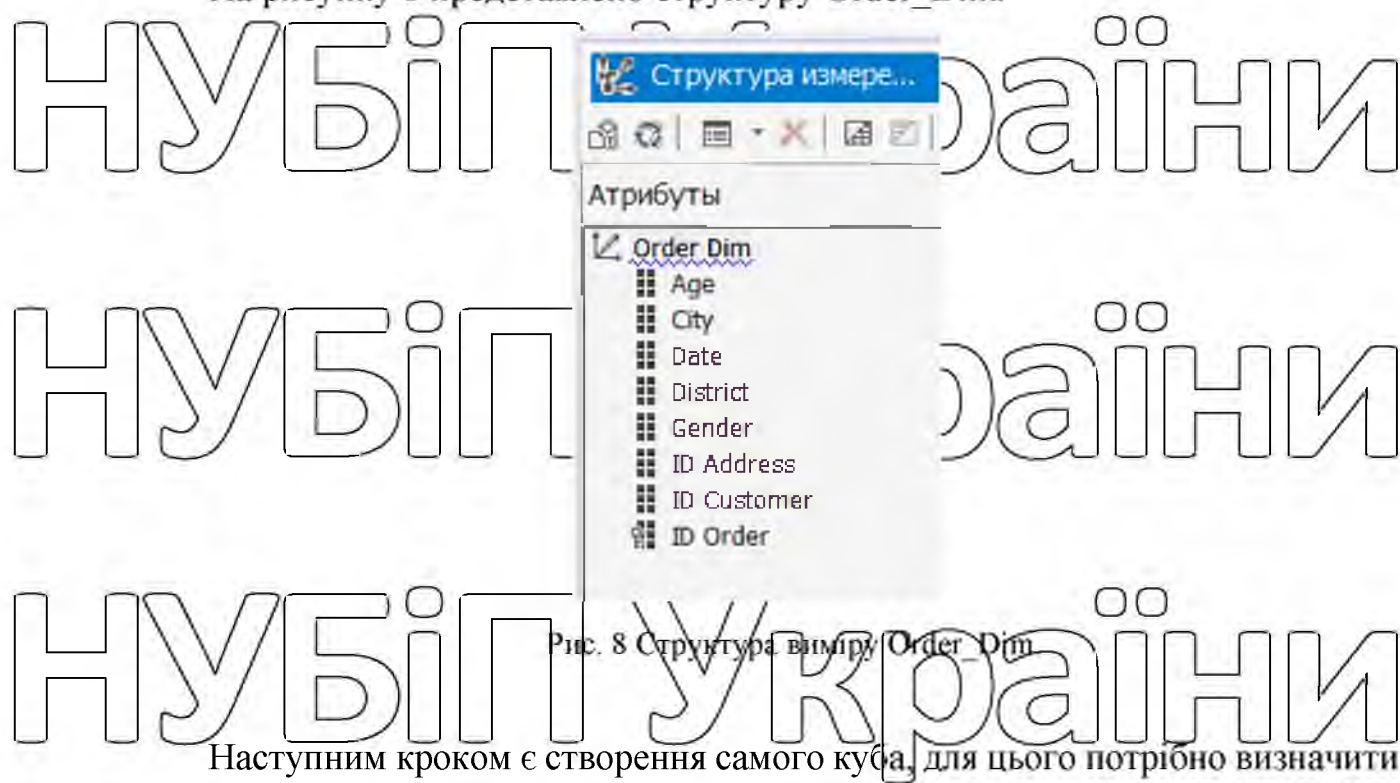


Рис. 8 Структура виміру Order_Dim

Наступним кроком є створення самого куба, для цього потрібно визначити міри, як представлено на рисунку 9

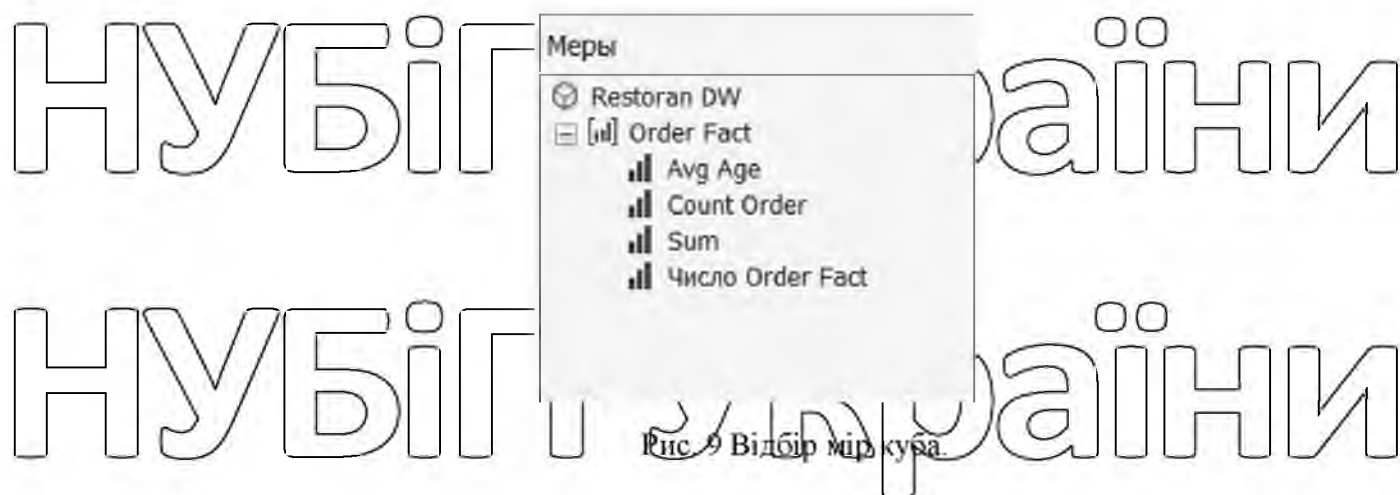


Рис. 9 Вибір мір куба

В результаті підключення всіх компонентів – вимірів та мір, що представлено на рисунку 10 можна розгорнути куб.

Объекты куба	Тип объекта
Restoran DW	Name DefaultMeasure
- Группы мер	
- Order Fact	MeasureGroup
Avg Age	Measure
Count Order	Measure
Sum	Measure
Число Order Fact	Measure
- Измерения	
+ Date Dim	CubeDimension
+ Order Dim	CubeDimension
+ Product Dim	CubeDimension

Рис. 10 Структура куба.

В результаті успішного обробки та розгортання куба було згенерована багатомірна структура куба, що представлено на рисунку 11

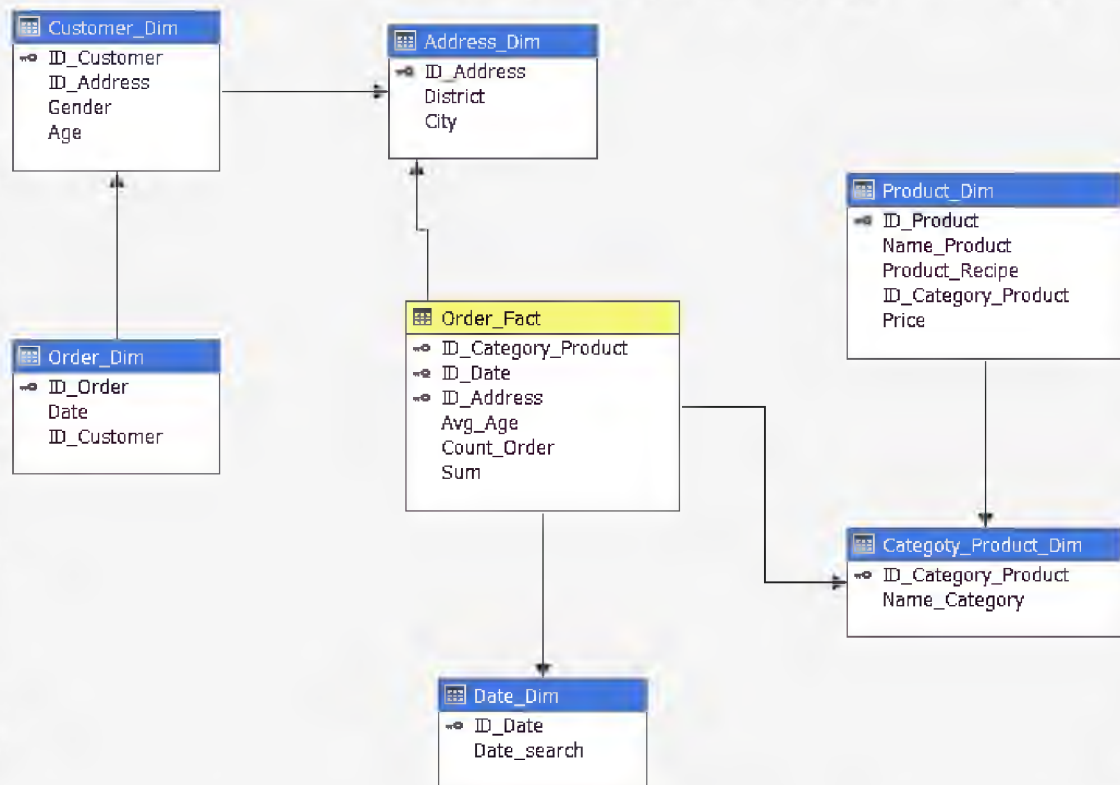


Рис. 11 Розгорнутий куб.

3. РОЗРОБКА СИСТЕМИ

НУБІП України

3.1 Архітектура системи

Якщо говорити про архітектуру системи (рис. 12) то можна виділити

наступні частини:

- сервер, на якому розміщена веб-платформа, що надає можливість вибору, ознайомлення та оформлення замовлення страв для клієнта;
- пристрій користувача з підтримкою сучасного веб-браузера через який відбувається взаємодія з веб-системою;
- сервер операційної бази даних;
- сервер сховища даних;
- середовище OLAP куба;
- робоче місце аналітика, яке зв'язане з OLAP кубом.

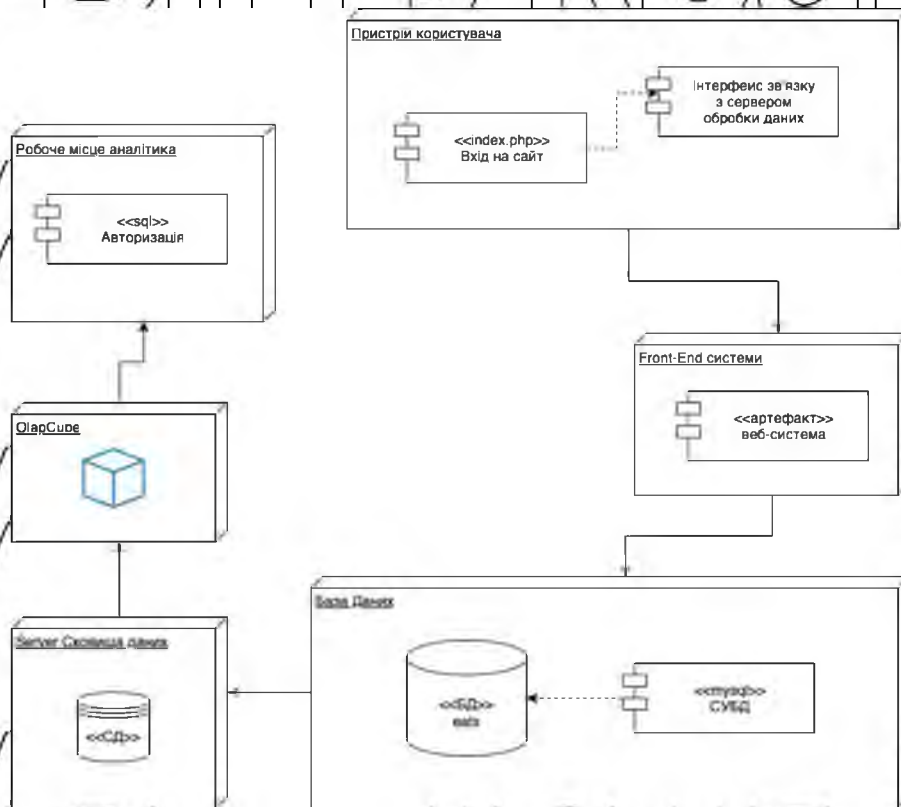


Рис. 12 Топологія системи

3.2 Інформація та забезпечення

3.2.1 Операційна база даних. Реляційна модель даних - логічна модель

даних, прикладна теорія побудови баз даних, яка є додатком до завдань обробки даних таких розділів математики як теорії множин і логіка першого порядку. На реляційній моделі даних будуються реляційні бази даних. Реляційна модель даних включає такі компоненти: Структурний аспект (складова) - дані в базі даних є набором відносин. Аспект (складова) цілісності - відносини (таблиці) відповідають певним умовам цілісності. РМД підтримує декларативні обмеження цілісності рівня домену (типу даних), рівня відносини і рівня бази даних. Аспект (складова) обробки (маніпулювання) - РМД підтримує оператори маніпулювання відносинами (реляційна алгебра, реляційне числення). Крім того, до складу реляційної моделі даних включають теорію нормалізації. [2].

В ній виділено 9 головних сутностей:

- **Administrator** – сутність зберігає інформацію про адміністратора для авторизації в адмін панелі.
- **Category** – сутність зберігає інформацію про категорії, до яких належать страви.
- **Category_Description** – сутність зберігає текстову інформацію метадані про категорії, до яких належать страви.
- **Product** – сутність зберігає інформацію про страви. Її вартість, кількість у наявності, статус і т.п.
- **Product_Description** – сутність зберігає текстову інформацію метадані про страви.
- **Product_Attribute** та **Attribute** – сутність зберігає характеристики про страви. Наприклад ваша, розмір, склад і т.п.
- **Product_Option** – сутність зберігає додаткові параметри для страв. Наприклад розмір, добавки до страви і т.п.
- **Order_Product** – сутність зберігає характеристики про страви які

НУБІП України

були додані в замовлення

- **Order** – сутність зберігає всю інформацію стосовно замовлення.
- **Customer** – сутність зберігає всю інформацію про користувача.

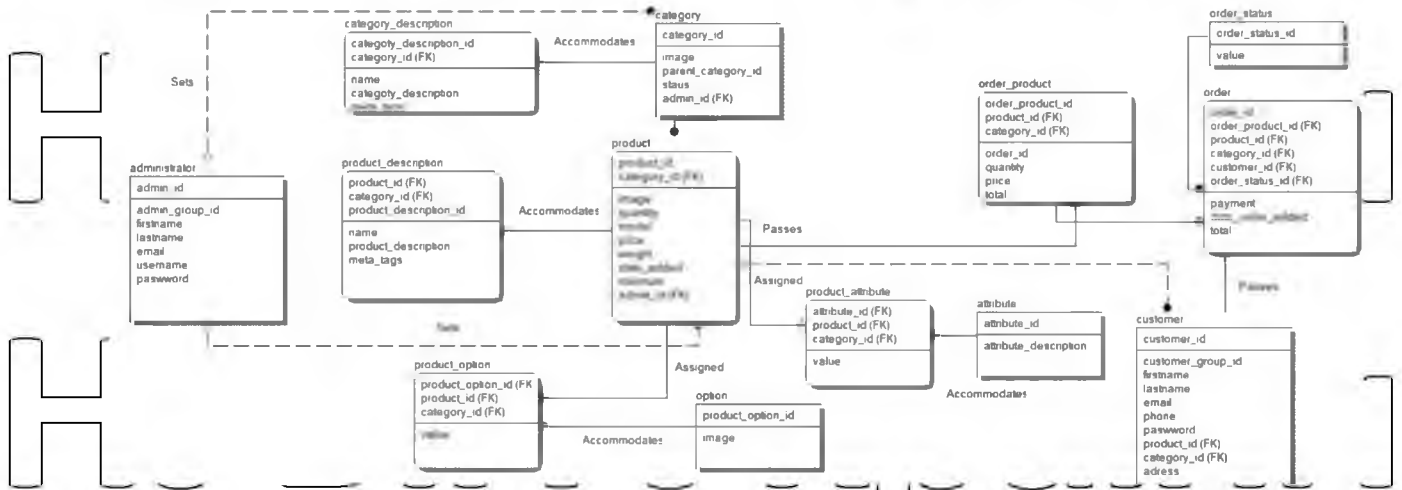


Рис. 13 Логічна модель даних

Для створення таблиць використовується функція «CREATE TABLE» після якої, вказується назва нової таблиці. Далі вказуються значення таблиці та їх типи. NOT NULL вказує на те, що значення не може бути пустим. PRIMARY KEY використовується для однозначної ідентифікації рядків в таблиці. Значення первинного ключа має бути завжди унікальним і не повторюватись. Приклад створення таблиці administrator та category зображено на рисунку 14 та 15.

```
CREATE TABLE administrator (admin_id INTEGER NOT NULL,
admin_group_id INTEGER NOT NULL,
firstname VARCHAR () NOT NULL,
lastname VARCHAR (20) NOT NULL,
email VARCHAR (20) NOT NULL,
username VARCHAR (20) NOT NULL,
password VARCHAR (20) NOT NULL);
ALTER TABLE administrator ADD PRIMARY KEY (admin_id);
```

Рис. 14 Створення таблиці «administrator»

НУБІП України

```
CREATE TABLE category (category_id INTEGER NOT NULL,  
image VARCHAR (20) NULL,  
parent_category_id INTEGER NOT NULL,  
staus TINYINT NULL, admin_id INTEGER NULL);  
ALTER TABLE category ADD PRIMARY KEY (category_id);
```

Рис. 15 Створення таблиці «category»

Для поточної системи було створено 13 таблиць. Запити на створення всіх таблиць можна побачити у додатку А.

3.2.2 Операційна база даних. OLAP (англ. online analytical processing, аналітична обробка у реальному часі) — це інтерактивна система, що дозволяє переглядати різні підсумки по багатовимірних даних. Термін "в реальному часі" (англ. online) означає, що нові результати отримуються протягом секунд, без довгого очікування на результат запиту.

Сновоположником OLAP є автор реляційної моделі даних Едгар Кодд, який запропонував у 1993 році «12 правил аналітичної обробки в реальному часі» за аналогією з раніше сформульованими ним 12-ма правилами для реляційних БД. У 1995 році Едгар Кодд додав ще 6 правил та переформатував їх. У 2001 році для визначення OLAP був запропонований більш простий тест FASMI (4 вимоги).

Як зазначив Е. Кодд, головною причиною розробки OLAP стала невідповідність класичної реляційної моделі даних потребам аналітиків щодо швидкого (online) отримання відповідей на різноманітні евристичні аналітичні запити. Реляційні БД зберігають сутності в окремих таблицях, які зазвичай добре нормалізовані — ця структура зручна для операційних БД (транзакційні системи, OLTP — Online Transaction Processing), але складні багатотабличні запити в ній виконуються відносно повільно. Зручнішою моделлю для виконання запитів (але не для внесення змін) є просторова БД. OLAP робить миттєвий знімок реляційної БД і структурує її в просторову модель для запитів. Заявлений час обробки запитів в OLAP становить близько 0,1 % від

аналогічних запитів до реляційної БД.

Основою концепції OLAP є ідея віртуально багатовимірного OLAP-куба (гіперкуба). Осей (вимірами) OLAP-кубу є числові або короткі лінгвістичні дані про предметну область роботи. Загальноприйнята назва "багатовимірний куб"

(OLAP-куб) є умовною, адже його осі даних мають різну довжину. Для аналізу

утворюють OLAP-гіперкуби, які мають як мінімум кілька осей різної координатної довжини. У великих системах вхідні дані для OLAP можуть бути попередньо узагальненими у сховищі даних (Data Warehouse), адже дані у

системах реєстрації транзакцій (OLTP-системах) безперервно змінюються, для

прикладу, дані у системах реєстрації продажів товарів, квитків тощо.

У теперішній час OLAP-куб часто створюють за допомогою реляційних баз даних із застосуванням схем "зірка", а також "сніжинка". В центрі «зірки»

знаходиться таблиця, яка містить ключові факти відповідно до їх назв у сховищі

чи кіоску даних. До таблиці фактів приєднується необхідна кількість таблиць-

вимірів, які є "променями зірки" у схемі бази даних ROLAP-моделі. Назви стовпчиків цих таблиць — це первинні дані, на основі яких можуть виконуватися базові OLAP-операції. Кількість можливих агрегацій визначається кількістю

способів, якими первинні дані можуть бути ієрархічно відображені. OLAP-куб

забезпечує відповіді на різноманітні аналітичні запити у рамках

даних у сховищі даних (кіоску даних) і їх OLAP-агрегатів. Через величезну кількість агрегатів, часто повний розрахунок відбувається тільки для деяких вимірювань, для останніх же проводиться «на вимогу».

Традиційно OLAP-системи поділяють на такі види:

багатовимірна OLAP (Multidimensional OLAP), MOLAP,

реляційна OLAP (Relational OLAP), ROLAP,

гібридна OLAP (Hybrid OLAP), HOLAP.

MOLAP це класична форма OLAP, так що її часто називають просто OLAP. Вона

використовує підсумовуючу БД, спеціальний варіант процесора просторових БД

і створює необхідну просторову схему даних зі збереженням як базових даних,

так і агрегатів. ROLAP працює безпосередньо з реляційним сховищем, факти і

таблиці з вимірюваннями зберігаються в реляційних таблицях, і для зберігання агрегатів створюються додаткові реляційні таблиці. MOLAP використовує реляційні таблиці для зберігання базових даних і багатовимірні таблиці для агрегатів. Особливим випадком ROLAP є ROLAP реального часу (Real-time ROLAP, або R-ROLAP). На відміну від ROLAP, в R-ROLAP для зберігання агрегатів не створюються додаткові реляційні таблиці, а агрегати розраховуються у момент запиту. При цьому багатовимірний запит до OLAP-системи автоматично перетворюється в SQL-запит до реляційних даних.

Кожен тип зберігання має певні переваги, хоча є розбіжності в їх оцінці у різних виробників. MOLAP краще всього підходить для невеликих наборів даних, він швидко розраховує агрегати і дає відповіді, але при цьому генеруються величезні обсяги даних. ROLAP оцінюється як більш масштабоване рішення, яке до того ж використовує найменший можливий простір. При цьому швидкість обробки значно знижується. HOLAP знаходиться між цими двома підходами, він досить добре масштабується і швидко обробляється. Архітектура R-ROLAP дозволяє проводити багатовимірний аналіз OLTP-даних в режимі реального часу.

Складність в застосуванні OLAP полягає в створенні запитів, виборі базових даних і розробці схеми, внаслідок чого більшість сучасних продуктів OLAP поставляються разом з величезною кількістю задалегідь сконфігурованих запитів. Інша проблема полягає в базових даних. Вони повинні бути повними і несуперечливими [3].

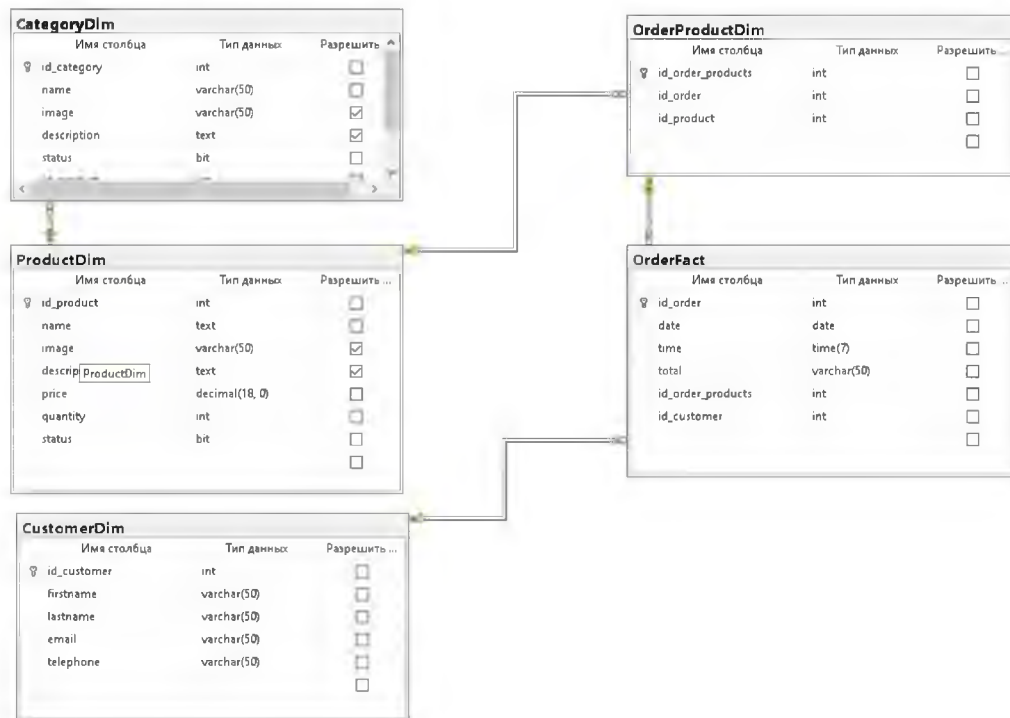
3.2.3 Структура сховища даних. За допомогою сховища даних ми змогли реалізувати можливість накопичення даних по діяльності платформи, але хоча сховище даних дозволяє зібрати в одному місці дані, які можуть повністю описати діяльність платформи, але сховище даних не надає додаткових інструментів для проведення аналізу.

Головним етапом створення системи є створення сховища даних (СД), наповнення якого буде відбуватись за рахунок оперативної БД. На рисунку 16 представлено створене СД.

Сховище даних містить 5 таблиць-вимірів та 1 таблиця-фактів.

Вимір – це безліч об'єктів одного або декількох типів, організованих у вигляді ієрархічної структури і забезпечує інформаційний контекст числового показника (факту).

Факт – це величина (зазвичай числова), яка є предметом аналізу.



Рисунк 16. Структура сховища даних

Для збереження необхідних даних були розроблені такі таблиці-вимірів:

CategoryDim – вимір зберігає інформацію про категорії, до яких належать страви.

ProductDim – вимір зберігає інформацію про страви.

CustomerDim – вимір інформацію про користувача системи.

OrderProductDim – вимір зберігає страви які були додані в замовлення.

Таблиця фактів представленого сховища:

OrderFact – вимір зберігає всю інформацію стосовно замовлення.

Таблиці вимірів є батьківськими щодо таблиці фактів, тож первинні ключі

таблиць вимірів є зовнішніми ключами таблиці фактів. Первинний ключ таблиці фактів є складеним і складається з усіх зовнішніх ключів.

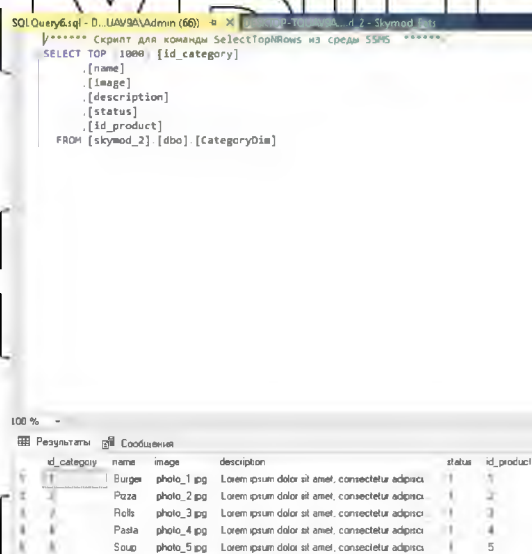
Інформація в таблицях вимірів є відносно постійною, тож дані цих таблиць заповнювалися за допомогою SQL запитів, приклад якого наведено на рисунку

17.

```
insert into ProductDim (id_product, name, image, description, price, quantity, status)
values
('1','Burger One','photo_1.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit',
'200.00', '100', '1'),
('2','Pizza Two','photo_2.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit',
'164.00', '100', '1'),
('3','Rolls California','photo_3.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing
elit', '120.00', '100', '1'),
('4','Pasta Moza','photo_4.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit',
'90.00', '100', '1'),
('5','Tomamto Soup','photo_5.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit',
'89.00', '100', '1')
```

Рисунок 17. Приклад SQL запиту на занесення даних

Виконавши всі необхідні запити, можна перевірити занесені дані. Приклад занесених даних відображено на рисунку 18.



```
SELECT TOP 1000 [id_category]
      . [name]
      . [image]
      . [description]
      . [status]
      . [id_product]
FROM [skymod_2] [dbo]. [CategoryDim]
```

id_category	name	image	description	status	id_product
1	Burger	photo_1.jpg	Lorem ipsum dolor sit amet, consectetur adipisci	1	1
2	Pizza	photo_2.jpg	Lorem ipsum dolor sit amet, consectetur adipisci	1	2
3	Rolls	photo_3.jpg	Lorem ipsum dolor sit amet, consectetur adipisci	1	3
4	Pasta	photo_4.jpg	Lorem ipsum dolor sit amet, consectetur adipisci	1	4
5	Soup	photo_5.jpg	Lorem ipsum dolor sit amet, consectetur adipisci	1	5

Рисунок 18. Приклад занесених даних

3.3 Тестування системи

На головній сторінці сайту користувач може одразу побачити категорії які йому пропонуються, побачити рекомендації по стравам та дізнатися які страви найбільш популярні за оцінками користувачів. Цією інформацією керує адміністратор сайту, який може вносити свої зміни в залежності від ситуації. Також на головній сторінці є банерний слайдер де адміністратори можуть розмістити акційні пропозиції або якісь новинки в меню. На рисунку 19 зображені головні категорії які може побачити користувач на головній сторінці.



Рисунок 19. Головна сторінка

Щоб користувачам було легше обрати страву для замовлення, кожна страву має окрему сторінку де про неї все детально розписано. Цю інформацію додають адміністратори сайту. Також на кожній страві є своя оцінка яку формують користувачі. Тобто кожен, може залишити свій відгук про певну страву і оцінити її по п'яти бальній шкалі. В подальшому, ця інформація може знадобитися для маркетингових дій в плані просування популярних страв. На рисунку 20 зображено приклад відгуків і оцінок на певну страву.



Рисунок 20. Відгук про страву

Для зручності, користувачам запропоновано скористуватися пошуком на сайті, щоб вдалося швидше знайти те що йому необхідно. Також при відкритті пошуку, він відобразить найбільш переглядаємі страви. Кожен запит користувача записується в базу даних і потім відображається в адмін-панелі адміністраторам. Завдяки цьому можна розуміти, що саме користувачі шукають найбільше, а що найменше. На рисунку 21 відображений пошук на сайті.

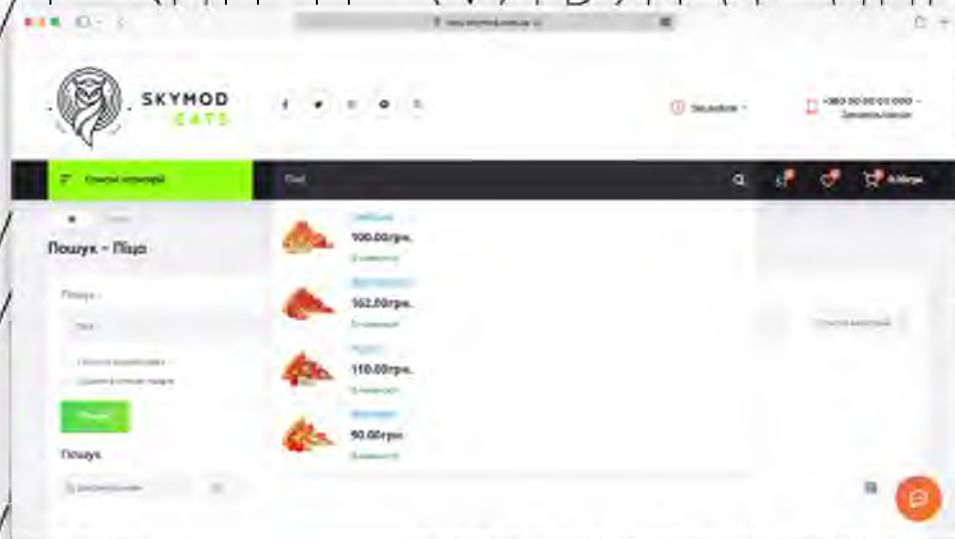


Рисунок 21. Пошук страви

Для покращення взаємодії із системою, користувачам пропонується створити власний обліковий запис. Завдяки цьому вони зможуть в декількох випадках: замовити страву, переглядати історію замовлень, отримувати персональні пропозиції та створювати списки «улюблених» страв. В свою чергу адміністратор зможе бачити статистику по замовленням певного користувача, що може надавати додаткові взаємодії із користувачем. На рисунку 22 зображено приклад «улюблених» страв користувача.

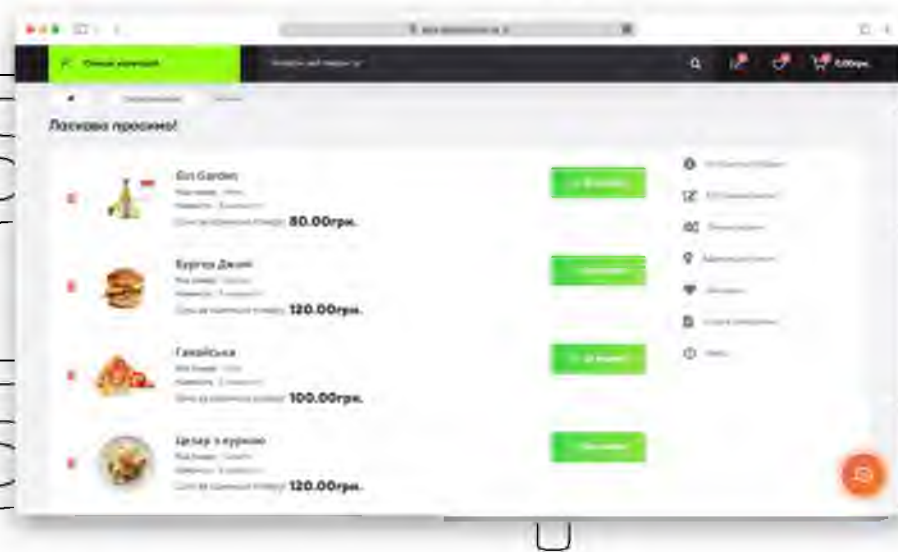


Рисунок 22. Улюблені страви користувача

Для покращення роботи персоналу та більш оперативного спілкування з замовленнями, всі замовлення що поступають з сайту, окрім адмін-панелі, надходять боту в телеграм, який в свою чергу розсилає ці замовлення необхідним працівникам. Таким чином, вдається отримати головну інформацію про замовлення моментально і не має необхідності в постійному перегляді адмін панелі. На рисунку 23 зображен приклад замовлення яке поступило в Телеграм.



Рисунок 23 Інформація про замовлення з сайту в Telegram

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

4.1 Нечітка експертна система

Експертні системи - це клас комп'ютерних програм, які пропонують рекомендації, проводять аналіз, виконують класифікацію, дають консультації і ставлять діагноз. Вони орієнтовані на розв'язування задач, вирішення яких вимагає проведення експертизи людиною-спеціалістом.

Головною відмінністю ЕС від інших програмних засобів є наявність бази знань, в якій зберігаються знання експертів про відповідну проблемну область.

Знання зберігаються у вигляді сукупності записів деякою мовою подання знань, яка дозволяє легко змінювати та поповнювати базу знань в формі, яку розуміють спеціалісти. В традиційних же програмах знання жорстко „зашиті” в алгоритм і скорегувати їх може лише сам автор програми, тобто лише програміст (за умови, що він пам'ятає за яким принципом працює одна із множини створених ним програм).

При розробленні експертних систем використовується спеціальний інструментарій, який дозволяє значно скоротити час розробки. Такий інструментарій включає як програмні, так і апаратні засоби. До апаратних засобів відносяться: ПЕОМ, інтелектуальні робочі станції, послідовні символьні ЕОМ (ЛІСП / і ПРОЛОГ машини), ЕОМ загального призначення та паралельні символьні ЕОМ. Крім того, для розширення можливостей числових процесорів всіх типів випускаються спеціальні символьні співпроцесори.

Після завершення розробки першого прототипу, експерт і інженер зі знань мають нагоду оцінити, що саме буде включено в розробку остаточного варіанта системи. Для цього іноді необхідно виділити додаткові етапи (стадії існування) при переході ЕС від прототипу до промислового зразка:

- демонстраційний прототип,
- дослідницький прототип,
- діючий прототип,

НУБІП України

- промислова система,
- комерційна система.

Найчастіше реалізується плавний перехід від демонстраційного прототипу до промислової системи.

НУБІП України

Таким чином, ми розглядатимемо інструментальні засоби побудови ЕС в якості систем програмування, які спрощують проектування ЕС. Відповідно, їх склад і структура визначаються особливостями вирішуваних експертними системами задач і технологій проектування ЕС. Необхідно відзначити, що вибір

НУБІП України

технології і інструментальних засобів реалізації ЕС – це ключове питання створення ЕС.

Для того щоб отримати певні висновки на базі яких можна приймати рішення, необхідно створити нечітку експертну систему. Для початку необхідно

НУБІП України

визначитись з вхідною та вихідною інформацією, а потім перейти до створення правил поведінки. Для побудови експертної системи будемо використовувати середовище MatLab, де одразу вкажемо вхідну та вихідну інформацію (рис. 24).

Вхідною інформацією буде кількість замовлень, оцінка страви, оцінка обслуговування, кількість переглядів страви та час приготування страви.

НУБІП України

Вихідною, тобто результатами на основі яких можна прийняти рішення буде певне рішення по меню ресторану та по роботі персоналу. [13]

НУБІП України

НУБІП України

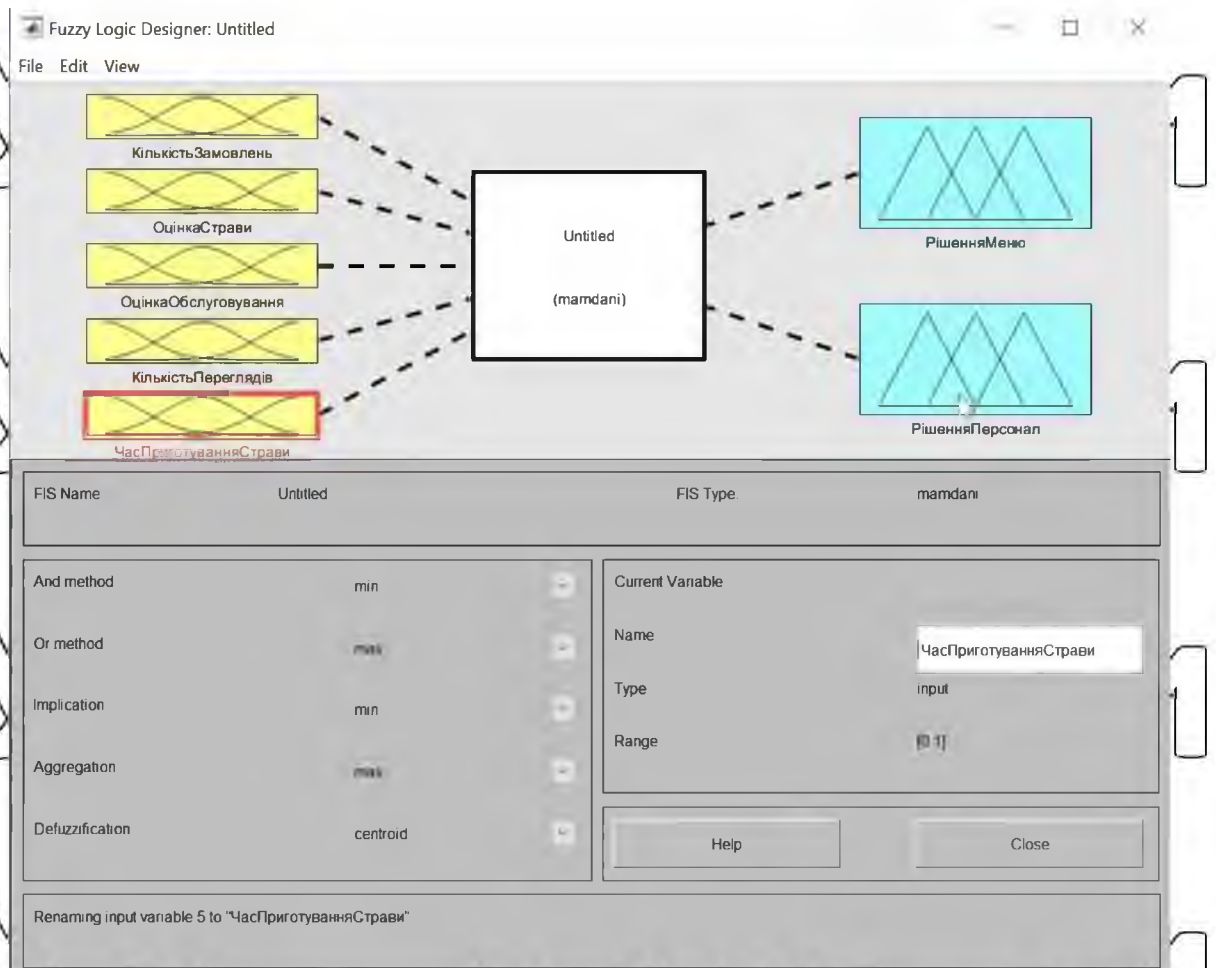
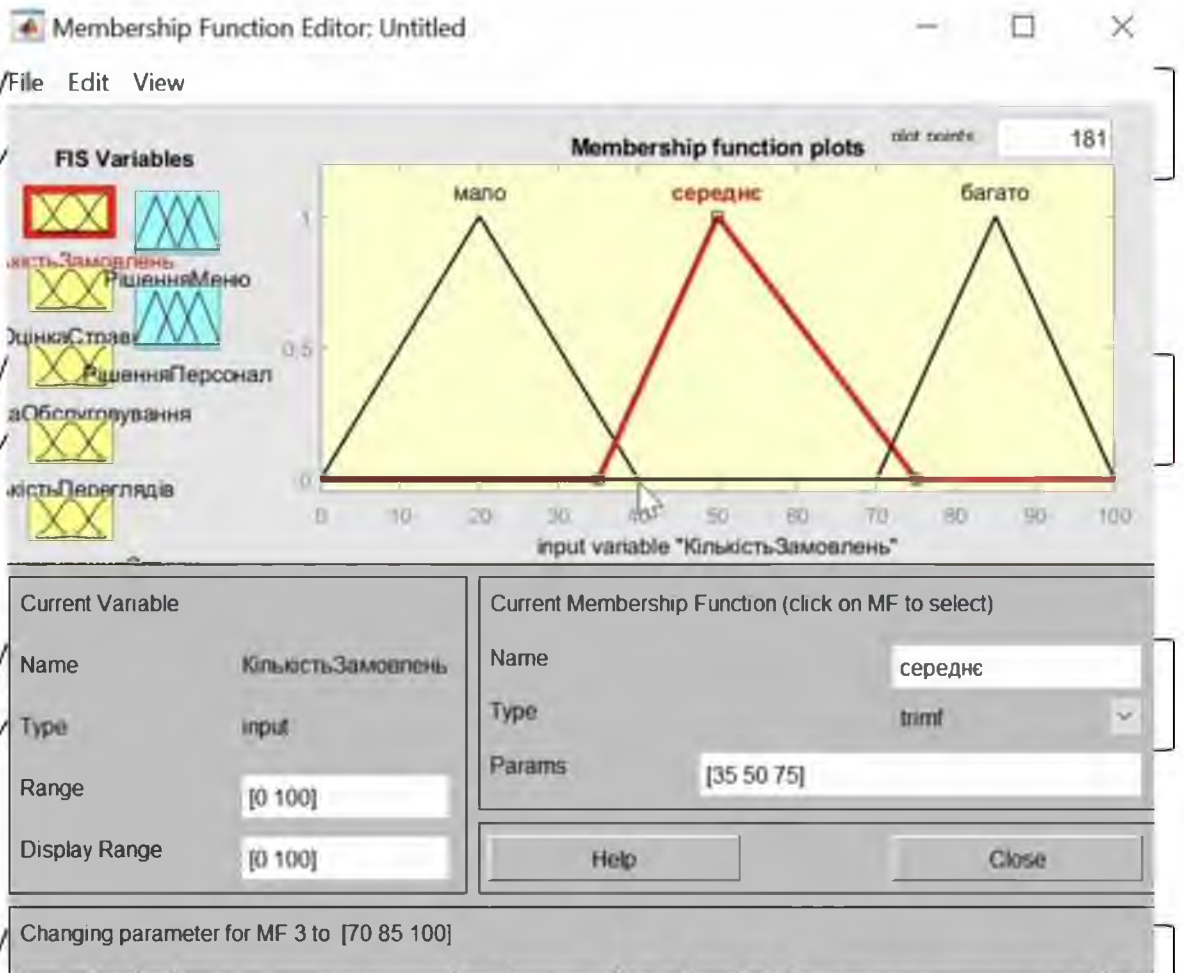


Рисунок 24. Нечітка експертна система

На кожну страву припадає певна кількість замовлень і знаючі цю інформацію ми можемо приймати певні рішення по зміні або залишенні страви для замовлення (рис. 25). Тому, ми створюємо умовний діапазон від 0 до 100, де діапазон від 0 до 40 буде вважатися малою к-стю замовлень, від 35 до 75 середньою к-стю замовлень і від 70 до 100 великою к-стю замовлень.



Рисунк 25. Відна змінна «Кількість замовлень»

На кожну страву користувач може залишити свій відгук та оцінити страву по п'яти бальній шкалі. Завдяки цій інформації ми можемо приймати певні рішення по зміні або залишенні страви для замовлення (рис. 26). Тому, ми створюємо умовний діапазон від 1 до 5, де діапазон від 1 до 2.5 буде вважатися низькою оцінкою, від 2.3 до 3.7 середньою оцінкою і від 3.4 до 5 високою оцінкою.

НУБІП України

НУБІП України

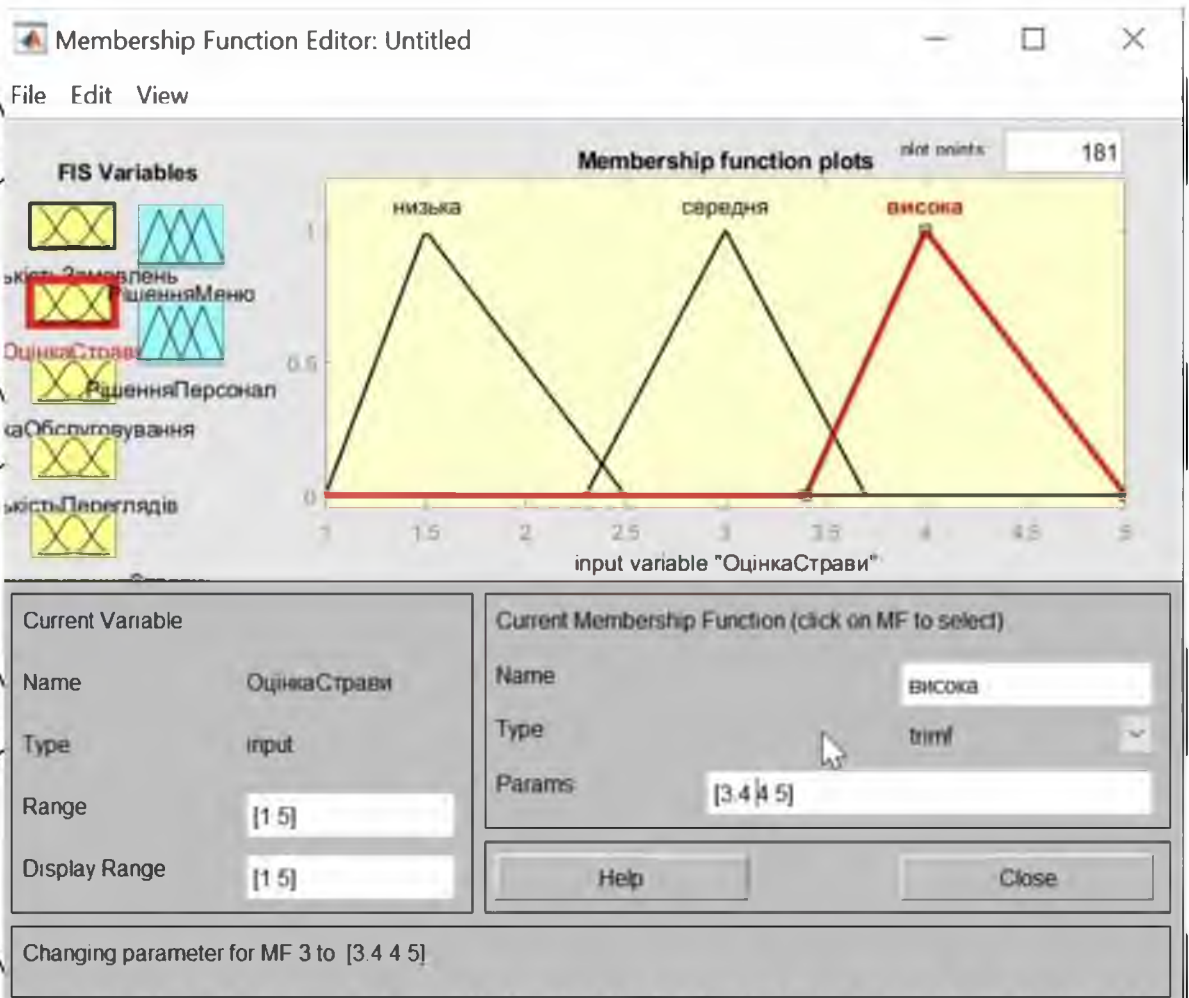


Рисунок 26. Вхідна змінна «Оцінка страви»

Перегляд страв є важливою інформаційною складовою так як на основі цієї аналітичної інформації ми можемо бачити інтерес користувачів та приймати певні рішення по страві (рис. 27). Тому, ми створюємо умовний діапазон від 0 до 100, де діапазон від 0 до 40 буде вважатися малою к-стю переглядів, від 25 до 75 середньою к-стю переглядів і від 60 до 100 великою к-стю переглядів.

НУБІП України

НУБІП України

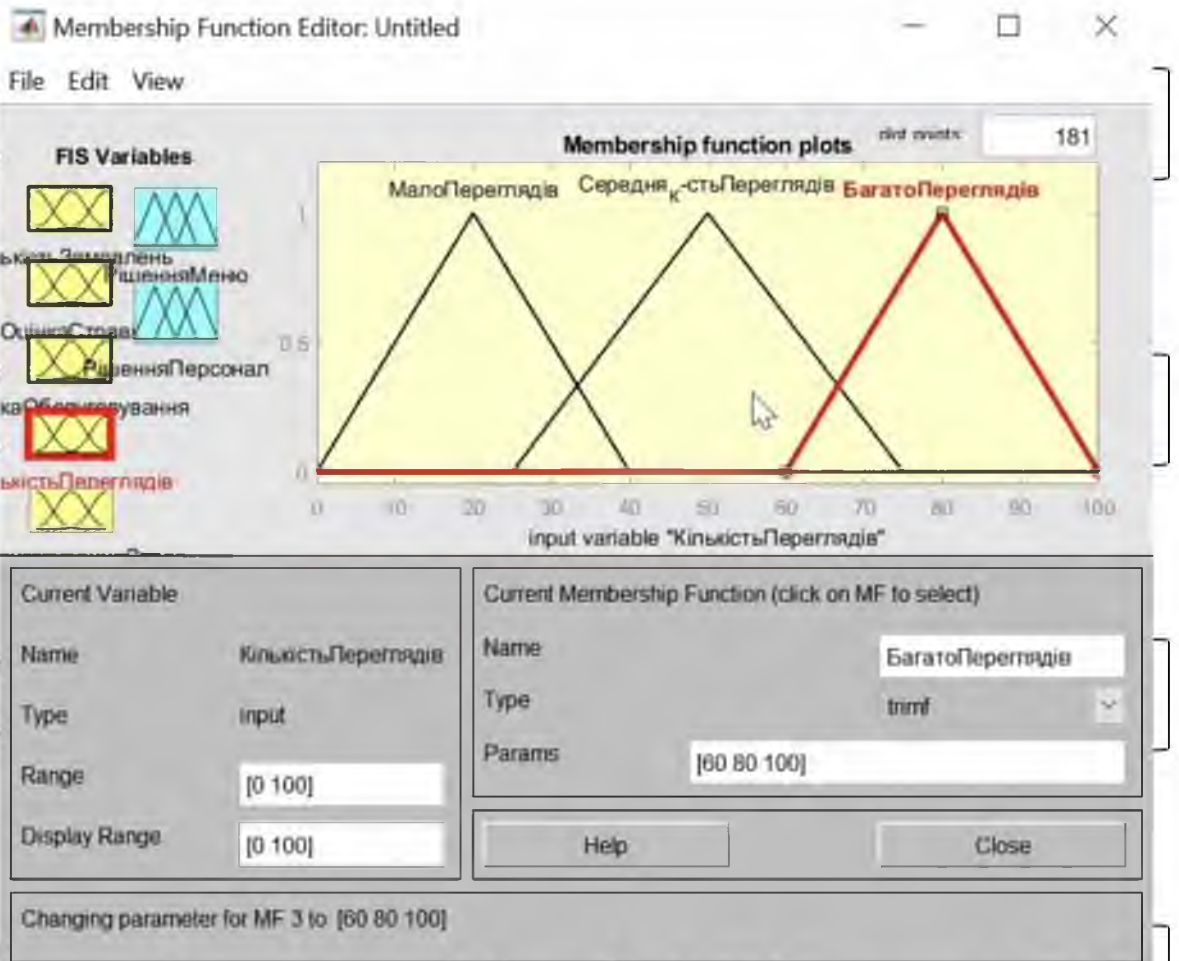


Рисунок 27. Вхідна змінна «Кількість переглядів страви»

В кожному закладі із сфери ресторанних послуг, клієнти оцінюють роботу персоналу, і завдяки оцінюванню персоналу клієнтами ми можемо приймати певні рішення по покращенню обслуговування (рис. 28). Тому, ми створюємо умовний діапазон від 1 до 5, де діапазон від 1 до 2.5 буде вважатися низькою оцінкою, від 1.8 до 3.8 середньою оцінкою і від 3.5 до 5 високою оцінкою обслуговування.

НУБІП України
 НУБІП України

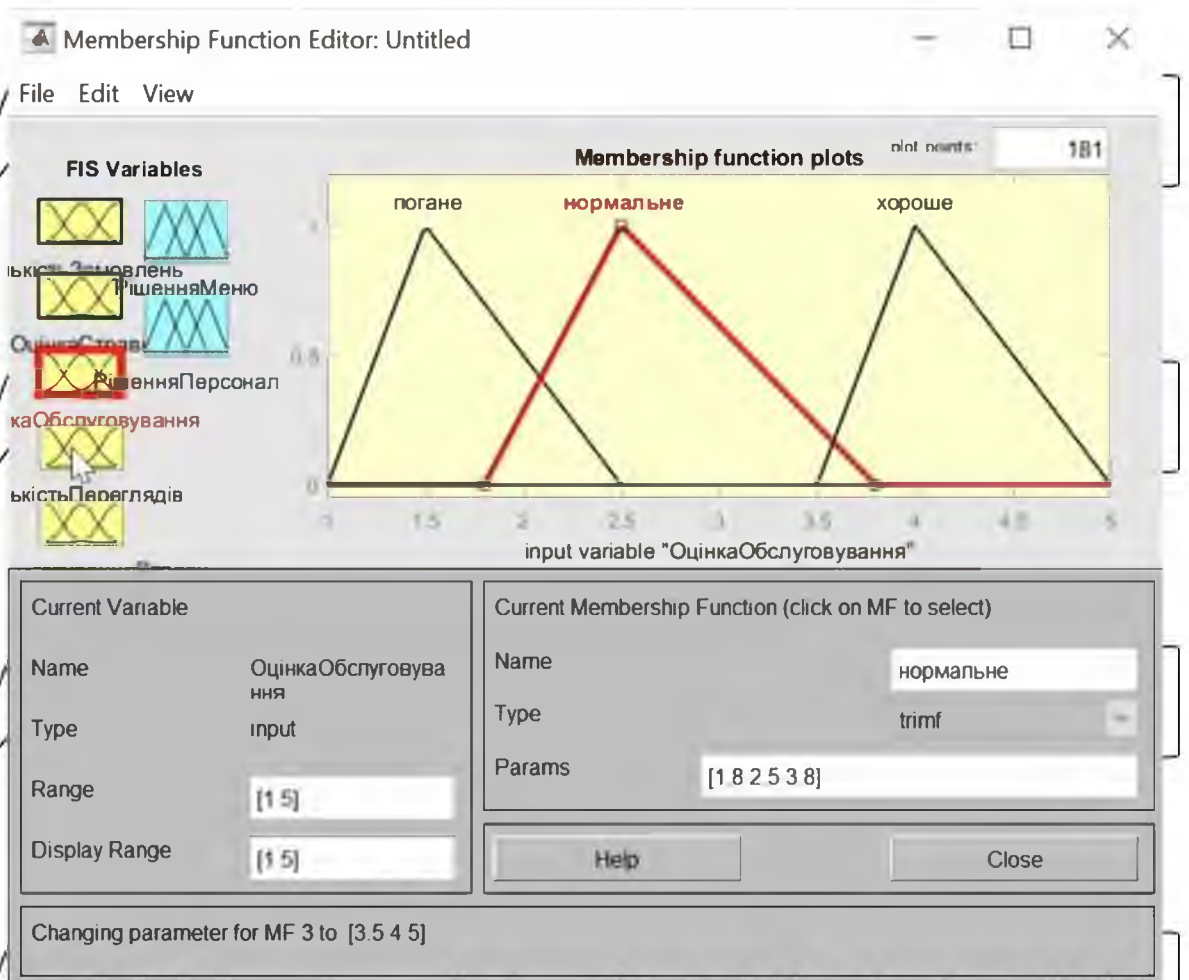


Рис. 28. Вхідна змінна «Оцінка обслуговування»

Окрім оцінки обслуговування слід окремо виділити час приготування замовлення, так як цей аспект є також дуже важливим для клієнтів (рис. 29). Тому, ми створюємо умовний діапазон від 0 до 1, де діапазон від 0 до 0.4 буде вважатися швидким приготуванням, від 0.25 до 0.75 середньою швидкістю і від 0.65 до 1 довгим приготуванням.

НУБІП України

НУБІП України

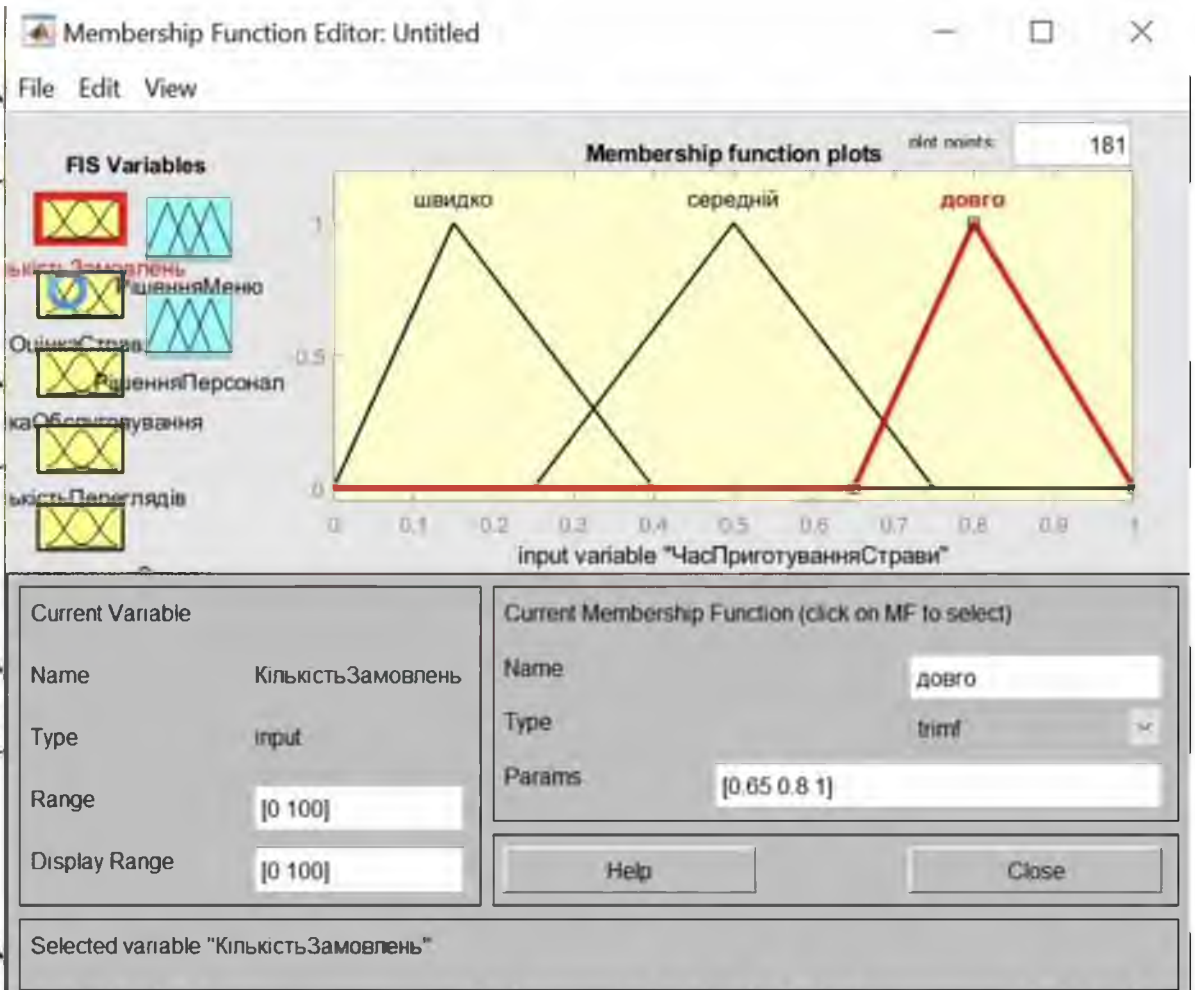


Рисунок 29. Вхідна змінна «Час приготування замовлення»

Наразі, ЕС має 5 вхідних змінних, і далі необхідно налаштувати вихідну інформацію і перейти до створення правил поведінки. Із вхідної інформації, в нас виникає дві вихідні:

- рішення по меню,
- рішення по персоналу

В плані «рішення по меню», ми створюємо умовний діапазон від 0 до 1, де створюємо три варіанта. Тобто, на основі вхідних даних та створених правил поведінки результат буде до 0.4 то ми пропонуємо «Змінити меню», якщо вище, але не досягне 1-го то ми пропонуємо «Покращити меню», і в останньому випадку «Залишити меню без змін» (рис.30).

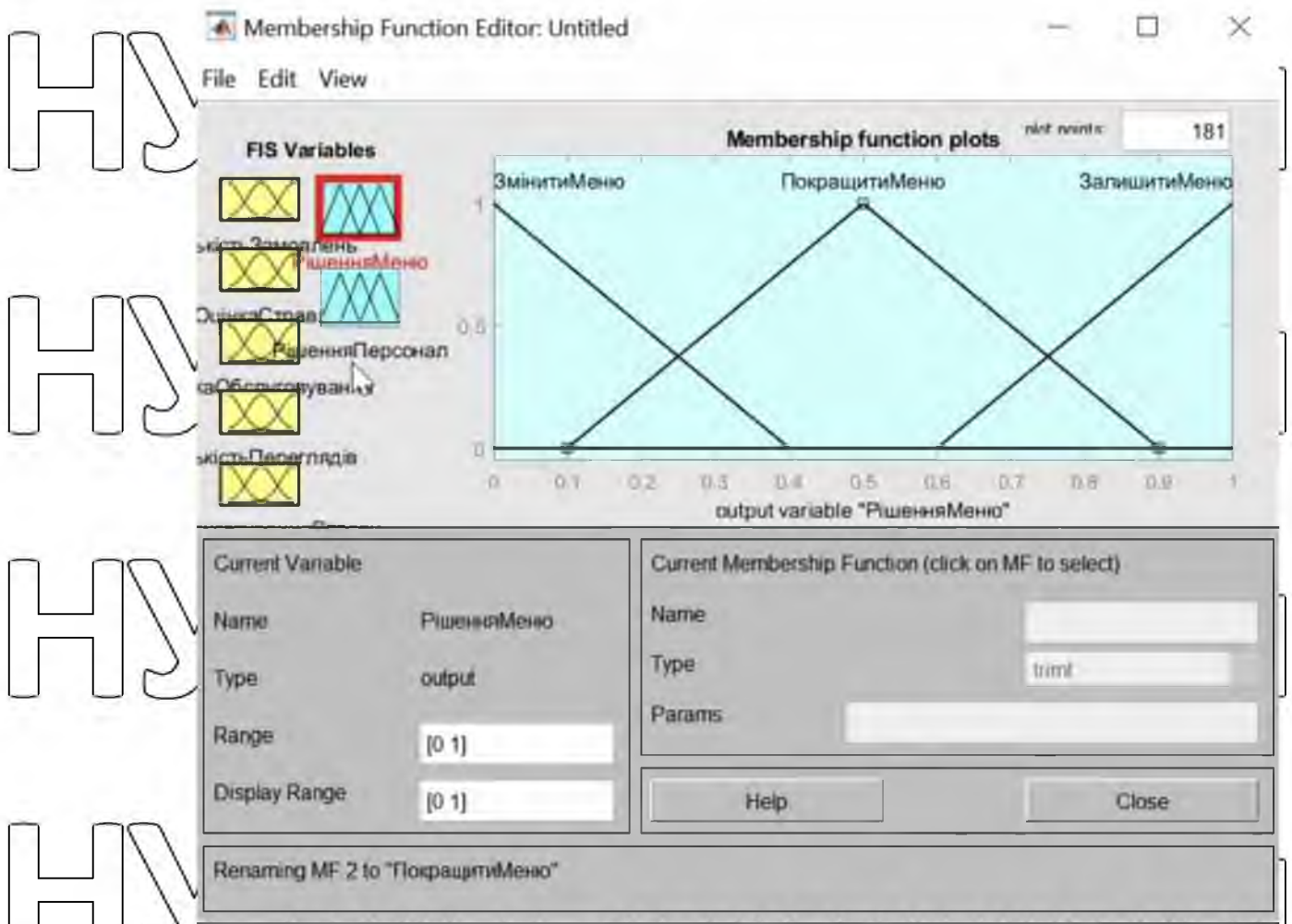


Рисунок 30. Вихідна змінна «Рішення по меню»

В плані «рішення по персоналу», ми також створюємо умовний діапазон від 0 до 1, де створюємо три варіанти. На основі вхідних даних та створених правил поведінки результат буде до 0.4 то ми пропонуємо «Змінити персонал», якщо вище, але не досягне 1-го то ми пропонуємо «Покращити роботу персоналу», і в останньому випадку «Залишити персонал без змін» (рис.31).

НУБІП України

НУБІП України

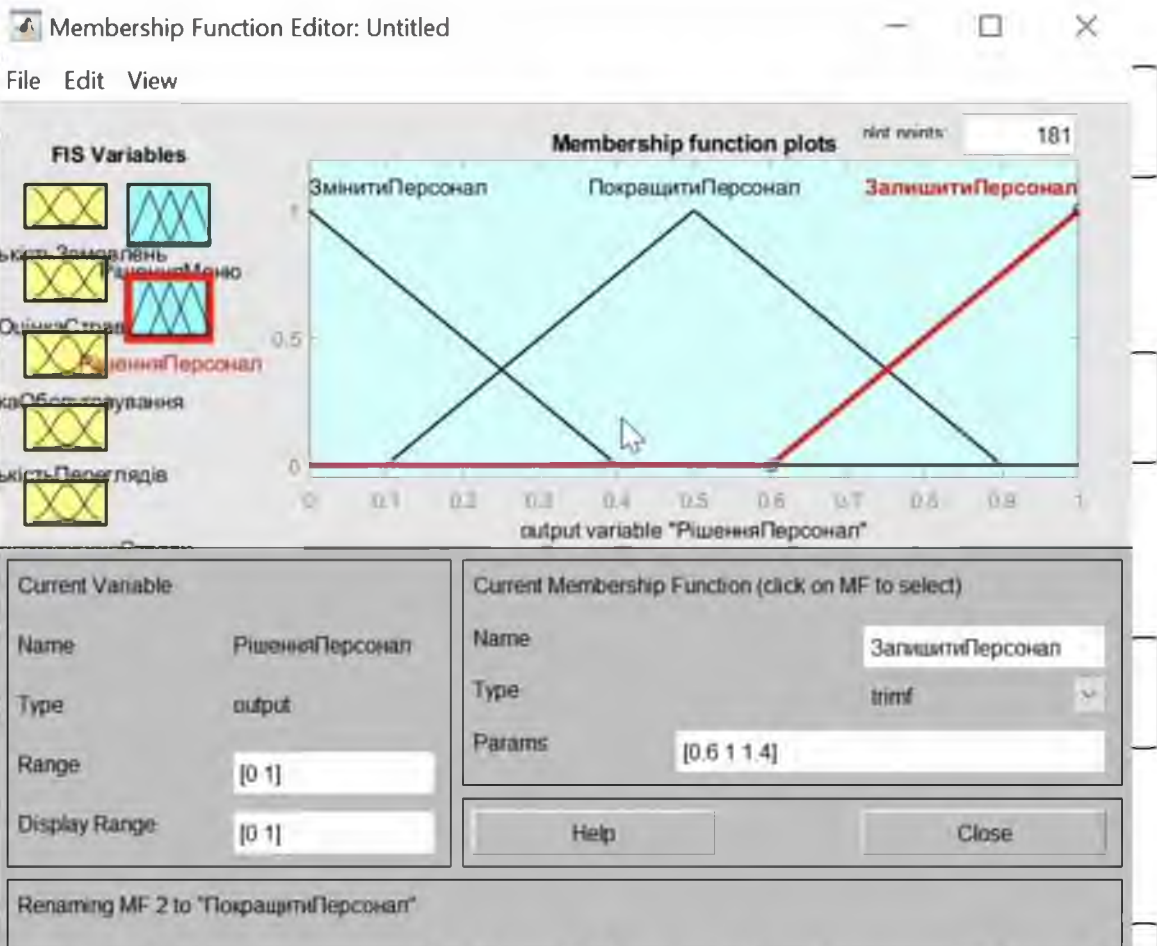


Рисунок 31. Вихідна змінна «Рішення по персоналу»

Наразі система має вхідну та вихідну інформацію. Для того щоб можна було отримувати певні висновки або ж рекомендації на основі цих даних необхідно створити правила поведінки. Наприклад, якщо к-сть замовлень мала, оцінка страви та середня к-сть переглядів то ми запропонуємо змінити меню. Якщо ж велика к-сть замовлень і велика к-сть переглядів, при цьому ми не враховуємо оцінку страви, то ми запропонуємо залишити меню без змін. І так створюємо правила на основі експертного дослідження для кожної можливої комбінації (рис. 32).

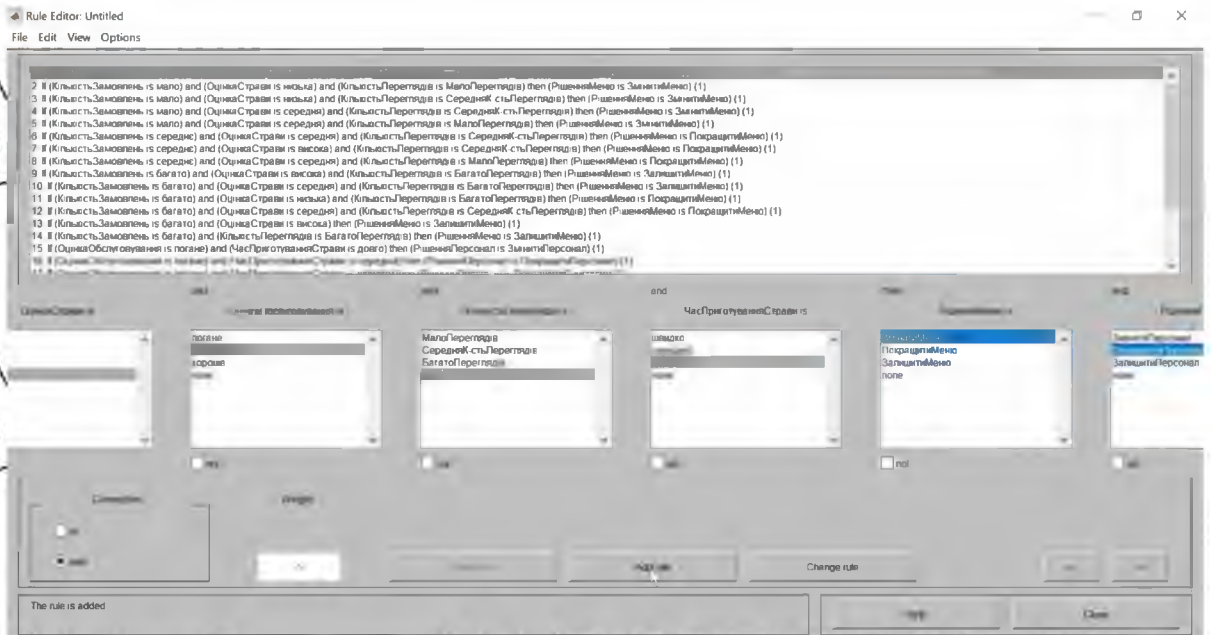


Рисунок 32. Правила поведінки нечіткої експертної системи

Далі можемо протестувати експертну систему. На рисунку 33

відображений випадок коли ми пропонуємо змінити меню. К-сть замовлень має

22.7 (низька), оцінка страви 1.83 (низька) та к-сть переглядів 21.4 (низька).

Рішення по меню має 0.131, тобто ми пропонуємо змінити меню.

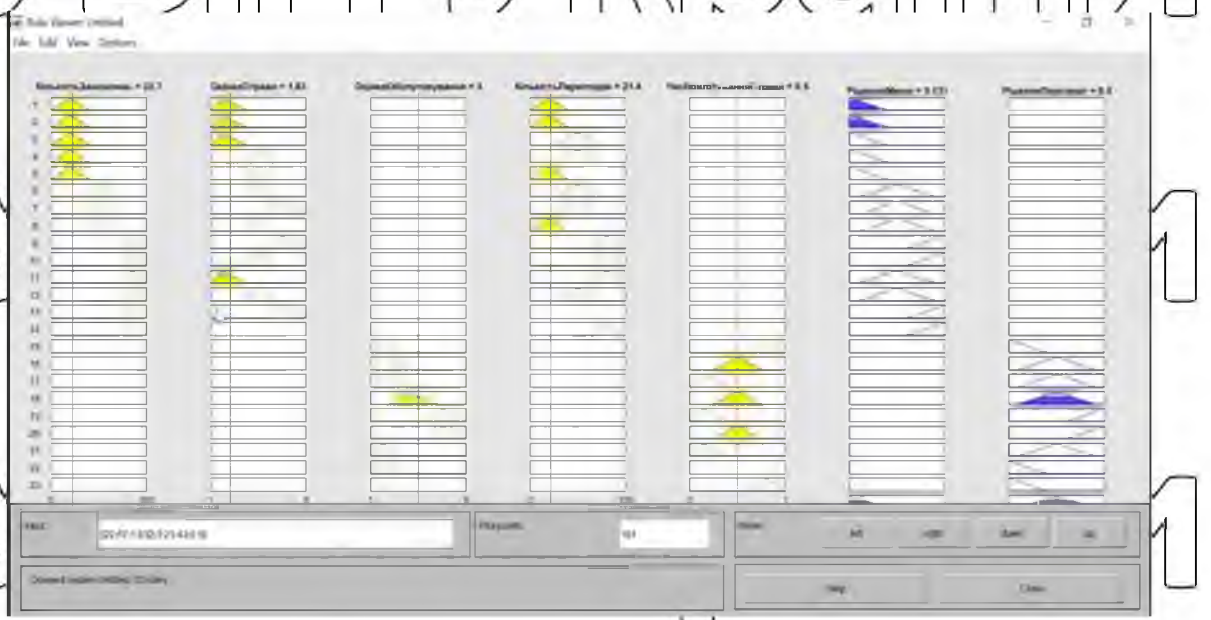


Рисунок 33. Тестування системи при низьких значеннях вхідних даних

Якщо ми покращимо к-сть замовлень, зробивши їх значення – 55 (середня),

та покращимо оцінку страви, зробивши значення – 3.2 (середня), але залишимо

к-сть переглядів на тому ж низькому рівні зі значенням 21.4 то як результат буде

запропоновано переглянути меню, але не змінювати його радикально (рис. 34).

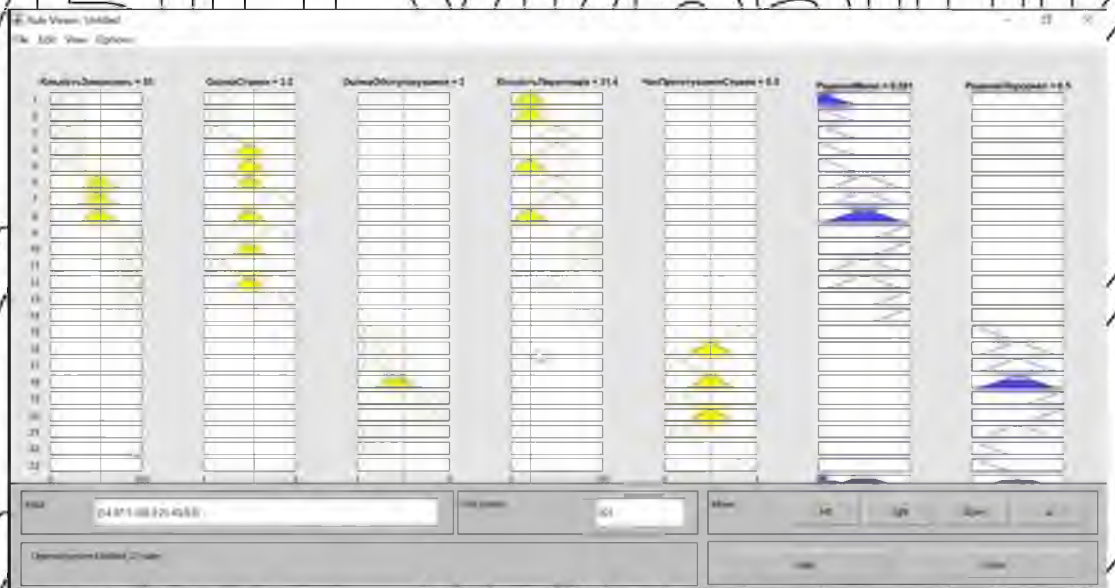


Рисунок 34. Тестування системи при середніх значеннях вхідних даних

На рисунку 35 змодельована поверхня всіх створених для експертної системи правил поведінки

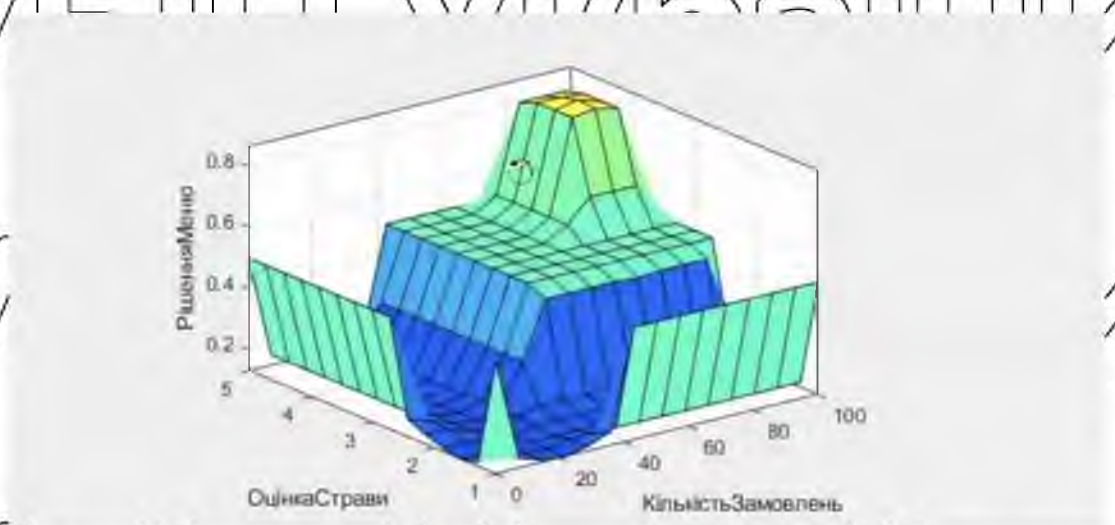


Рисунок 35. Поверхня правил поведінки

4.2 Дослідження використання задач класифікації

4.1.1 Decision Trees Algorithm. Деревя рішень [4]— це непараметричний метод навчання з керівництвом, який використовується для класифікації та регресії. Мета полягає в тому, щоб створити модель, яка прогнозує значення цільової змінної, вивчаючи прості правила прийняття рішень, виведені з

характеристик даних. Дерево можна розглядати як кусково-постійне наближення.

Деякі переваги дерев рішень:

- Простий для розуміння та інтерпретації. Дерева можна візуалізувати.
- Вимагає невеликої підготовки даних. Інші методи часто вимагають нормалізації даних, створення фіктивних змінних і видалення пустих значень. Однак зауважте, що цей модуль не підтримує пропущені значення.

- Вартість використання дерева (тобто прогнозування даних) є логарифмічною кількістю точок даних, які використовуються для навчання дерева.

- Здатний обробляти як числові, так і категоріальні дані. Однак реалізація наукового навчання наразі не підтримує категоріальні змінні. Інші методи зазвичай спеціалізуються на аналізі наборів даних, які мають лише один тип змінної. Для отримання додаткової інформації див. алгоритми.

- Здатний вирішувати проблеми з кількома виходами.
- Використовується модель білої коробки. Якщо дана ситуація спостерігається в моделі, пояснення умови легко пояснюється булевою логікою. Навпаки, у моделі чорного ящика (наприклад, у штучній нейронній мережі) результати можуть бути складнішими для інтерпретації.

- Можливість перевірити модель за допомогою статистичних тестів. Це дозволяє пояснити надійність моделі.

- Працює добре, навіть якщо його припущення дещо порушуються справжньою моделлю, на основі якої були згенеровані дані.

•

До недоліків дерев рішень можна віднести:

- Учні з дерева рішень можуть створювати надскладні дерева, які погано узагальнюють дані. Це називається переобладнанням. Щоб уникнути цієї проблеми, необхідні такі механізми, як обрізка, встановлення мінімальної

кількості зразків, необхідних у вузлі листа, або встановлення максимальної глибини дерева.

- Деревя рішення можуть бути нестабільними, оскільки невеликі варіації в даних можуть призвести до створення зовсім іншого дерева. Ця проблема пом'якшується використанням дерев рішень в ансамблі.

- Прогнози дерев рішень не є ні гладкими, ні безперервними, а кусково-постійними наближеннями, як показано на малюнку вище. Тому вони погано вміють екстраполувати.

- Відомо, що проблема вивчення оптимального дерева рішень є NP-повною за кількох аспектів оптимальності і навіть для простих концепцій. Отже, практичні алгоритми навчання дерева рішень засновані на евристичних алгоритмах, таких як жадібний алгоритм, де в кожному вузлі приймаються локально оптимальні рішення. Такі алгоритми не можуть гарантувати

повернення глобально оптимального дерева рішень. Це можна пом'якшити шляхом навчання кількох дерев у учень ансамблю, де функції та зразки випадково відбираються із заміною.

- Існують поняття, які важко вивчити, оскільки дерева рішень не легко виражають їх, наприклад, XOR, проблеми парності чи мультиплектора.

- Учні дерева рішень створюють упереджені дерева, якщо деякі класи домінують.

Тому рекомендується збалансувати набір даних перед підданням з деревом рішень.

На основі створеного раніше куба Restoran_DW створимо структуру інтелектуального аналізу на основі алгоритму дерева прийняття рішення, що представлено на рисунку 36.

НУБІП УКРАЇНИ

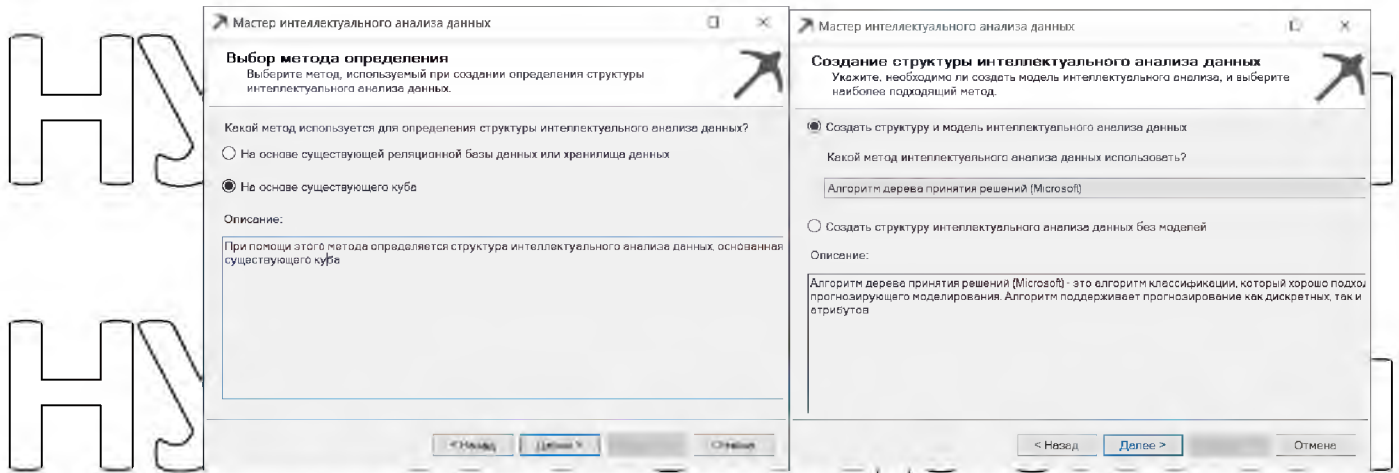


Рис. 36 Створення нової структури на основі алгоритму дерева рішень.

Далі визначаємо вимір кубу – Order_Dim та визначаємо ключове поле часу – Id_Order, що представлено на рисунку 37

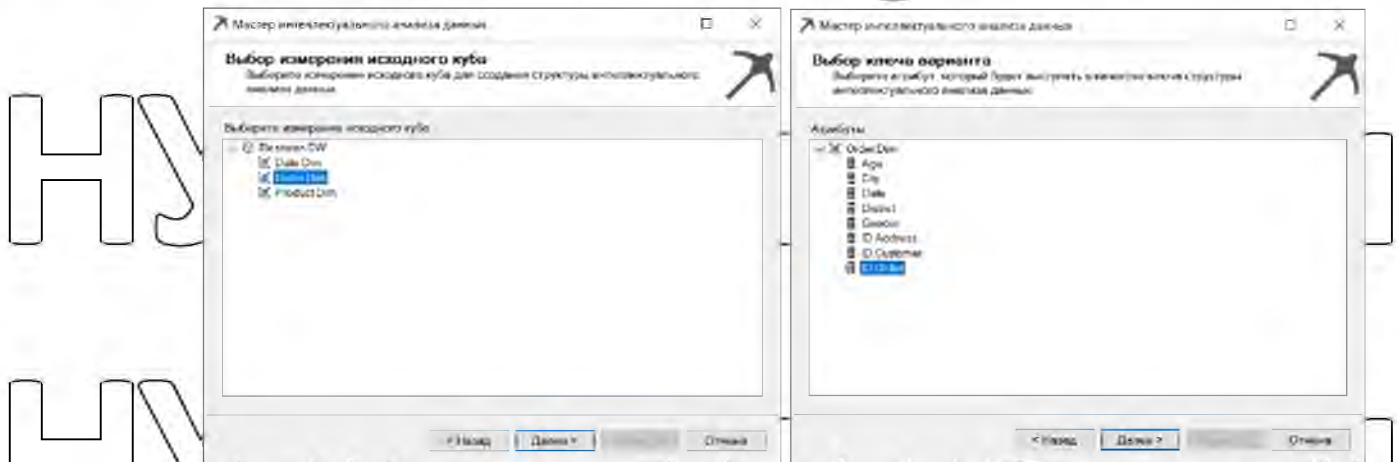


Рис. 37 Вибір ключового виміру куба та ключа структури.

Наступним кроком є вибір досліджуваних даних, а саме Sum, City, Date, District, Gender, та прогнозування значення Age, як представлено на рисунку 38

НУБІП України

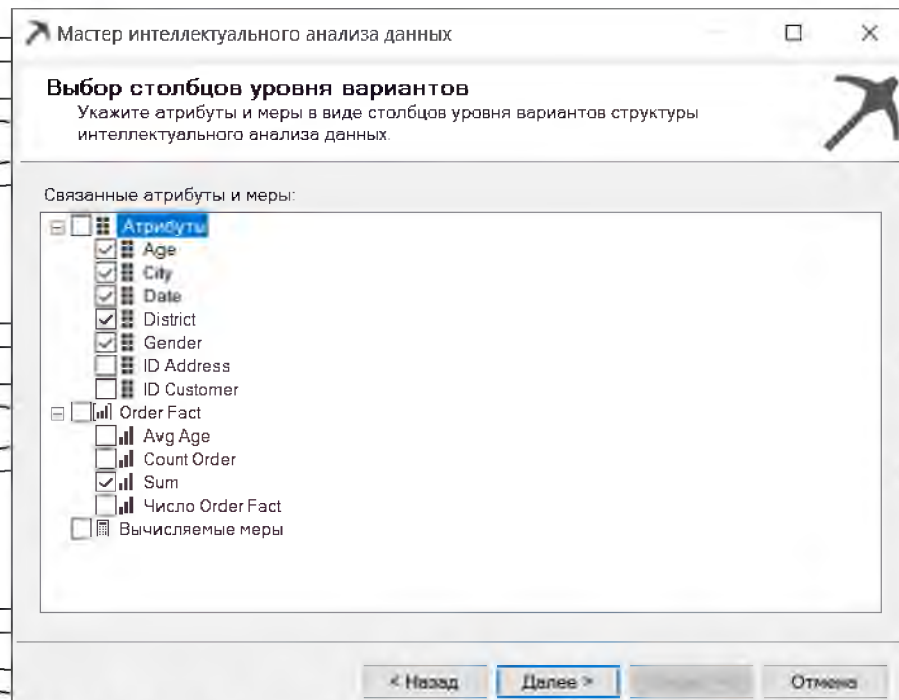


Рис. 38 Вибір полів, що аналізуємо та прогнозуємо для створюваної структури.

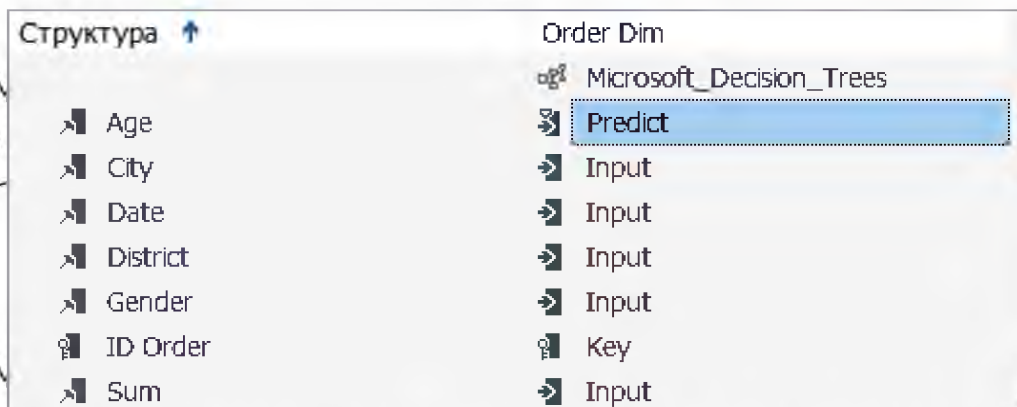


Рис. 39 Вибір полів, що аналізуємо та прогнозуємо для створюваної структури.

Визначення типу полів структури та її збереження, що представлено на рисунку 40.

НУБІП України

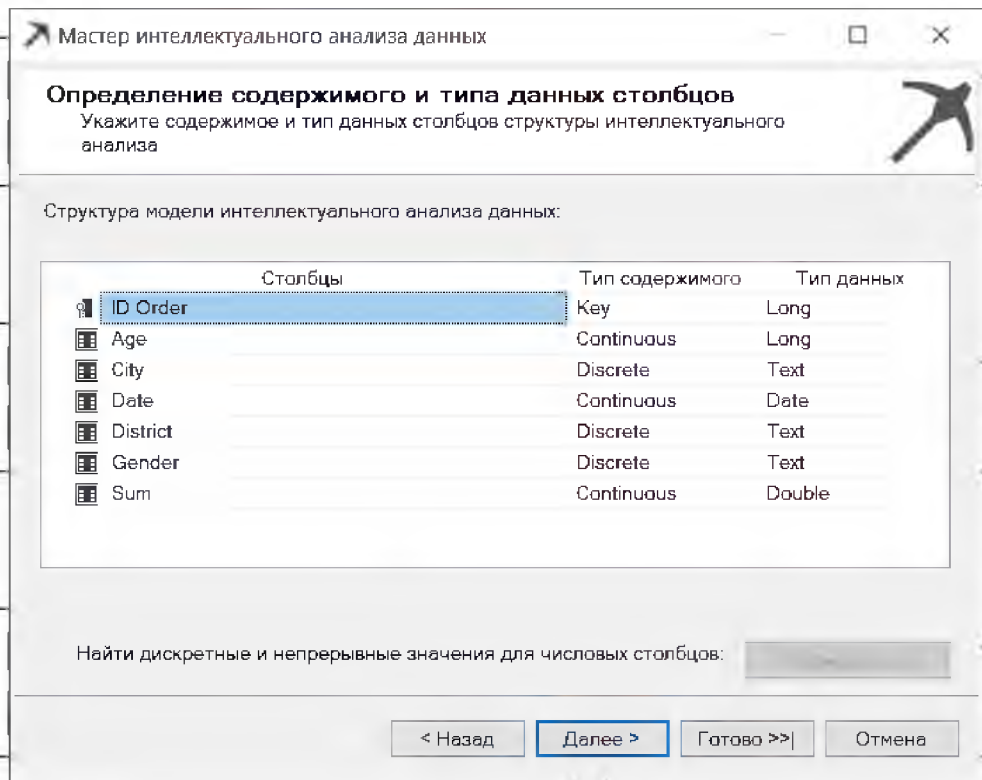


Рис. 40 Визначення типів полів та збереження структури.

Результат розторгання структури інтелектуального аналізу показано на рисунку 41.



Рис. 41 Дерево рішень.

Представлене на рисунку, дерево рішень представляє собою вітки з різними містами. Алгоритм визначив залежність віку клієнтів від їх місця знаходження та дати замовлення.

Загалом: вік клієнтів складає близько 30 років.

Якщо замовлення надійшло з міста Ірпінь, вік клієнтів складає близько 29

років.
Якщо замовлення надійшло з міста Вишгород: вік клієнтів складає близько 45 років.

Якщо замовлення надійшло з міста Київ: вік клієнтів складає близько 30

років.
Якщо замовлення надійшло з інших міст: вік клієнтів складає близько 29 років.

4.1.2 Naive Bayes Algorithm. Наївні методи Байєса [5] — це набір алгоритмів навчання з наглядом, заснованих на застосуванні теореми Байєса з «наївним» припущенням про умовну незалежність між кожною парною ознак

заданою значенням змінної класу. Теорема Байєса стверджує наступний зв'язок

із заданою змінною класу y_i залежний вектор ознак x_1 через x_n :

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)}$$

Використовуючи наївне припущення умовної незалежності, що

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),$$

для усіх i , це співвідношення спрощується до

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)}$$

Оскільки $P(x_1, \dots, x_n)$ є константою з урахуванням вхідних даних, ми можемо використовувати таке правило класифікації:

НУБІП УКРАЇНИ

$$P(y | x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i | y)$$

$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i | y),$$

Його можна використовувати для оцінки максимальну апостеріорну (MAP) оцінку $P(y)$ і $P(x_i | y)$; перша - це відносна частота класу y в тренувальній комплект.

Різні наївні байєсівські класифікатори відрізняються головним чином за припущеннями, які вони роблять щодо розподілу $P(x_i | y)$.

Незважаючи на їх, на перший погляд, загато спрощені припущення, наївні байєсівські класифікатори досить добре працювали в багатьох реальних ситуаціях, як відомо, класифікація документів і фільтрація спаму. Вони вимагають невеликої кількості навчальних даних для оцінки необхідних параметрів.

Наївні байєсівські класифікатори можуть бути надзвичайно швидкими порівняно з більш складними методами. Відокремлення розподілів умовних ознак класу означає, що кожен розподіл може бути незалежно оцінений як одновимірний розподіл. Це, у свою чергу, допомагає полегшити проблеми, пов'язані з прокляттям розмірності.

На основі створеного раніше куба `Restoran_DW` створюємо структуру інтелектуального аналізу на основі спрощеного алгоритму Баєса, що представлено на рисунку 42.

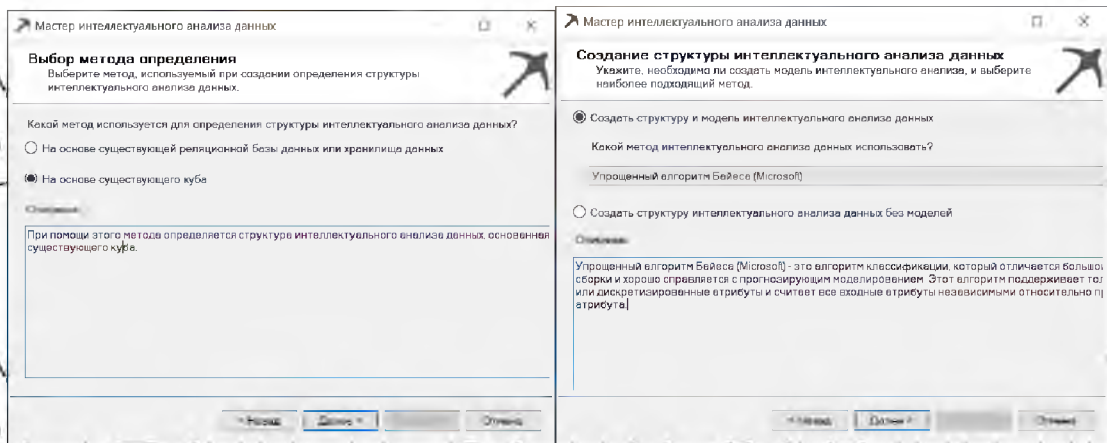


Рис. 42 Створення нової структури на основі спрощеного алгоритму Баеса.

Далі визначаємо часовий вимір кубу – Order_Dim та визначаємо ключове поле часу – Id_Order, що представлено на рисунку 43.

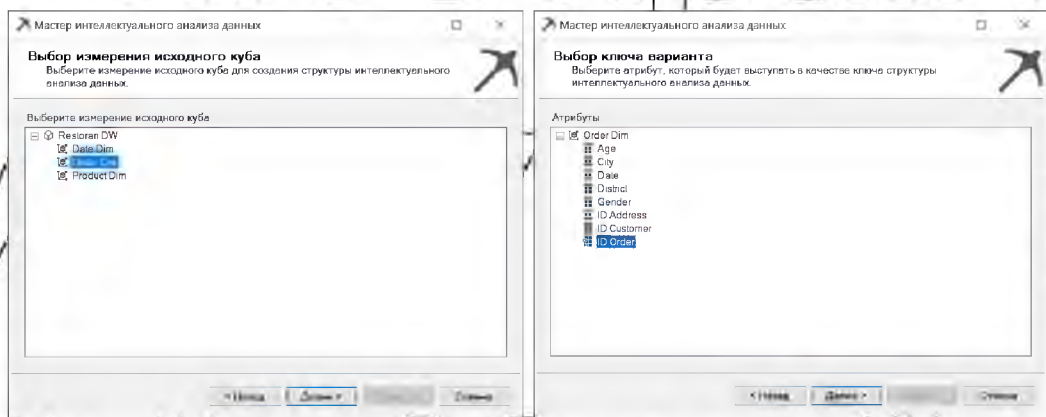


Рис. 43 Вибір ключового виміру куба та ключа структури.

Наступним кроком є вибір досліджуваних даних, а саме Age, City, Date, District, Gender, Sum, Count Order, Avg Age, та прогнозування значення Gender, як представлено на рисунку 44.

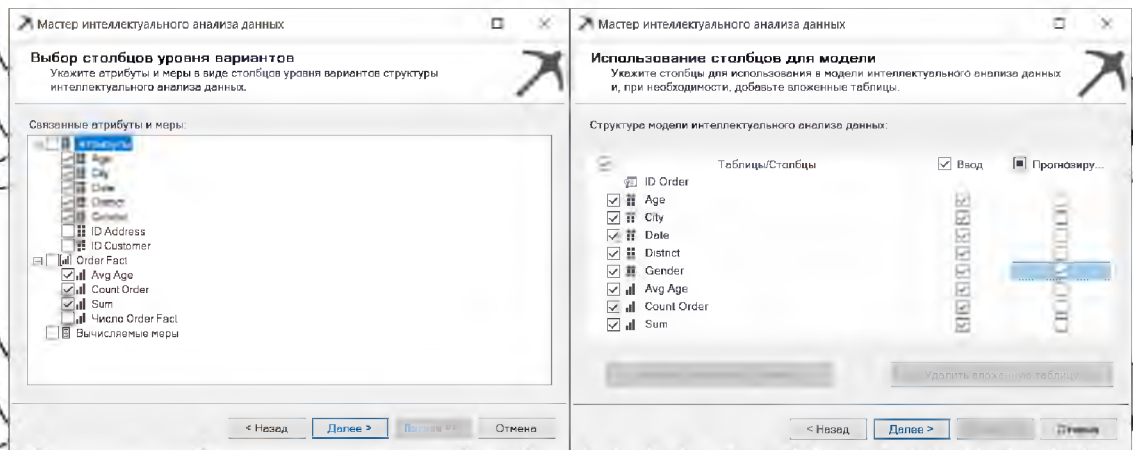


Рис. 44 Вибір полів, що аналізуємо та прогнозуємо для створеної структури.

Визначення типу полів структури та її збереження, що представлено на рисунку 45.

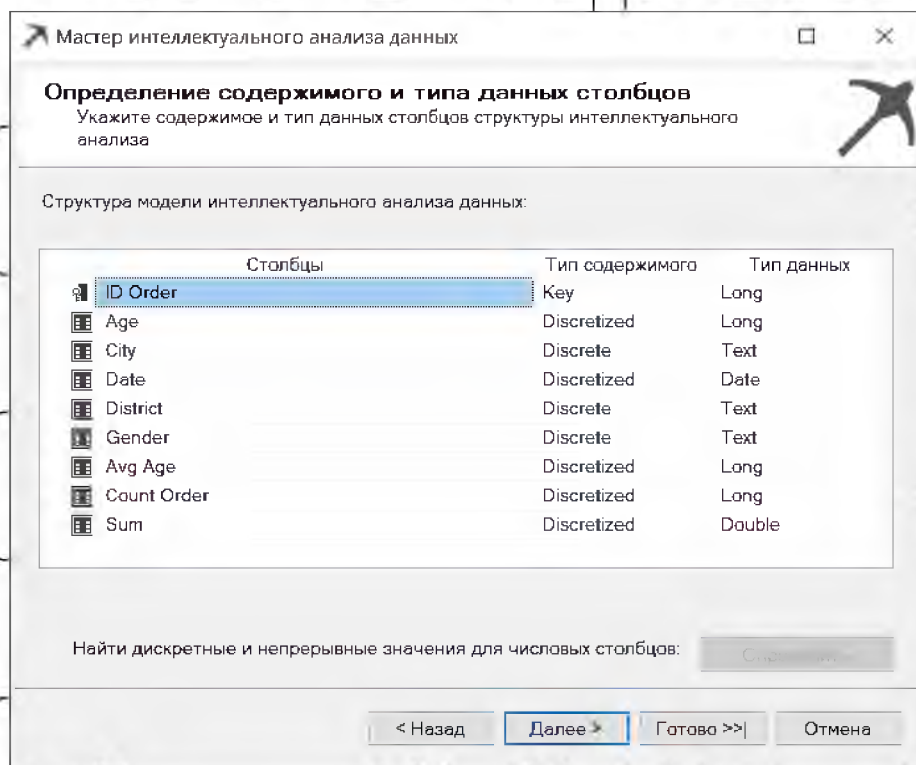


Рис. 45 Визначення типів полів та збереження структури.

Результат розгортання структури інтелектуального аналізу показано на рисунку 46.

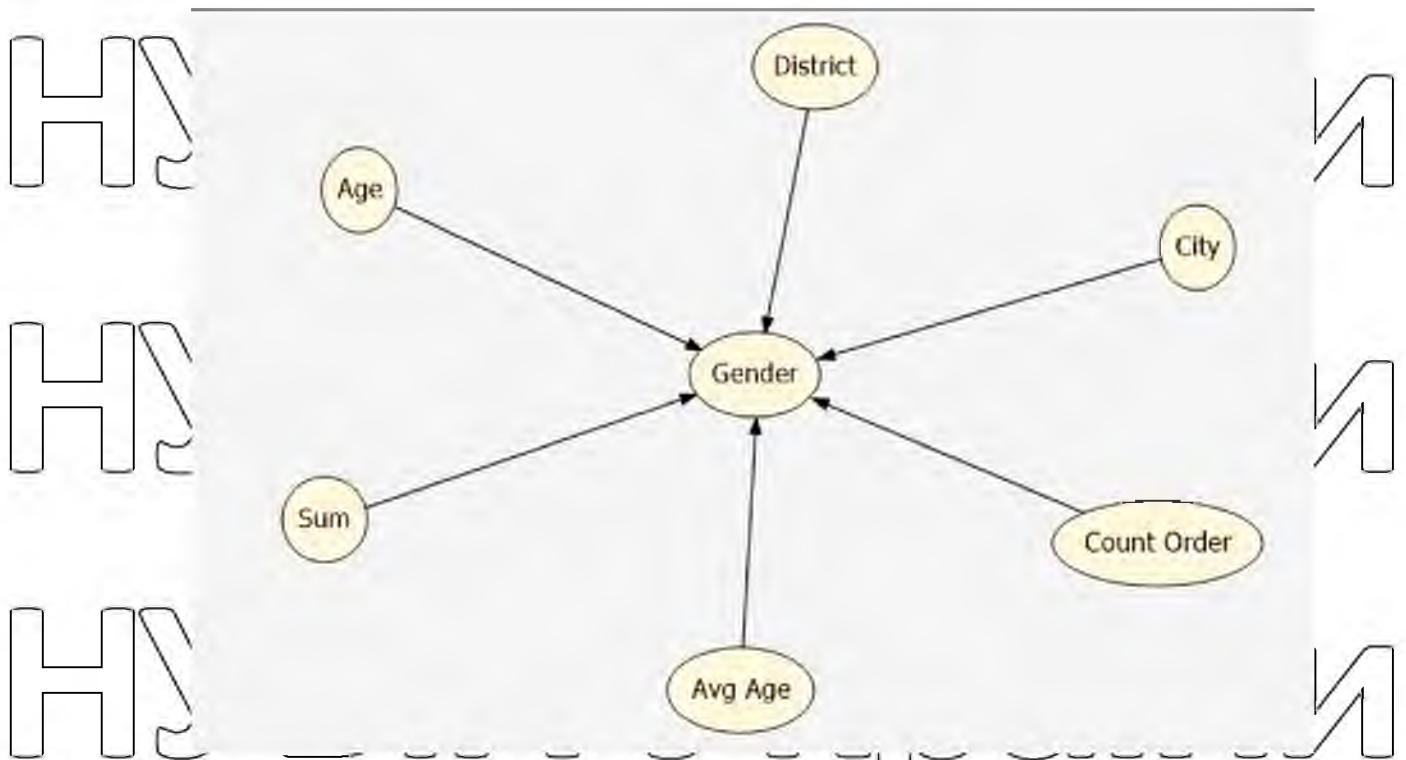


Рис. 46 Мережа зв'язків.

Представлена на рисунку 46, мережа дозволяє зрозуміти, що стать клієнта залежить від багатьох факторів:

- Жінки, віком 26-30 років (38%) замовляють зазвичай з міста Київ (35%), Ірпінь(30%) чи Боярка(16%).
- Чоловіки, віком до 40 років (84%) замовляють зазвичай з міста Київ (64%).

4.3 Дослідження використання методу асоціативних правил

Правила асоціації — це заснований на правилах метод машинного навчання для виявлення цікавих зв'язків між змінними у великих базах даних. Він призначений для визначення сильних правил, виявлених у базах даних, використовуючи певні показники цікавості. [6] У будь-якій транзакції з різноманітними елементами правила асоціації призначені для виявлення правил, які визначають, які чому певні елементи пов'язані.

Правила асоціації використовуються сьогодні в багатьох областях застосування, включаючи аналіз використання Інтернету, виявлення вторгнень

безперервне виробництво та біоінформатику. На відміну від аналізу послідовності, навчання правил асоціації зазвичай не враховує порядок елементів у транзакції або між транзакціями.

Алгоритм правила асоціації сам по собі складається з різних параметрів, які можуть ускладнити виконання тим, хто не має певного досвіду в аналізі даних, з багатьма правилами, які важко зрозуміти. [7]

Незважаючи на це, навчання правил асоціації є чудовою системою для прогнозування поведінки у взаємозв'язках даних. Це робить його гідною уваги технікою класифікації або виявлення закономірностей у даних під час реалізації методів машинного навчання.

Правила асоціації створюються шляхом пошуку даних для поширених моделей «якщо-то», а також за допомогою певного критерію в розділі «Підтримка та впевненість» для визначення найважливіших зв'язків. Підтримка — це доказ того, як часто той чи інший елемент з'являється у наведених даних, оскільки впевненість визначається тим, скільки разів твердження якщо-то визнаються істинними. Однак є третій критерій, який можна використовувати, він називається Lift, і його можна використовувати для порівняння очікуваної впевненості та фактичної впевненості. Підвищення покаже, скільки разів очікується, що твердження якщо-то буде істинним.

Правила асоціації створені для обчислення з наборів елементів, які створені двома або більше елементами. Якби правила були побудовані на основі аналізу всіх можливих наборів елементів із даних, то правил було б стільки, що вони не мали б жодного значення. Ось чому правила асоціації зазвичай складаються з правил, які добре представлені даними.

Існує багато різних методів аналізу даних, які можна використовувати для пошуку певної аналітики та результатів, наприклад, є класифікаційний аналіз, кластеризаційний аналіз та регресійний аналіз. [8] Залежно від того, що ви шукаєте зі своїми даними, залежить від того, яку техніку ви повинні використовувати. Правила асоціації в основному використовуються для пошуку аналітики та прогнозування поведінки клієнтів. Для класифікаційного аналізу,

швидше за все, він буде використаний для запитань, прийняття рішень і прогнозування поведінки. Кластеризаційний аналіз в основному використовується, коли немає припущень щодо ймовірних зв'язків у даних. [9]

Регресійний аналіз Використовується, коли потрібно передбачити значення безперервної залежності від ряду незалежних змінних.

Переваги. Використання правил асоціації має багато переваг, як-от пошук шаблону, який допомагає зрозуміти кореляції та спільні випадки між наборами даних. Дуже хорошим прикладом із реального світу, який використовує правила

асоціації, може бути медицина. Медицина використовує правила асоціації, щоб

допомогти діагностувати пацієнтів. Під час діагностики пацієнтів необхідно враховувати багато змінних, оскільки багато захворювань мають схожі симптоми. Використовуючи правила Асоціації, лікарі можуть визначити умовну ймовірність захворювання, порівнюючи зв'язки симптомів з минулих випадків.

[10]

Недоліки. Однак правила асоціації також призводять до багатьох різних проблем, таких як пошук відповідних параметрів і порогових налаштувань для алгоритму майнінгу. Але є також недолік великої кількості виявлених правил.

Причина в тому, що це не гарантує, що правила будуть визнані відповідними, але

це також може призвести до низької продуктивності алгоритму. Іноді реалізовані алгоритми будуть містити занадто багато змінних і параметрів. Для тих, хто не має гарної концепції інтелекту даних, це може спричинити проблеми з його розумінням. [11]

На основі створеного раніше куба Restoran_DW створюємо структуру інтелектуального аналізу на основі алгоритму правил взаємозв'язків, що представлено на рисунку 47.

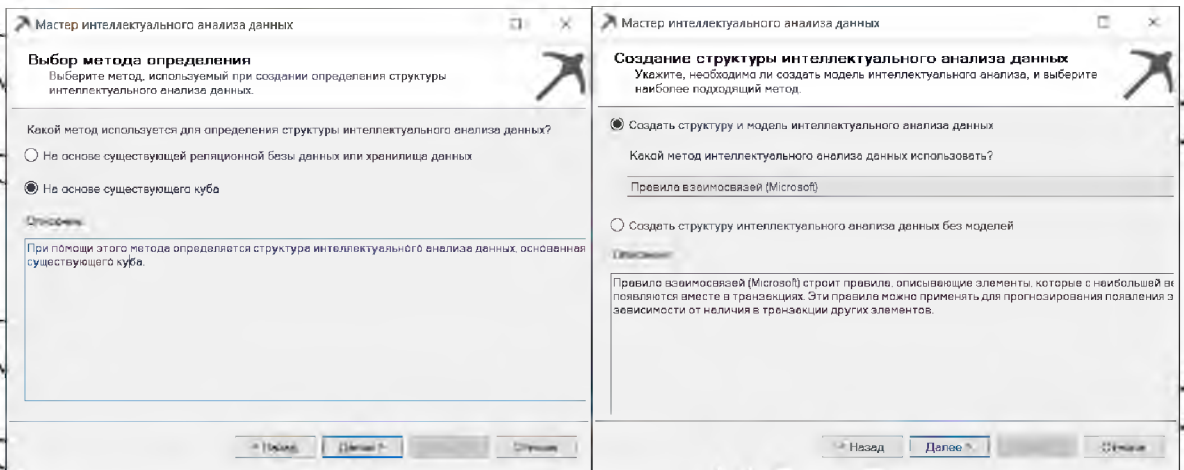


Рис. 47 Створення нової структури на основі алгоритму правил взаємозв'язків.

Далі визначаємо часовий вимір кубу – Order Dim та визначаємо ключове поле часу – Id_Order, що представлено на рисунку 48.

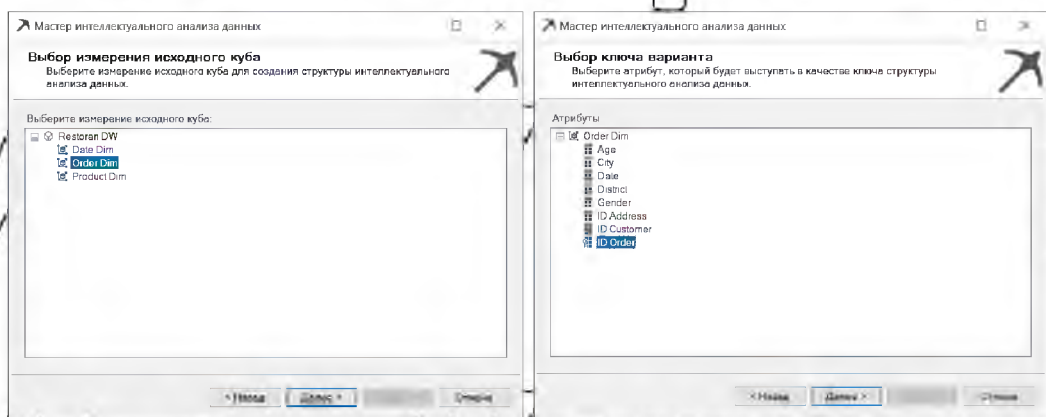


Рис. 48 Вибір ключового виміру куба та ключа структури.

Наступним кроком є вибір досліджуваних даних, а саме Age, Avg Age, City, Count Order, Date, District, Gender, Sum, та прогнозування значення City, як представлено на рисунку 49.

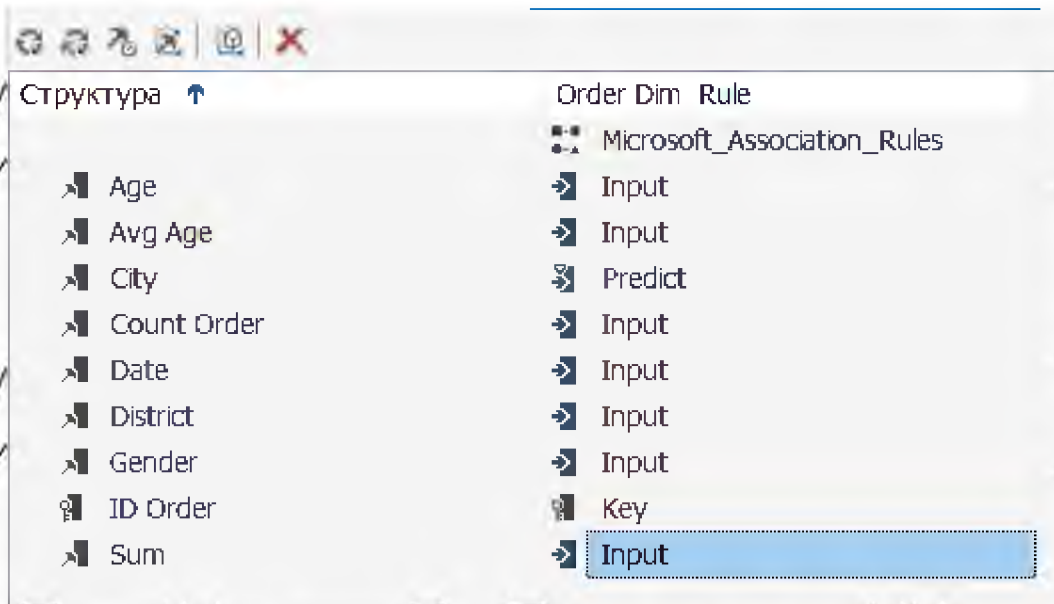


Рис. 49 Вибір полів, що аналізуємо та прогнозуємо для створеної структури.

Результат розгортання структури інтелектуального аналізу представлено в таблиці 1.

Таблиця 1. Результати розгортання

Імовірність	Важливість	Правило	Місто
1,000	1,305	Age = 33, Date = 28.07.2021-11.10.2021	Бровари
1,000	1,528	Age = 33, Date < 12.03.2021	Бровари
1,000	1,852	Age = 33, Gender = ж	Бровари
1,000	1,852	Age = 33, Count Order >= 9	Бровари
1,000	1,852	Age = 33, Avg Age >= 44	Бровари
1,000	1,074	Age = 25	Боярка
1,000	0,984	Age = 25, Date = 12.03.2021 - 21.05.2021	Боярка
1,000	1,074	Age = 25, Gender = ж	Боярка
1,000	1,087	Age = 35	Софіївська Борщягівка
1,000	1,087	Age = 35, Gender = ч	Софіївська Борщягівка
1,000	1,087	Age = 35, District = Київська	Софіївська Борщягівка
1,000	1,270	Age = 45	Боярка
1,000	0,984	Age = 45, Date = 21.05.2021-28.07.2021	Боярка
1,000	1,270	Age = 45, Gender = ж	Боярка

1,000	1,270	Age = 45, District = Київська	Боярка
1,000	0,969	Age = 31	Вишгород
1,000	0,969	Age = 31, Gender = ч	Вишгород
1,000	0,969	Age = 31, District = Київська	Вишгород
1,000	1,852	Age = 33	Бровари
1,000	1,100	Age = 27	Вишгород
1,000	1,100	Age = 27, Gender = ж	Вишгород

Загалом можна підсумувати, що більшість правила стосуються віку та статі.

4.4 Дослідження використання алгоритмів кластеризації

Кластеризація — це завдання поділу сукупності або точок даних на кілька груп таким чином, щоб точки даних в тих самих групах були більш схожими на інші точки даних у тій же групі, ніж в інших групах. Простіше кажучи, мета полягає в тому, щоб відокремити групи зі схожими ознаками та об'єднати їх у кластери. [12]

Загалом, кластеризацію можна розділити на дві підгрупи:

- Жорстка кластеризація: у жорсткій кластеризації кожна точка даних або повністю належить до кластера, або ні

- М'яка кластеризація: у м'якій кластеризації замість того, щоб помістити кожен точку даних в окремий кластер, призначається ймовірність або ймовірність того, що ця точка даних буде в цих кластерах.

Оскільки завдання кластеризації є суб'єктивним, то засобів, які можна використати для досягнення цієї мети, достатньо. Кожна методологія дотримується різного набору правил для визначення «подібності» між точками даних. Насправді відомо більше 100 алгоритмів кластеризації. Але деякі алгоритми широко використовуються, давайте розглянемо їх докладніше:

- Моделі підключення: як випливає з назви, ці моделі засновані на уявленні про те, що точки даних, розташовані ближче в просторі даних, демонструють більше схожості одна з одною, ніж точки даних, розташовані далі. Ці моделі можуть мати два підходи. У першому підході вони

НУБІП УКРАЇНИ

починають з класифікації всіх точок даних в окремі кластери, а потім їх агрегування в міру зменшення відстані. У другому підході всі точки даних класифікуються як один кластер, а потім розділяються у міру збільшення відстані. Крім того, вибір функції відстані є суб'єктивним. Ці моделі дуже легко інтерпретувати, але їм бракує масштабованості для обробки великих наборів даних. Прикладами цих моделей є алгоритм ієрархічної кластеризації та його варіанти.

НУБІП УКРАЇНИ

- Центроїдні моделі: це ітераційні алгоритми кластеризації, в яких поняття подібності впливає з близькості точки даних до центроїда кластерів. Алгоритм кластеризації K-Means є популярним алгоритмом, який відноситься до цієї категорії. У цих моделях немає кластерів, необхідних в кінці, необхідно вказати заздалегідь, тому важливо мати попередні знання про набір даних. Ці моделі виконуються ітераційно, щоб знайти локальний оптимум.

НУБІП УКРАЇНИ

- Моделі розподілу: ці моделі кластеризації засновані на уявленні про те, наскільки ймовірно, що всі точки даних в кластері належать до одного розподілу (наприклад, нормальний, гаусів). Ці моделі часто страждають від переобтяжки. Популярним прикладом цих моделей є алгоритм максимізації очікування, який використовує багатомірні нормальні розподіли.

НУБІП УКРАЇНИ

- Моделі щільності: ці моделі здійснюють пошук у просторі даних областей із різною щільністю точок даних у просторі даних. Він ізолює різні регіони з різною щільністю та призначає точки даних у межах цих регіонів в одному кластері. Популярними прикладами моделей щільності є DBSCAN і OPTICS.

НУБІП УКРАЇНИ

На основі створеного раніше куба Restoran DW створюємо структуру інтелектуального аналізу на основі алгоритму кластеризації, що представлено на рисунку 50.

НУБІП УКРАЇНИ

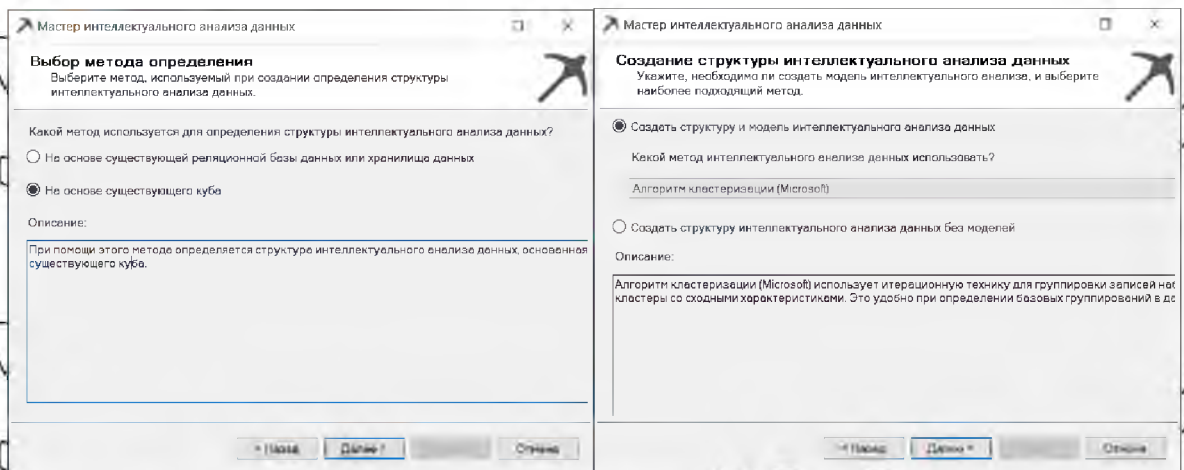


Рис. 50 Створення нової структури на основі алгоритму кластеризації.

Далі визначаємо часовий вимір кубу – Order Dim та визначаємо ключове поле часу – Id_Order, що представлено на рисунку 51.

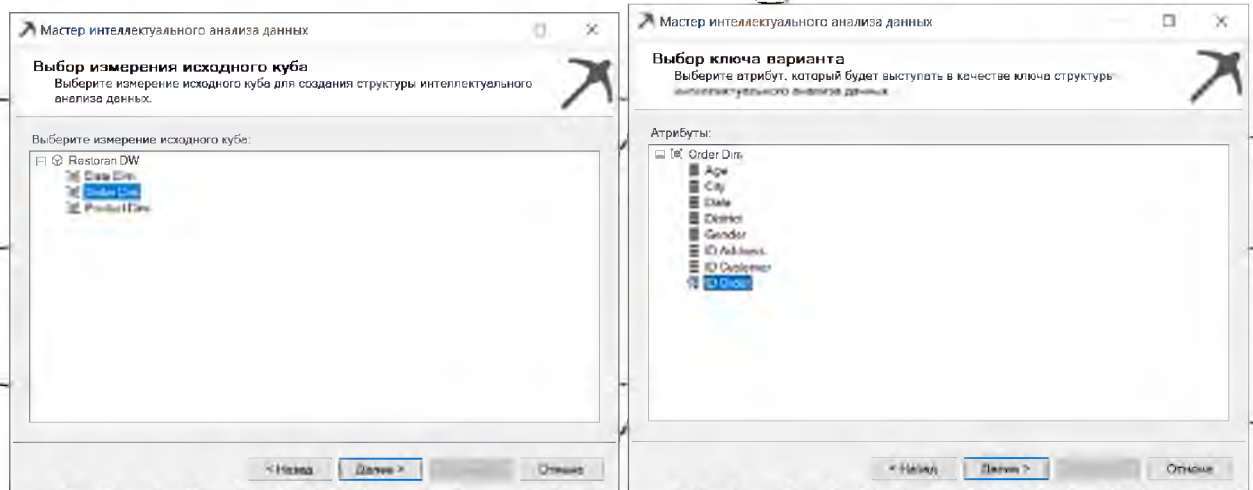


Рис. 51 Вибір ключового виміру куба та ключа структури.

Наступним кроком є вибір досліджуваних даних, а саме Age, Avg Age, City, Count Order, Date, District, Gender, Sum, як представлено на рисунку 52.

НУБІП України

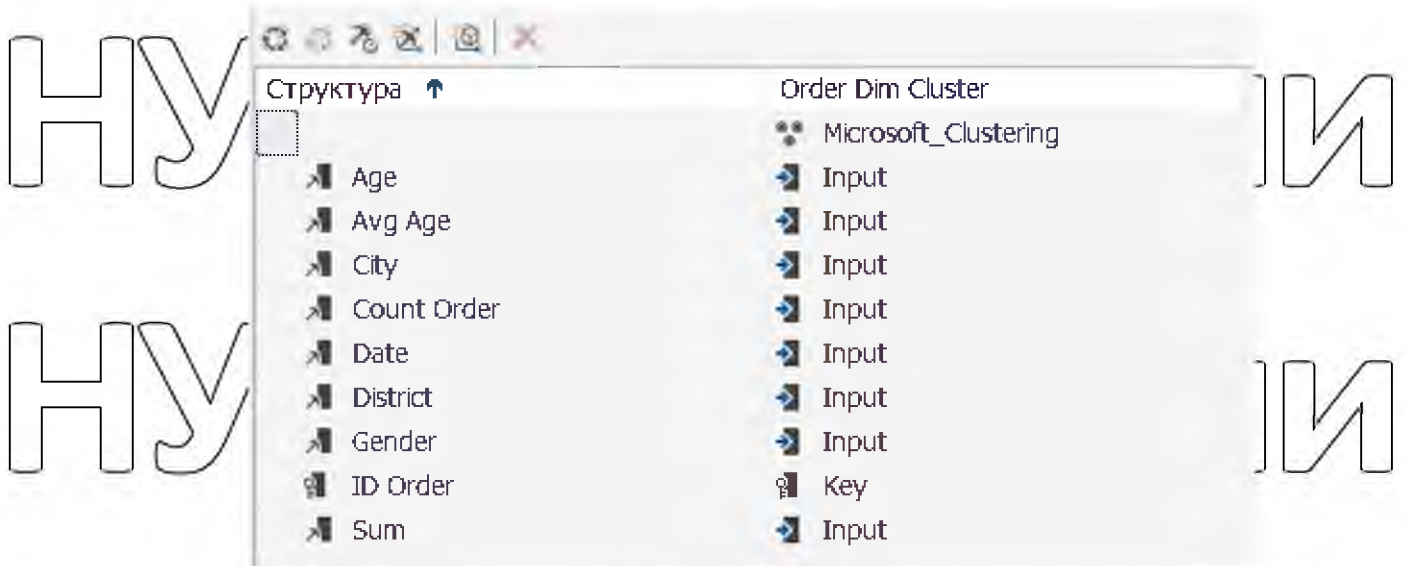


Рис. 52 Вибір полів, що аналізуємо та прогнозуємо для створення структури.

Результат розгортання структури інтелектуального аналізу представлено на рисунку 53.

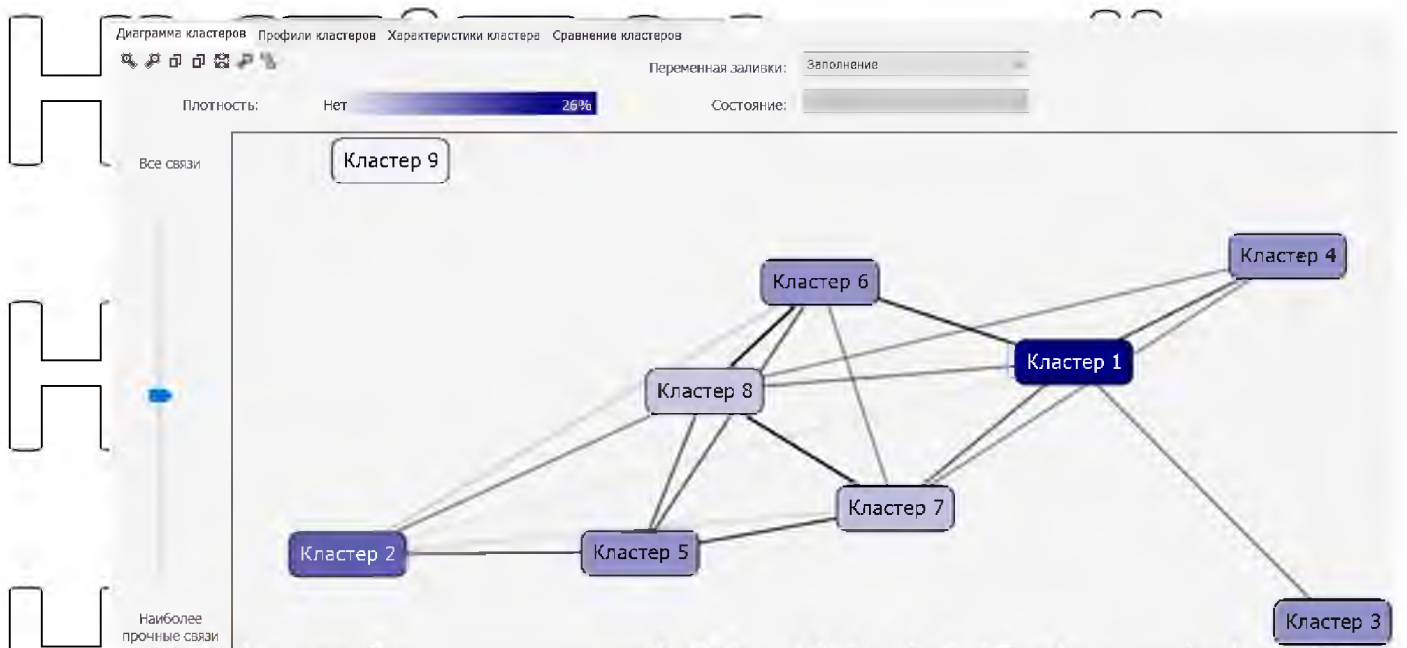


Рис. 53 Кластерна діаграма.

Аналіз кластерів:

- Кластер 1: містить замовлення клієнтів, віком до 25 років, з міста Київ, переважну більшість яких складають чоловіки (80%);
- Кластер 2: містить замовлення клієнтів, жіночої статі, віком до 36

років, з міст Ірпінь (95%) та Вишгород (5%);

- Кластер 3: містить замовлення клієнтів, чоловічої статі, віком до 36 років, з міст Софіївська Борщагівка чи Вишгород;

- Кластер 4: містить замовлення клієнтів, чоловічої статі, віком від 30 до 52 років, з міст Софіївська Борщагівка(42%) чи Київ(58%);

років, з міст Боярка(31%) чи Вишгород (65%);

- Кластер 6: містить замовлення клієнтів, жіночої статі, віком до 36 років, з міст Київ (66%) чи Бровари (31%);

• Кластери 7-9: містять дані що мають певні відхилення.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВИСНОВКИ

У ході виконання магістерської роботи було проведено аналіз предметної області, визначено основні елементи та модулі, які в подальшому необхідні для створення експертної системи, змодельована та розроблена інформаційна система у вигляді веб-системи.

Система була розроблена користуючись мовою розмітки HTML/CSS, серверною мовою програмування PHP, бібліотекою JQuery та додатковими CSS фреймворками Bootstrap та UIKit.

Результатом даної роботи є готова експертна система на основі маркетингових досліджень ринку ресторанних послуг яка дозволить покращити конверсію продажів тим самим покращити прибутковість та впізнаваність.

Також система дозволяє автоматизувати прийом замовлень у ресторані при двох варіантів замовлення (замовлення у ресторані та замовлення доставки), автоматизувати роботу працівників ресторану та надає змогу вести можливість обліку продажу страв, отримання фідбеку від клієнтів та деякі маркетингові можливості.

СПИСК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського" [Електронний ресурс]:
«ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ» - Режим доступу:
https://ela.kpi.ua/bitstream/123456789/33651/1/PIS_KL.pdf
2. Тернопільський державний економічний університет [Електронний ресурс]: «Організація баз даних» - Режим доступу:
<http://dspace.wunu.edu.ua/jspui/bitstream/316497/9109/1/%D0%BE%D0%BF%D0%BE%D1%80%D0%BD%D0%B8%D0%B9%20%D0%BA%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82%20%D0%BB%D0%B5%D0%BA%D1%86%D1%96%D0%B9.pdf>
3. Database system concepts / Silberschatz, Abraham, Korth, Henry F.; Sudarshan [Електронний ресурс]: «Видавничі бази даних» - Режим доступу:
<https://dspace.nau.edu.ua/bitstream/NAU/54872/1/%D0%A0%D0%9F%D0%9D%20%D0%92%D0%B8%D0%B4%D0%B0%D0%B2%D0%BD%D0%B8%D1%87%D1%96%20%D0%91%D0%94%2022.pdf>
4. ScikitLearn / [Електронний ресурс]: «Decision Trees» - Режим доступу:
<https://scikit-learn.org/stable/modules/tree.html>
5. ScikitLearn [Електронний ресурс]: «Naive Bayes» - Режим доступу:
https://scikit-learn.org/stable/modules/naive_bayes.html
6. Piatetsky-Shapiro, Gregory (1991), Discovery, analysis, and presentation of strong rules, in Piatetsky-Shapiro, Gregory; and Frawley, William J.; eds., Knowledge Discovery in Databases, AAAI/MIT Press, Cambridge, MA.
7. Garcia, Enrique (2007) / [Електронний ресурс]: «Drawbacks and solutions of applying association rule mining in learning management systems» - Режим доступу:
<https://sci2s.ugr.es/keel/pdf/specific/congreso/3-associationrules-Final.pdf>
8. Precisely / [Електронний ресурс]: «Техніки аналізу даних: 5 найкращих для розгляду» - Режим доступу:
<https://www.precisely.com/blog/datagovernance/top-5-data-mining-techniques>

9. Talend - A Leader in Data Integration & Data Integrity / [Електронний ресурс]: «16 Data Mining Techniques. The Complete List - Talend» - Режим доступу: <https://www.talend.com/resources/data-mining-techniques/>

10. SearchBusinessAnalytics / [Електронний ресурс]: «What are Association Rules in Data Mining (Association Rule Mining)?» - Режим доступу: <https://searchbusinessanalytics.techtarget.com/definition/association-rules-in-data-mining>

11. ResearchGate / [Електронний ресурс]: «Drawbacks and solutions of applying association rule mining in learning management systems» - Режим доступу: <https://www.researchgate.net/publication/289657906>

12. Analyticsvidhya / [Електронний ресурс]: «An Introduction to Clustering and different methods of clustering» - Режим доступу: <https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/#:~:text=Soft%20Clustering%3A%20In%20soft%20clustering,clusters%20of%20the%20retail%20store.>

13. ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА / [Електронний ресурс]: «Експертні системи» - Режим доступу: https://financial.lnu.edu.ua/wp-content/uploads/2019/09/ES_konspekt-tekst_v.pdf

НУБІЛ ПІЯНІ
України

НУБІЛ ПІЯНІ
України

НУБІЛ ПІЯНІ
України

НУБІЛ ПІЯНІ
України

НУБІЛ ПІЯНІ
України

НУБІЛ ПІЯНІ
України

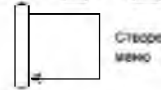
НУБІЛ ПІЯНІ
України

НУБІП України

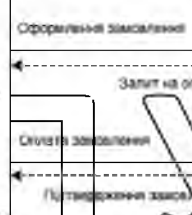
2022

Сторінка 1

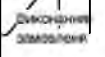
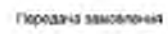
НУБІП України



НУБІП України



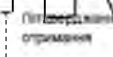
НУБІП України



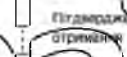
НУБІП України



НУБІП України



НУБІП України



НУБІП України

Сторінка 2

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

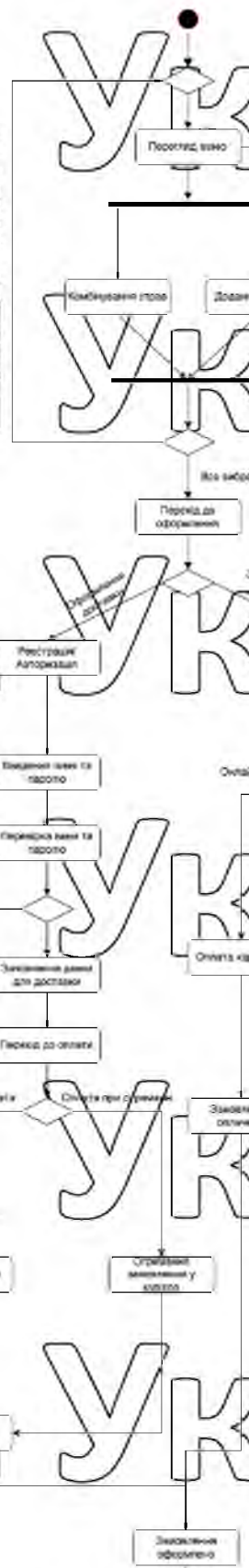
НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України



ДОДАТОК Б
НУБІП України

НУБІП України

НУБІП України

СТВОРЕННЯ БАЗИ ДАНИХ

НУБІП України

НУБІП України

НУБІП України

Сторінок 5
НУБІП України

```
CREATE TABLE `oc_category` (  
  `category_id` int(11) NOT NULL AUTO_INCREMENT,  
  `image` varchar(255) DEFAULT NULL,  
  `parent_id` int(11) NOT NULL DEFAULT '0',  
  `top` tinyint(1) NOT NULL,  
  `status` tinyint(1) NOT NULL,  
  `date_added` datetime NOT NULL,  
  `date_modified` datetime NOT NULL,  
  PRIMARY KEY (`category_id`),  
  KEY `parent_id` (`parent_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_category_description` (  
  `category_id` int(11) NOT NULL,  
  `language_id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `description` text NOT NULL,  
  `meta_title` varchar(255) NOT NULL,  
  `meta_description` varchar(255) NOT NULL,  
  `meta_keyword` varchar(255) NOT NULL,  
  `meta_h1` varchar(255) NOT NULL,  
  PRIMARY KEY (`category_id`, `language_id`),  
  KEY `name` (`name`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_admin` (  
  `admin_id` int(11) NOT NULL AUTO_INCREMENT,  
  `admin_group_id` int(11) NOT NULL,  
  `username` varchar(20) NOT NULL,  
  `password` varchar(40) NOT NULL,  
  `firstname` varchar(32) NOT NULL,  
  `lastname` varchar(32) NOT NULL,  
  `email` varchar(96) NOT NULL,
```

```
status` tinyint(1) NOT NULL,  
PRIMARY KEY (`admin_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_product` (  
`product_id` int(11) NOT NULL AUTO_INCREMENT,  
`model` varchar(64) NOT NULL,  
`quantity` int(4) NOT NULL DEFAULT '0',  
`stock_status_id` int(11) NOT NULL,  
`image` varchar(255) DEFAULT NULL,  
`shipping` tinyint(1) NOT NULL DEFAULT '1',  
`price` decimal(15,4) NOT NULL DEFAULT '0.0000',  
`points` int(8) NOT NULL DEFAULT '0',  
`weight` decimal(15,8) NOT NULL DEFAULT '0.00000000',  
`weight_class_id` int(11) NOT NULL DEFAULT '0',  
`minimum` int(11) NOT NULL DEFAULT '1',  
`sort_order` int(11) NOT NULL DEFAULT '0',  
`status` tinyint(1) NOT NULL DEFAULT '0',  
`date_added` datetime NOT NULL,  
`date_modified` datetime NOT NULL,  
PRIMARY KEY (`product_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_product_attribute` (  
`product_id` int(11) NOT NULL,  
`attribute_id` int(11) NOT NULL,  
`language_id` int(11) NOT NULL,  
`text` text NOT NULL,  
PRIMARY KEY (`product_id`, `attribute_id`, `language_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_product_attribute` (  
`product_id` int(11) NOT NULL,  
`attribute_id` int(11) NOT NULL,  
`language_id` int(11) NOT NULL,  
`text` text NOT NULL,  
PRIMARY KEY (`product_id`, `attribute_id`, `language_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_product_attribute` (  
`product_id` int(11) NOT NULL,  
`attribute_id` int(11) NOT NULL,  
`language_id` int(11) NOT NULL,  
`text` text NOT NULL,  
PRIMARY KEY (`product_id`, `attribute_id`, `language_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_product_attribute` (  
`product_id` int(11) NOT NULL,  
`attribute_id` int(11) NOT NULL,  
`language_id` int(11) NOT NULL,  
`text` text NOT NULL,  
PRIMARY KEY (`product_id`, `attribute_id`, `language_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_product_attribute` (  
`product_id` int(11) NOT NULL,  
`attribute_id` int(11) NOT NULL,  
`language_id` int(11) NOT NULL,  
`text` text NOT NULL,  
PRIMARY KEY (`product_id`, `attribute_id`, `language_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_product_description` (
```

```
  `product_id` int(11) NOT NULL,
```

```
  `language_id` int(11) NOT NULL,
```

```
  `name` varchar(255) NOT NULL,
```

```
  `description` text NOT NULL,
```

```
  `tag` text NOT NULL,
```

```
  `meta_title` varchar(255) NOT NULL,
```

```
  `meta_description` varchar(255) NOT NULL,
```

```
  `meta_keyword` varchar(255) NOT NULL,
```

```
  `meta_h1` varchar(255) NOT NULL,
```

```
  PRIMARY KEY (`product_id`, `language_id`),
```

```
  KEY `name` (`name`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE IF NOT EXISTS categories `oc_order_status` ( `order_status_id` int(11)
```

```
NOT NULL AUTO_INCREMENT,
```

```
  `language_id` int(11) NOT NULL,
```

```
  `name` varchar(32) NOT NULL,
```

```
  PRIMARY KEY (`order_status_id`, `language_id`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_product_option` (
```

```
  `product_option_id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `product_id` int(11) NOT NULL,
```

```
  `option_id` int(11) NOT NULL,
```

```
  `value` text NOT NULL,
```

```
  `required` tinyint(1) NOT NULL,
```

```
  PRIMARY KEY (`product_option_id`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `oc_order` (
```



```

`order_id` int(11) NOT NULL AUTO_INCREMENT,
`customer_id` int(11) NOT NULL DEFAULT '0',
`customer_group_id` int(11) NOT NULL DEFAULT '0',
`firstname` varchar(32) NOT NULL,
`lastname` varchar(32) NOT NULL,
`email` varchar(96) NOT NULL,
`telephone` varchar(32) NOT NULL,
`comment` text NOT NULL,
`total` decimal(15,4) NOT NULL DEFAULT '0.0000',
`order_status_id` int(11) NOT NULL DEFAULT '0',
`language_id` int(11) NOT NULL,
`date_added` datetime NOT NULL,
`date_modified` datetime NOT NULL,
PRIMARY KEY (`order_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci

```

Внесення умовно-постійної інформації

```

INSERT INTO `oc_category` (`category_id`, `language_id`, `image`, `parent_id`,
`main_menu`, `sort_order`, `status`, `date_added`) VALUES
(1, 1, 'images/category/burgers.jpg', 0, 1, 3, 1, '2021-03-10 03:08:25'),
(2, 1, 'images/category/sushi.jpg', 13, 1, 2, 1, '2021-03-11 05:11:38'),
(3, 1, 'images/category/pizza.jpg', 0, 1, 1, 1, '2021-03-12 20:32:45'),
(4, 1, 'images/category/soup.jpg', 0, 1, 2, 1, '2021-03-12 18:49:33'),
(5, 1, 'images/category/pasta.jpg', 17, 1, 0, 1, '2021-03-13 12:40:21');
INSERT INTO `oc_category_description` (`category_id`, `language_id`, `name`,
`description`, `meta_title`, `meta_description`, `meta_keyword`, `meta_htl`) VALUES
(1, 1, 'Бургери', '', 'Бургери - замовити у ресторані Skymod Eats', '', '', ''),
(2, 1, 'Суши', '', 'Суши - замовити у ресторані Skymod Eats', '', '', ''),
(3, 1, 'Піцца', '', 'Піцца - замовити у ресторані Skymod Eats', '', '', ''),
(4, 1, 'Супи', '', 'Супи - замовити у ресторані Skymod Eats', '', '', ''),

```

НУБІП України
 (5, 1, 'Паста', 'Паста - замовити у ресторані Skymod Eats', '"', '"');

INSERT INTO `oc_order_status` (`order_status_id`, `language_id`, `name`) VALUES

НУБІП України
 (1, 1, 'В обробці'),
 (2, 1, 'Доставлено'),
 (3, 1, 'Скасовано'),
 (4, 1, 'Угода завершена'),

(5, 1, 'Скасування та анулювання'),

(6, 1, 'Повністю змінений'),

НУБІП України
 (7, 1, 'Опрацьовано'),
 (8, 1, 'Час оплати минув');

НУБІП України

НУБІП України

НУБІП України

НУБІП України