

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І  
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) \_\_\_\_\_ інформаційних технологій \_\_\_\_\_

**ПОГОДЖЕНО**  
Декан факультету  
інформаційних технологій

**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**  
Завідувач кафедри  
комп'ютерних наук

\_\_\_\_\_ / Ігор Болбот /

(підпис)

(ім'я прізвище)

« \_\_\_\_ » \_\_\_\_\_ 2025 р.

\_\_\_\_\_ / Белла Голуб /

(підпис)

(ім'я прізвище)

« \_\_\_\_ » \_\_\_\_\_ 2025 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему Програмне забезпечення аналітичної системи управління  
працевлаштуванням

Спеціальність 121 Інженерія програмного забезпечення  
(код і найменування)

Освітня програма Програмне забезпечення інформаційних систем  
(назва)

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

**Гарант освітньої програми**

\_\_\_\_\_ доц.к.ф.-м.н.

Науковий ступень та вчене звання

підпис

\_\_\_\_\_ / Віктор Кириченко /

ім'я прізвище

**Керівник магістерської кваліфікаційної роботи**

\_\_\_\_\_ доц.к.т.н.

Науковий ступень та вчене звання

підпис

\_\_\_\_\_ / Белла Голуб /

ім'я прізвище

**Виконав** \_\_\_\_\_

\_\_\_\_\_ / Дмитро Рудой /

підпис

ім'я прізвище

**КИЇВ – 2025**

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук  
доцент, к.т.н. Белла Голуб  
(науковий ступінь, вчене звання) (підпис) (ім'я прізвище)  
" 01 " листопада 2024 року

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Рудому Дмитру Ігоровичу  
(прізвище, ім'я, по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»  
(код і назва)

Освітня програма Програмне забезпечення інформаційних систем  
(назва)

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Програмне забезпечення аналітичної системи управління працевлаштуванням

затверджена наказом ректора НУБіП України від " 01 " листопада 2024р. №1963 «С»

Термін подання завершеної роботи на кафедру 28.11.2025

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи: структурований набір даних, зібраний з веб-порталу працевлаштування work.ua. Датасет містить детальну інформацію про вакансії, включаючи їх назву, рівень заробітної плати, вимоги до кандидатів, перелік необхідних навичок, географічне розташування та галузеві категорії.

Перелік питань, що підлягають дослідженню:

1. Ідентифікація та аналіз ключових викликів на сучасному ринку праці (наприклад, невідповідність навичок, дисбаланс попиту та пропозиції), які можуть бути досліджені методами інтелектуального аналізу даних.

2. Дослідження та обґрунтування застосування технологій OLAP-кубів та алгоритмів Data Mining для виявлення прихованих тенденцій, кластеризації вакансій та прогнозування змін на ринку праці.

3. Виявлення та оцінка впливу ключових атрибутів вакансії (зарплата, тип зайнятості, регіон, набір технологій) на тривалість пошуку кандидата та загальну привабливість пропозиції для потенційних співробітників.

4. Аналіз часових закономірностей та патернів, що визначають появу нових вакансій у різних секторах, а також факторів, які корелюють з успішним заповненням вакантних посад.

Перелік графічного матеріалу (за потреби)

Дата видачі завдання " 01 " листопада 2024 р.

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_

Белла Голуб

(підпис)

(ім'я прізвище)

Завдання прийняв до виконання \_\_\_\_\_

Дмитро Рудой

(підпис)

(ім'я прізвище здобувача)

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 ОПИС ПРОЦЕСІВ У СФЕРІ УПРАВЛІННЯ ПЕРСОНАЛОМ	8
1.2 АНАЛІЗ НАЯВНИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ	11
1.3 ІДЕНТИФІКАЦІЯ КЛЮЧОВИХ ВИКЛИКІВ НА РИНКУ ПРАЦІ	14
1.4 ПОСТАНОВКА ЗАВДАННЯ МАГІСТЕРСЬКОГО ДОСЛІДЖЕННЯ	16
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ АНАЛІТИЧНИХ ПРОЦЕСІВ	19
2.1 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ	19
2.2 ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ	23
3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ АНАЛІТИЧНОЇ СИСТЕМИ	25
3.1 КОНЦЕПЦІЯ АРХІТЕКТУРИ	25
3.2 КОМПОНЕНТ ЗБОРУ ДАНИХ	29
3.3 СХОВИЩЕ ДАНИХ ТА ETL-ПРОЦЕСИ	31
3.4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ АНАЛІТИКИ	40
4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ РИНКУ ПРАЦІ	49
4.1 АНАЛІЗ КЛЮЧОВИХ ВИКЛИКІВ РИНКУ ПРАЦІ	49
4.2 ДОСЛІДЖЕННЯ ПРИХОВАНИХ ТЕНДЕНЦІЙ, КЛАСТЕРИЗАЦІЯ ТА ПРОГНОЗУВАННЯ	53
4.3 ОЦІНКА ВПЛИВУ АТРИБУТІВ НА ТРИВАЛІСТЬ ВАКАНСІЙ	55
4.4 АНАЛІЗ ЧАСОВИХ ЗАКОНОМІРНОСТЕЙ ЗА СЕКТОРАМИ	57
4.5 ІНТЕРАКТИВНА ВІЗУАЛІЗАЦІЯ	58
ВИСНОВКИ	64
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	67

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	–	Інтерфейс прикладного програмування (Application Programming Interface)
ATS	–	Система відстеження кандидатів (Applicant Tracking System)
BI	–	Бізнес-аналітика (Business Intelligence)
DDL	–	Мова визначення даних (Data Definition Language)
DFD	–	Діаграма потоків даних (Data Flow Diagram)
DWH	–	Сховище даних (Data Warehouse)
ETL	–	Процес вилучення, трансформації та завантаження (Extract, Transform, Load)
FDD	–	Діаграма функціональної декомпозиції (Feature Decomposition Diagram)
HR	–	Управління людськими ресурсами (Human Resources)
KPI	–	Ключовий показник ефективності (Key Performance Indicator)
MDX	–	Багатовимірні вирази (Multidimensional Expressions)
OLAP	–	Оперативна аналітична обробка (Online Analytical Processing)
SSAS	–	Служби аналітики SQL Server (SQL Server Analysis Services)
SSIS	–	Служби інтеграції SQL Server (SQL Server Integration Services)
SSRS	–	Служби звітності SQL Server (SQL Server Reporting Services)
SQL	–	Мова структурованих запитів (Structured Query Language)

## ВСТУП

**Актуальність теми.** Сучасний ринок праці характеризується експоненційним зростанням обсягів даних про вакансії, компанії, галузі та навички кандидатів. Ручний аналіз цієї інформації є неефективним, трудомістким і не дозволяє виявляти приховані тенденції та нетривіальні закономірності. В умовах високої конкуренції за кваліфікованих фахівців, компанії потребують інструментів, що дозволяють оптимізувати кадрову політику, прогнозувати зміни на ринку та приймати науково обґрунтовані рішення в сфері управління персоналом. У цьому контексті виникає гостра необхідність у створенні автоматизованих аналітичних систем на базі сховищ даних (Data Warehouse) та застосуванні алгоритмів інтелектуального аналізу даних (Data Mining).

Створення такої системи є актуальним та перспективним напрямком, оскільки дозволяє перейти від інтуїтивних методів до науково обґрунтованих рішень у сфері HR, що сприяє підвищенню ефективності процесів пошуку роботи, управління людськими ресурсами та стратегічного планування.

**Об'єктом дослідження** є процес управління працевлаштуванням, що включає пошук вакансій, аналіз ринку праці та відповідні дані, зібрані з веб-порталу work.ua.

**Предметом дослідження** є методи та засоби побудови аналітичної системи на основі сховища даних та застосування алгоритмів Data Mining.

**Мета дослідження** - розробити та реалізувати систему підтримки прийняття рішень для аналізу ринку праці, використовуючи технології оперативного(OLAP) та інтелектуального аналізу даних(Data Mining).

Для досягнення поставленої мети були виконані наступні **завдання**.

- Системний аналіз предметної області управління працевлаштуванням та сучасних підходів до аналізу ринку праці.

- Проектування архітектури сховища даних та реалізація ETL-процесів для завантаження, трансформації та очищення даних.
- Дослідження та обґрунтування застосування алгоритмів Data Mining для вирішення задач кластеризації вакансій, класифікації попиту та прогнозування часових рядів.
- Проведено експериментальні дослідження, проаналізовано отримані результати та візуалізовано виявлені тенденції та закономірності.

**Методи дослідження.** Для вирішення поставлених завдань у роботі використано методи системного аналізу, теорії баз даних, технології сховищ даних (ETL, OLAP), а також методи інтелектуального аналізу даних, зокрема: алгоритм кластеризації K-Means, класифікатор Наївного Байєса, модель регресії Random Forest та модель прогнозування часових рядів Exponential Smoothing.

**Наукова новизна** полягає у розробці аналітичної системи, що поєднує технології сховищ даних та Data Mining для аналізу специфічних факторів українського ринку праці, зокрема тривалості закриття вакансій.

**Практичне значення отриманих результатів.** Розроблена система та результати аналізу надають науково обґрунтовану інформацію для HR-фахівців та аналітиків ринку, дозволяючи краще розуміти структуру ринку та прогнозувати його динаміку.

**Апробація результатів.** Основні результати роботи були представлені на:

1. Конкурс студентських наукових робіт, 25 березня 2025, м. Київ, <https://nubip.edu.ua/node/162723>;
2. Теоретичні та прикладні аспекти розробки комп'ютерних систем, 24 квітня 2025, м. Київ, <http://econference.nubip.edu.ua/index.php/taacsd/2025>;
3. XVI міжнародна науково-практичній конференції студентів, аспірантів та молодих вчених «інформаційні технології: економіка, техніка, освіта», 28 жовтня 2025, м. Київ, <http://econference.nubip.edu.ua/index.php/itete/XVI/index>.

**Структура та обсяг роботи.** Магістерська робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 67 сторінок, містить 37 рисунків та 4 таблиці. Список використаних джерел налічує 25 найменувань.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Опис процесів у сфері управління персоналом

Предметна область управління працевлаштуванням охоплює складну та екосистему взаємодій між ключовими учасниками ринку: роботодавцями, що прагнуть залучити кваліфікованих спеціалістів для досягнення своїх бізнес-цілей, та кандидатами, які перебувають в активному пошуку роботи або відкриті до нових кар'єрних можливостей. Цей процес є вкрай динамічним, оскільки на нього постійно впливають економічні цикли, соціальні зміни, технологічний прогрес та глобальні події. Ефективне управління цим процесом вимагає глибокого розуміння його внутрішньої структури та зовнішніх факторів впливу.

**1.1.1 Загальна характеристика сучасного ринку праці.** Сучасний ринок праці в Україні характеризується значними структурними змінами, що призвели до формування так званого "ринку роботодавця". На відміну від попередніх років, коли спостерігався дефіцит кваліфікованих кадрів, сьогоднішня ситуація відзначається надзвичайно високою конкуренцією серед шукачів. На популярні позиції, особливо на початковому та середньому рівнях у сфері ІТ, кількість відгуків на одну вакансію може сягати сотень. Це створює нові виклики як для кандидатів, яким стає все важче виділитися, так і для роботодавців, які стикаються з проблемою ефективної фільтрації величезної кількості резюме.

По-друге, спостерігається **постійне зростання вимог до навичок спеціалістів**. Технологічний стек, актуальний сьогодні, може застаріти вже за кілька років. Це створює проблему "розриву в навичках" (Skills Gap), коли система освіти не встигає за потребами ринку, а компанії змушені інвестувати у перекваліфікацію та навчання персоналу.

По-третє, значний вплив на ринок мають **глобальні тренди та локальні виклики**. Пандемія COVID-19 нормалізувала дистанційну роботу, що кардинально змінило географію пошуку талантів. Компанії отримали доступ до

спеціалістів з будь-якого міста, а кандидати – можливість працювати на міжнародні компанії, не змінюючи місце проживання. Водночас, поточні соціально-політичні події в Україні призвели до формування значного попиту на специфічні, зокрема військові, спеціальності. Цей фактор суттєво вплинув на структуру вакансій та рівень заробітних плат, що буде детально проаналізовано в наступних розділах.

**1.1.2 Життєвий цикл рекрутингу в компаніях.** Процес управління працевлаштуванням з боку компанії є циклічним та багатоетапним. Його можна представити у вигляді воронки, де на кожному етапі відбувається відбір та уточнення (рис. 1.1).

## Етапи воронки рекрутингу



Рис. 1.1 Воронка процесу рекрутингу

Зазвичай основні фази цього процесу включають:

1. **Планування потреби у персоналі.** На основі стратегічних цілей компанії HR-відділ та керівники підрозділів визначають, які саме ролі необхідно закрити для досягнення бізнес-результатів.

2. **Формування профілю вакансії.** Створюється детальний опис посади ("job description"), що є ключовим документом. Він включає перелік обов'язків, вимоги до кандидата (технічні навички, soft skills, досвід), рівень заробітної плати та опис умов праці.

3. **Пошук кандидатів (Sourcing).** Активна фаза, що включає розміщення вакансій на спеціалізованих веб-порталах (job boards), прямий пошук ("headhunting") у професійних мережах, роботу з кадровими агентствами та внутрішніми реферальними програмами.

4. **Відбір та оцінка кандидатів (Screening & Assessment).** Цей етап складається з кількох підрівнів: аналіз резюме, телефонний скринінг, технічна співбесіда, співбесіда з HR-менеджером для оцінки soft skills та відповідності корпоративній культурі, а також виконання тестових завдань.

5. **Прийняття рішення та пропозиція роботи (Offer).** Формування та надсилання офіційної пропозиції про працевлаштування (job offer) найуспішнішому кандидату.

6. **Адаптація нового співробітника (Onboarding).** Процес інтеграції нового члена команди в робочі процеси, знайомство з колективом та корпоративною культурою.

Кожен з цих етапів генерує значний обсяг даних, аналіз яких дозволяє оптимізувати процес, скоротити час на закриття вакансій та підвищити якість найму.

**1.1.3 Роль веб-порталів з працевлаштування.** Веб-портали, такі як work.ua [25], виступають центральною ланкою в екосистемі працевлаштування. Вони виконують функцію агрегатора, що з'єднує роботодавців та кандидатів, і є основним джерелом даних для аналізу ринку праці. Ці платформи надають структуровану інформацію про вакансії, яка є вихідними даними для даного

дослідження. Кожна публікація на такому порталі є записом, що містить багатий набір атрибутів, готових для збору та аналізу. Саме ці дані дозволяють відстежувати динаміку ринку, аналізувати вимоги до навичок та досліджувати зарплатні тенденції.

## **1.2 Аналіз наявних рішень та технологій**

Для ефективного управління процесами працевлаштування та аналізу ринку праці існує широкий спектр технологічних рішень, від спеціалізованих HR-платформ до універсальних інструментів для роботи з великими даними. Цей підрозділ розглядає ключові категорії існуючих систем та технологій, що є релевантними для даного дослідження.

Сучасні HR-системи можна умовно поділити на кілька категорій за їх функціональним призначенням:

- **Портали з працевлаштування та агрегатори вакансій.**

Такі платформи, як work.ua, rabota.ua, djinni.co та dou.ua, є первинними джерелами даних про вакансії. Вони надають базові аналітичні інструменти, наприклад, статистику зарплат за певними посадами або регіонами.

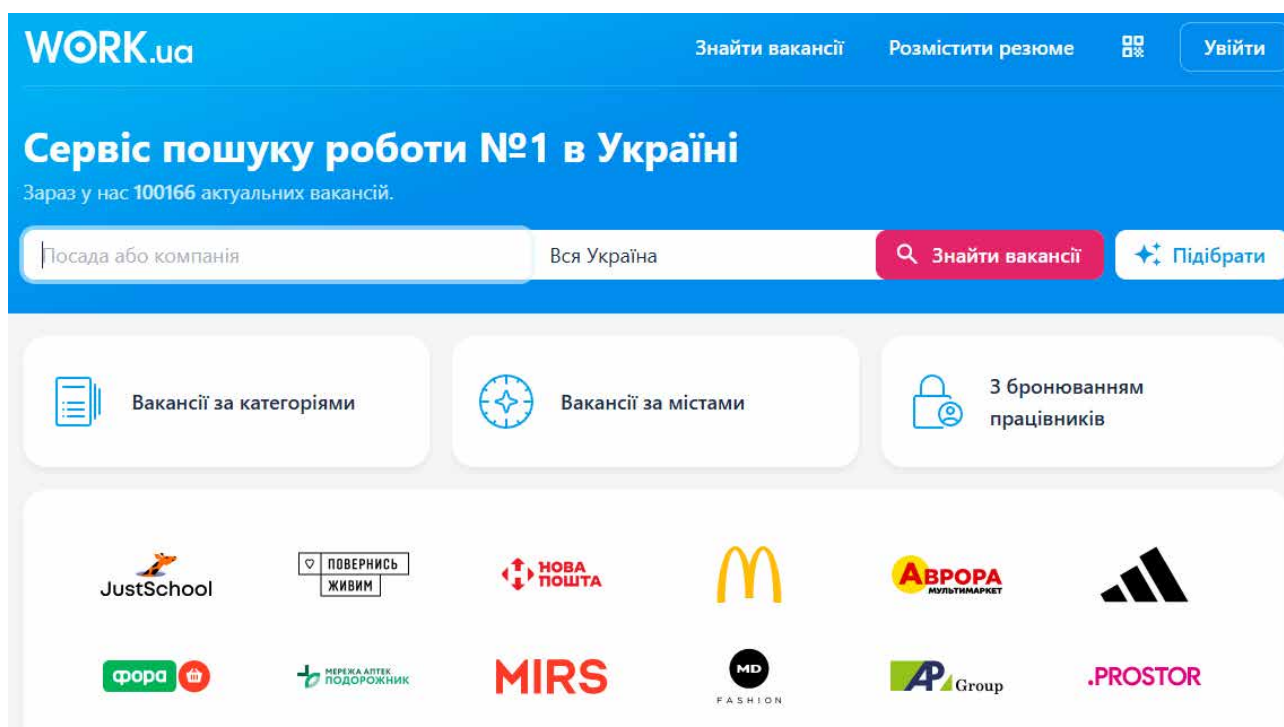


Рис. 1.3 Головна сторінка порталу Work.ua

## Вакансії на Djinni 9779

Для мене **Всі**

**A** Affiliate Marketing · 80 переглядів · 10 відгуків · щойно

### B2B Sales Manager

Тільки віддалено · Україна · 1 рік досвіду · B2 – Вище середнього

We are a fast-growing fintech company and an industry leader in innovative digital solutions for corporate clients. Our team is looking for a proactive Sales Manager with proven expertise in B2B sales. This role is focused on actively driving sales,... [Більше](#)

[Зберегти](#) [Сховати](#)

**M** Мрія · 158 переглядів · 17 відгуків · 1хв

### General QA Engineer

Гібридний формат роботи · Україна (Київ) · Продукт · 2 роки досвіду

Мрія — освітня екосистема для учнів, батьків/опікунів і вчителів, що надихає вчитися та допомагає знайти себе в сучасному світі. Це ініціатива Президента України Володимира Зеленського, яку реалізують Мінцифри та МОН за підтримки Програми EGAP, що... [Більше](#)

[Зберегти](#) [Сховати](#)

Шукати за посадою

Наприклад: Front-end engineer →

[Розширений пошук](#)

Знайти компанію

Наприклад: Djinni

Категорія

JavaScript / Front-End ▶ Fullstack  
 Java C# / .NET ▶ Python PHP ▶  
 Node.js iOS Android React Native  
 C / C++ / Embedded ▶ Flutter Golang  
 Ruby Scala Salesforce Rust Elixir  
 Kotlin ERP Systems ▶ No Code  
 QA Manual QA Automation Design ▶  
 2D/3D Artist / Illustrator ▶ Gamedev ▶  
 Project Manager Product Manager  
 Product Owner Delivery Manager

Рис. 1.4 Головна сторінка порталу djinni.co

- **Applicant Tracking Systems (ATS).**

Системи типу Greenhouse, Lever, Workable призначені для автоматизації процесу рекрутингу. Вони допомагають керувати воронкою кандидатів, відстежувати етапи співбесід та комунікувати з претендентами.

- **Професійні соціальні мережі.**

LinkedIn є найяскравішим прикладом. Його інструмент LinkedIn Talent Insights надає потужні аналітичні можливості для аналізу ринку талантів, міграції спеціалістів та попиту на навички.

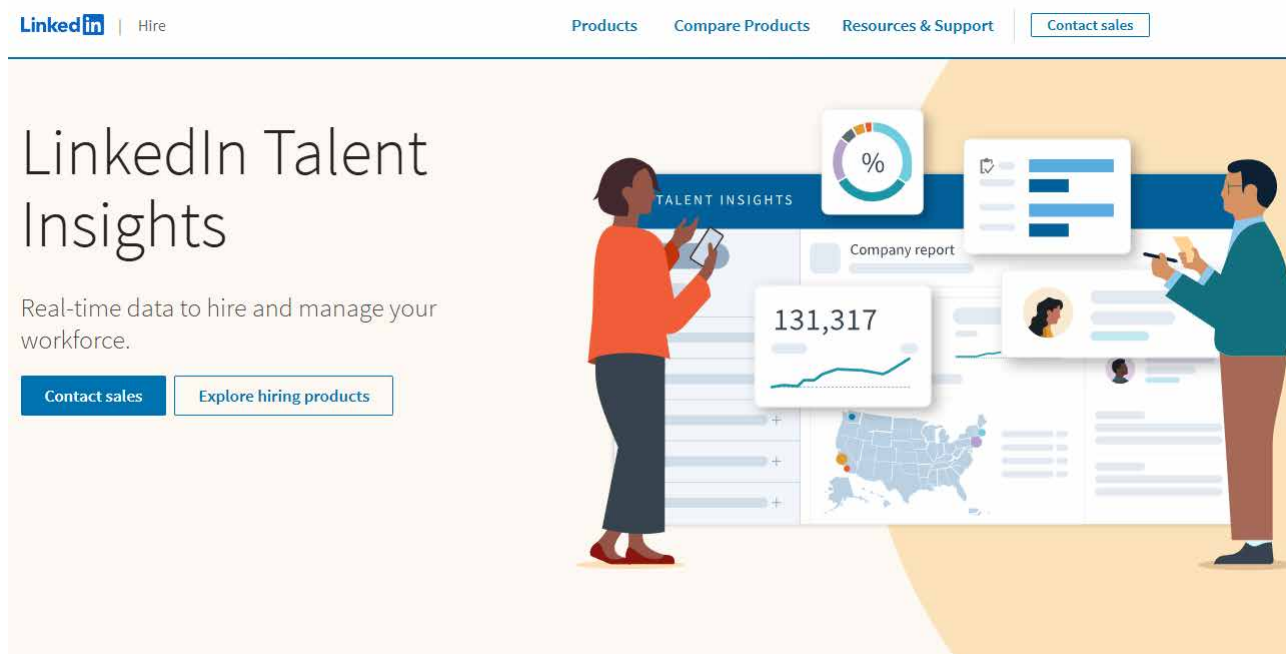


Рис. 1.5 Головна сторінка LinkedIn Talent Insights

Порівняльний аналіз існуючих HR-систем наведений в таблиці 1.1.

Таблиця 1.1

Порівняльний аналіз існуючих HR-систем

Тип системи	Приклади	Основне призначення	Аналітичні можливості	Недоліки
<b>Портали працевлаштуван ня</b>	work.ua, djinni.co	Агрегація вакансій	Базові звіти	Обмежена глибина аналізу
<b>Applicant Tracking Systems</b>	Greenhouse, Lever	Автоматизація рекрутингу	Внутрішня аналітика	Не аналізують зовнішній ринок

Тип системи	Приклади	Основне призначення	Аналітичні можливості	Недоліки
<b>Професійні соц. мережі</b>	LinkedIn Talent Insights	Аналіз ринку	Глибока аналітика, сегментація	Комерційний доступ, закрита екосистема

Існуючі рішення або надають обмежену аналітику, або є закритими комерційними продуктами. Це створює нішу для розробки користувацьких аналітичних систем, які можуть інтегрувати дані з різних джерел та застосовувати специфічні, адаптовані під конкретні бізнес-задачі моделі аналізу.

### 1.3 Ідентифікація ключових викликів на ринку праці

На основі аналізу предметної області та сучасних тенденцій, можна виділити кілька ключових викликів українського ринку праці, які можуть бути досліджені методами інтелектуального аналізу даних. Ці виклики створюють проблеми як для роботодавців, так і для шукачів, і їх кількісна оцінка є важливим завданням.

**1.3.1 Невідповідність навичок (Skills Gap).** Розрив між навичками, якими володіють кандидати на ринку, та навичками, які вимагають роботодавці для виконання конкретної роботи називають Skills Gap. Цей виклик має кілька проявів:

- **Надлишкові навички.** Кандидати можуть володіти великою кількістю загальних або застарілих навичок, які не є вирішальними для отримання високооплачуваної роботи.
- **Дефіцитні навички.** Існують специфічні, часто нові, технології або компетенції, попит на які значно перевищує кількість спеціалістів, що ними володіють.

В умовах "ринку роботодавця" ця проблема стає ще гострішою. Компанії можуть обирати з великої кількості кандидатів, підвищуючи вимоги та шукаючи ідеального "єдинорога" з рідкісним набором умінь. Для кандидатів це означає необхідність постійного навчання та адаптації, щоб відповідати цим вимогам.

**1.3.2 Дисбаланс попиту та пропозиції на ринку.** Цей виклик стосується не лише навичок, але й географічних та галузевих аспектів ринку. Дисбаланс може проявлятися у наступному:

- **Географічний дисбаланс.** Висока концентрація вакансій у великих містах (Київ, Львів) і значно менша – в регіонах. Водночас, завдяки поширенню дистанційної роботи, конкуренція за віддалені позиції зростає в рази, оскільки на них претендують кандидати з усієї країни.
- **Галузевий дисбаланс.** Деякі сектори економіки можуть генерувати значно більше вакансій, ніж інші (наприклад, виробництво або освіта), що створює нерівномірний розподіл можливостей.
- **Кваліфікаційний дисбаланс.** Переважна більшість вакансій може бути орієнтована на спеціалістів Middle або Senior рівня, тоді як на ринку присутня велика кількість фахівців-початківців (Junior, Intern, Trainee), для яких вхід у професію стає серйозним викликом.

**1.3.3 Вплив зовнішніх факторів на структуру ринку.** Ринок праці є вкрай чутливим до зовнішніх економічних, соціальних та політичних подій. Повномасштабне вторгнення суттєво трансформувало ринок:

- **Зростання попиту на військові спеціальності.** З'явився і стрімко зріс попит на професії, безпосередньо пов'язані з обороною країни.
- **Зміна пріоритетів у зарплатах.** Аналіз показує, що ці спеціальності часто потрапляють у високий зарплатний сегмент, що змінює загальну структуру ринку.
- **Релокація бізнесу та спеціалістів.** Внутрішня міграція та релокація компаній вплинули на географічний розподіл вакансій.

## 1.4 Постановка завдання магістерського дослідження

Потрібно розробити програмне забезпечення аналітичної системи, яка б дозволяла вирішувати задачі, недоступні або обмежено доступні в існуючих HR-платформах, а саме: проводити глибокий інтелектуальний аналіз даних про вакансії для виявлення прихованих тенденцій, патернів та закономірностей.

**1.4.1 Формулювання бізнес-вимог.** З точки зору кінцевих користувачів (рекрутерів, HR-аналітиків, керівників), система повинна надавати відповіді на ключові бізнес-питання.

- Яка поточна структура ринку праці? Які зарплатні сегменти існують і які посади є для них типовими?
- Як змінюється попит на спеціалістів у різних регіонах та галузях? Де спостерігається найвища конкуренція за кандидатів?
- Які фактори (зарплата, кількість навичок, регіон) найбільше впливають на тривалість пошуку кандидатів?
- Які навички є найбільш затребуваними та як вони корелюють між собою? Який "розрив у навичках" існує між масовим та високооплачуваним сегментами ринку?
- Яка динаміка появи нових вакансій? Чи існують сезонні коливання і який прогноз на найближче майбутнє?

**1.4.2 Визначення функціональних та нефункціональних вимог.** На основі бізнес-вимог, до системи висуваються наступні вимоги.

### **Функціональні вимоги:**

1. Система повинна забезпечувати автоматизований збір даних про вакансії з зовнішніх джерел (веб-порталів).
2. Система повинна виконувати очищення, трансформацію та завантаження даних (ETL) у централізоване сховище даних.
3. Система повинна надавати можливість проведення OLAP-аналізу за ключовими вимірами: Час, Локація, Галузь, Навички.

4. Система повинна реалізовувати алгоритми Data Mining для:
  - 4.1 Кластеризації вакансій за рівнем заробітної плати.
  - 4.2 Класифікації попиту для комбінацій "локація-галузь".
  - 4.3 Прогнозування кількості нових вакансій.
5. Система повинна забезпечувати візуалізацію результатів аналізу.

#### **Нефункціональні вимоги:**

1. **Продуктивність** - система повинна обробляти великі обсяги даних (100,000+ записів) та виконувати аналітичні запити в прийнятний час (до 10-15 секунд для типових агрегацій).
2. **Надійність** - ETL-процеси повинні бути стійкими до збоїв та забезпечувати однорідність даних у сховищі.
3. **Масштабованість** - архітектура системи повинна дозволяти майбутнє розширення за рахунок додавання нових джерел даних або нових аналітичних модулів.
4. **Зручність використання** - результати аналізу повинні бути представлені у зрозумілій та легко інтерпретованій формі.

**1.4.3. Формулювання основних задач дослідження.** Виходячи з поставлених вимог та ідентифікованих викликів предметної області, дане магістерське дослідження буде сфокусовано на вирішенні наступних науково-практичних задач:

- **Задача моделювання даних.** Розробити та обґрунтувати структуру сховища даних, яка буде оптимальною для аналітичних запитів та побудови OLAP-кубів у предметній області управління працевлаштуванням.
- **Задача інтелектуального аналізу.** Дослідити та адаптувати алгоритми Data Mining для вирішення бізнес-задач.
- **Задача оцінки та інтерпретації.** Провести експериментальні дослідження з використанням розроблених моделей на реальному наборі

даних, проаналізувати отримані результати, візуалізувати їх та сформулювати практичні висновки для кінцевих користувачів системи.

## 2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ АНАЛІТИЧНИХ ПРОЦЕСІВ

### 2.1 Моделювання бізнес-процесів

**2.1.1 Процес взаємодії у сфері HR.** Для візуалізації динамічної взаємодії між ключовими учасниками предметної області розглянемо типовий сценарій: "Пошук та відбір кандидатів на вакансію". Цей процес ідеально ілюструється за допомогою **діаграми послідовності (Sequence Diagram)**, яка показує обмін повідомленнями між об'єктами в часі.

Діаграма (рис. 2.1) демонструє, як HR-менеджер, використовуючи веб-портал, взаємодіє з кандидатами та внутрішніми системами компанії для закриття вакансії.

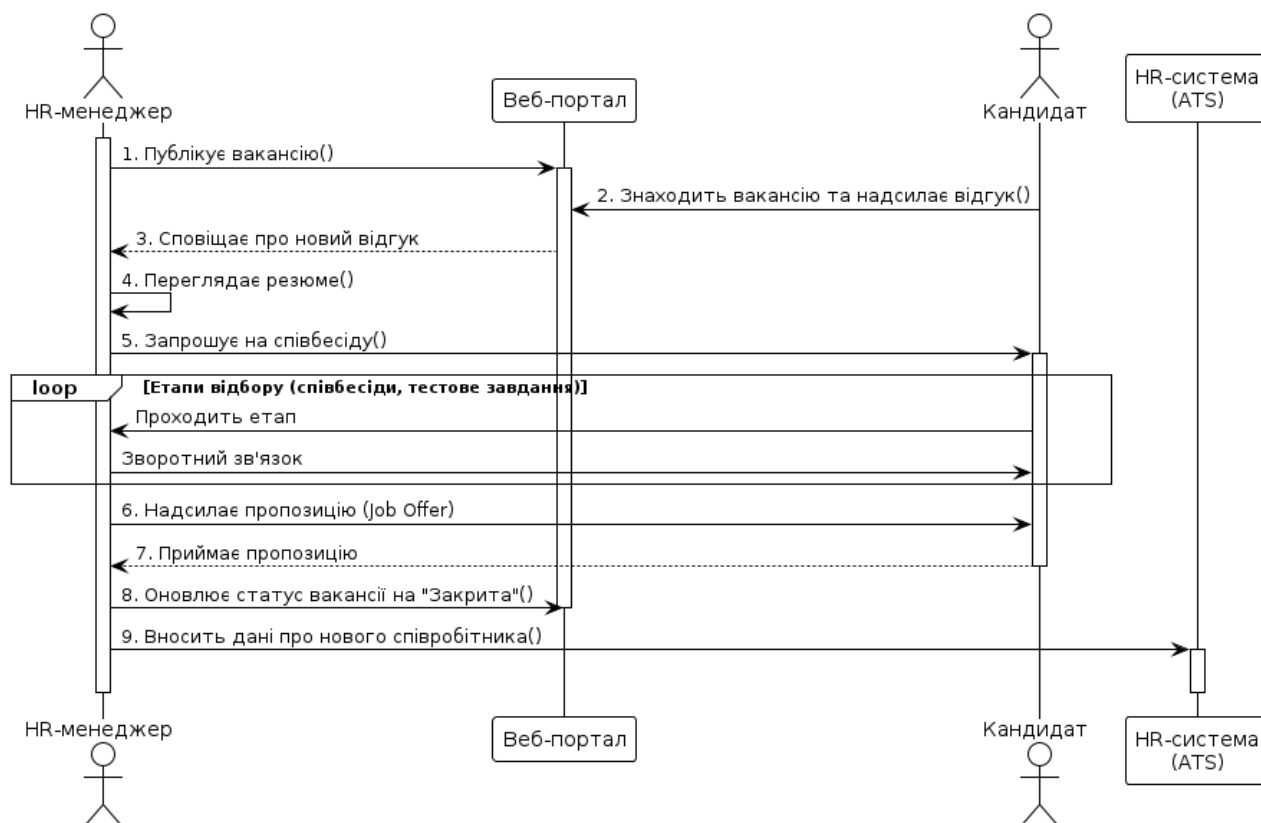


Рис. 2.1 Діаграма послідовності для процесу "Пошук та відбір кандидатів"

Діаграма демонструє, як HR-менеджер, використовуючи веб-портал, взаємодіє з кандидатами та внутрішніми системами компанії (ATS). Процес починається з публікації вакансії, проходить через етапи відгуку кандидата, співбесід, пропозиції роботи і завершується оновленням статусу вакансії та внесенням даних про нового співробітника. Ця діаграма ілюструє потік інформації, який є основою для подальшого проектування аналітичної системи.

Опис процесу, зображеного на діаграмі:

1. HR-менеджер створює та публікує нову вакансію на веб-порталі (напр., work.ua).
2. Кандидат, який знаходиться в пошуку роботи, знаходить цю вакансію та надсилає свій відгук (резюме).
3. Веб-портал сповіщає HR-менеджера про новий відгук.
4. HR-менеджер переглядає резюме і, якщо кандидат виглядає перспективним, запрошує його на співбесіду.
5. Після успішного проходження всіх етапів відбору (які для спрощення показані як один цикл), HR-менеджер формує та надсилає кандидату пропозицію про роботу (Job Offer).
6. Кандидат приймає пропозицію.
7. HR-менеджер оновлює статус вакансії на веб-порталі на "Закрита" та вносить дані про нового співробітника у внутрішню HR-систему (ATS) компанії.

**2.1.2 Процес пошуку роботи кандидатом.** Для моделювання поведінки іншого ключового учасника предметної області – Кандидата – було побудовано діаграму активності (Activity Diagram). На відміну від діаграми послідовності, яка фокусується на взаємодії об'єктів, діаграма активності ідеально візуалізує покроковий потік робіт та рішень, що приймаються користувачем.

Діаграма (рис. 2.2) ілюструє типовий шлях кандидата на веб-порталі, починаючи від загального пошуку вакансій і закінчуючи надсиланням заявки на конкретну позицію.

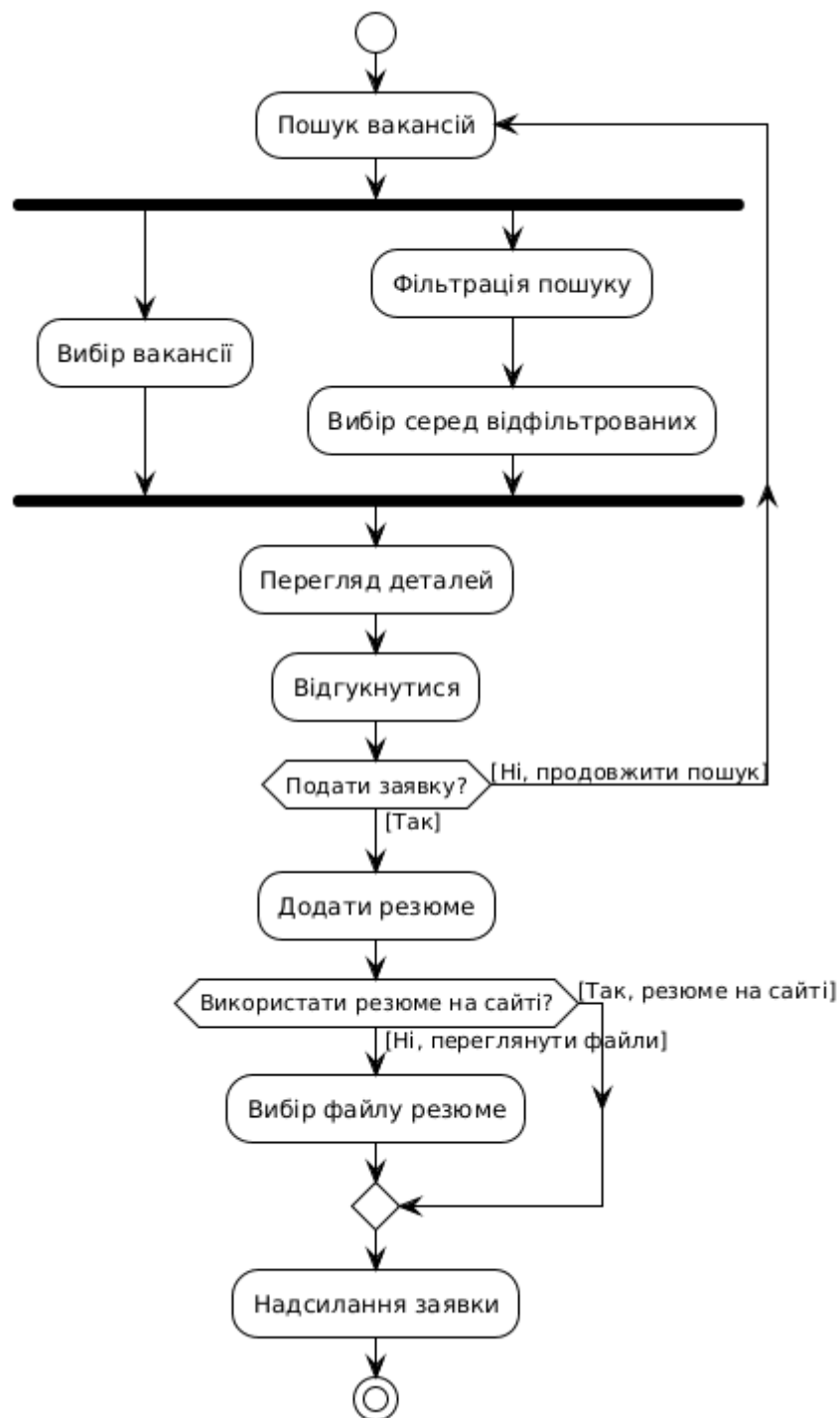


Рис. 2.2 Діаграма активності для процесу "Пошук роботи кандидатом"

Процес починається з активності "**Пошук вакансій**". Далі потік розгалужується, ілюструючи два паралельні сценарії поведінки користувача: він може або одразу **вибрати вакансію** з загального списку, або спочатку **застосувати фільтри** (за зарплатою, локацією тощо) і потім зробити вибір серед відфільтрованих результатів.

Після вибору вакансії обидва сценарії сходяться до єдиного потоку. Кандидат переходить до перегляду деталей вакансії. Якщо вона його зацікавила, він натискає "**Відгукнутися**". На цьому етапі виникає перша точка прийняття рішення: кандидат може або **подати заявку**, або **завершити перегляд** і вийти з процесу.

Якщо кандидат вирішує подати заявку, наступний крок – **додавання резюме**. Тут виникає друга точка прийняття рішення, що залежить від того, чи є у кандидата збережене резюме на сайті. Він може або **використати існуюче резюме**, або **завантажити новий файл** з комп'ютера. Обидва шляхи ведуть до фінальної дії – **надсилання заявки**, після чого процес для даної вакансії завершується.

**2.1.3 Процес проведення аналізу.** Діаграма прецедентів (Use Case Diagram) описує основні функції, які система надає користувачу. Для нашої аналітичної системи головним актором є "Аналітик" (HR-аналітик, рекрутер), який виконує низку аналітичних завдань (рис. 2.3).



Рис. 2.3 Діаграма прецедентів аналітичної системи

Діаграма показує, що Аналітик може виконувати такі ключові функції, як аналіз динаміки ринку, сегментація вакансій, аналіз попиту, прогнозування та керування ETL-процесами. Це визначає функціональні межі розроблюваної системи.

## 2.2 Функціональне моделювання

**2.2.1 Діаграма функціональної декомпозиції (FDD).** Ця діаграма ієрархічно розбиває основну функцію системи на менші, більш керовані підфункції.

Для нашої аналітичної системи головною функцією є "Аналіз ринку праці". Діаграма (рис. 2.4) показує її декомпозицію на три основні

функціональні блоки: "Збір та обробка даних", "Інтелектуальний аналіз" та "Візуалізація та звітність".

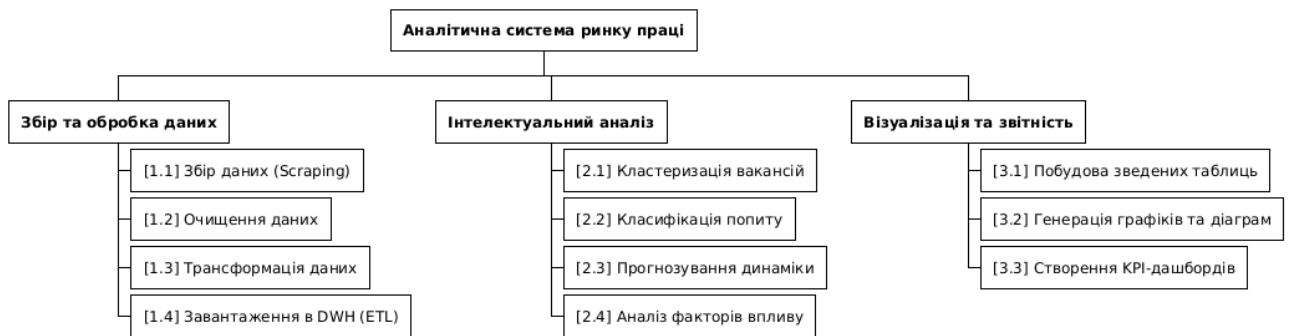


Рис. 2.4 Діаграма функціональної декомпозиції аналітичної системи

**2.2.2 Діаграма потоків даних (DFD).** Діаграма потоків даних моделює, як інформація рухається через систему. Вона показує зовнішні сутності, що взаємодіють із системою, процеси, що трансформують дані, та сховища, де дані зберігаються.

Діаграма (рис. 2.5) ілюструє повний цикл життя даних у системі: від збору на зовнішньому Веб-порталі до представлення кінцевому користувачу, Аналітику.

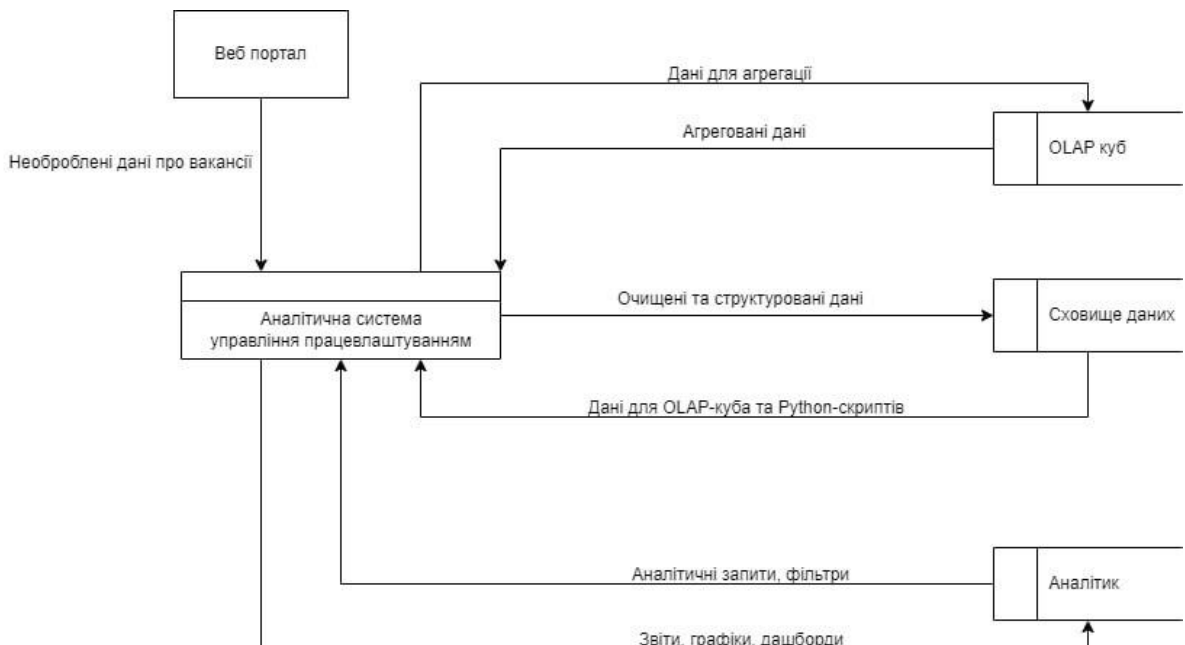


Рис. 2.5 Контекстна діаграма потоків даних (DFD, рівень 0)

## 3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ АНАЛІТИЧНОЇ СИСТЕМИ

### 3.1 Концепція архітектури

**3.1.1 Архітектурний стиль.** Для системи було обрано **трирівневу архітектуру**. Цей підхід передбачає логічний поділ системи на три основні рівні, кожен з яких виконує свою специфічну функцію.

- **Рівень представлення (Presentation Layer).** Відповідає за взаємодію з кінцевим користувачем (аналітиком). Цей рівень включає інструменти візуалізації, такі як звіти та графіки, що дозволяють користувачу легко інтерпретувати результати аналізу.
- **Рівень бізнес-логіки та аналітики (Business Logic / Analytics Layer).** Це "мозок" системи, де відбуваються всі основні обчислення. Цей рівень включає OLAP-сервер для обробки багатовимірних запитів, а також середовище для виконання скриптів машинного навчання.
- **Рівень даних (Data Layer).** Відповідає за зберігання та управління даними. Центральним компонентом цього рівня є сховище даних, а також проміжні бази даних та файлові сховища, що використовуються в процесі ETL.

**3.1.2. Опис основних компонентів системи та їх взаємодії.** Архітектура системи складається з кількох ключових компонентів, кожен з яких виконує свою унікальну роль у загальному потоці даних:

- **Компонент збору даних.** Реалізований у вигляді **API скрапера**, розгорнутого на сервері застосунку. Його завдання – періодично звертатися до зовнішнього джерела (веб-порталу work.ua), збирати необроблені дані про вакансії та зберігати їх у проміжному сховищі (наприклад, у вигляді CSV-файлів) для подальшої обробки.

- **Компонент обробки та зберігання даних:**
  - **ETL-сервіс (на базі SSIS [6])** - вилучення необроблених даних, їх очищення, трансформацію (наприклад, розбиття рядка навичок, приведення до єдиних форматів) та завантаження у структуроване сховище.
  - **Сховище даних** - реляційна база даних (MS SQL Server [5]), спроектована за багатовимірною моделлю для оптимізації аналітичних запитів.
- **Компонент аналітики.** Цей рівень реалізує бізнес-логіку та інтелектуальний аналіз.
  - **OLAP-сервер (на базі SSAS [7])** - багатовимірні OLAP-куби на основі даних зі сховища, що дозволяє миттєво отримувати агреговані дані та розраховувати KPI.
  - **Середовище виконання Python-скриптів [8]** - виконання складних алгоритмів Data Mining (кластеризації, класифікації, прогнозування), які виходять за межі можливостей стандартного OLAP.
- **Компонент представлення:** Надає інтерфейс для взаємодії аналітика з результатами.
  - **Статичні та інтерактивні звіти (на базі SSRS або Power BI [16])** - дані з OLAP-кубів у вигляді таблиць та дашбордів.
  - **Візуалізація (на базі Matplotlib [13] / Seaborn [14])** - графіки, згенеровані Python-скриптами, що ілюструють результати роботи моделей машинного навчання.

Для ілюстрації процесу аналізу, що відбувається в реальному часі, був змодельований типовий сценарій взаємодії аналітика з системою за допомогою діаграми послідовності (рис. 3.1). Ця діаграма показує, як запит, ініційований користувачем, проходить через усі рівні архітектури, від клієнтського

інструменту до сховища даних, і повертається у вигляді готового візуального результату.

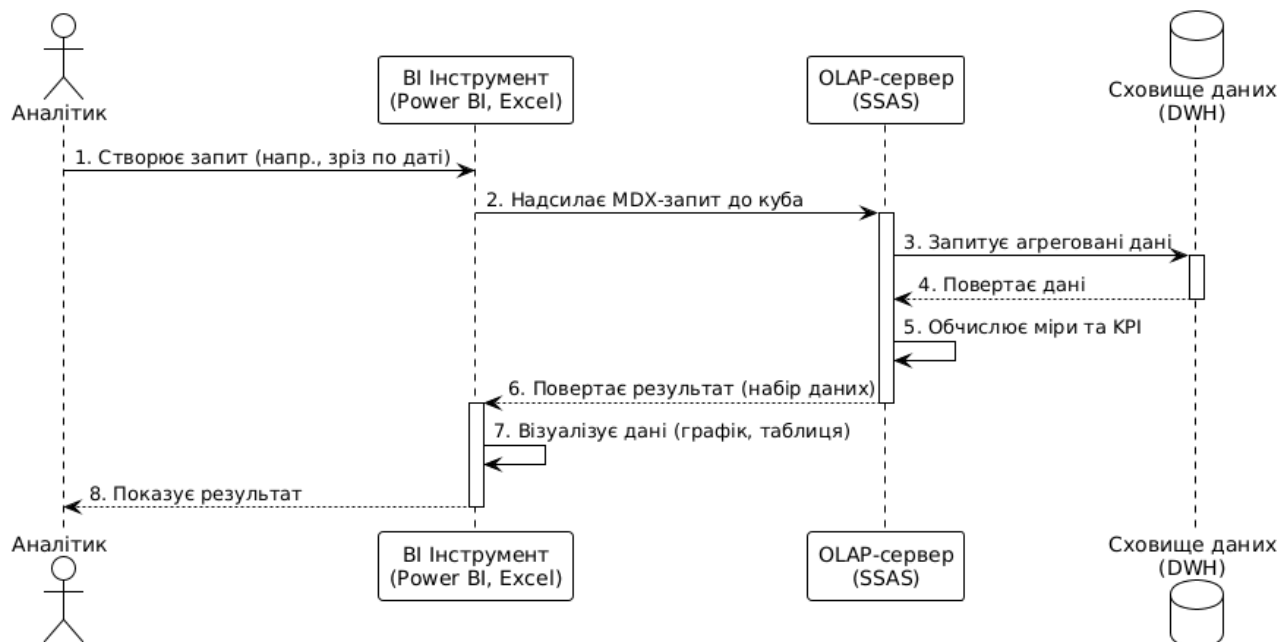


Рис. 3.1 Діаграма послідовності сценарію "Аналіз динаміки ринку"

1. Аналітик створює запит, взаємодіючи з візуальними елементами – наприклад, застосовує фільтр по даті, обирає певні категорії або додає новий вимір на вісь графіка.

2. BI-інструмент автоматично трансформує дію користувача у формалізований запит мовою MDX (Multidimensional Expressions) і надсилає його до OLAP-сервера (SSAS).

3. OLAP-сервер аналізує MDX-запит і визначає, які агреговані дані йому потрібні для обчислень. Він надсилає оптимізований SQL-запит до Сховища даних (DWH). На цьому етапі SSAS може використовувати свої внутрішні кеші або заздалегідь прораховані агрегації, що значно прискорює процес.

4. DWH виконує запит і повертає на OLAP-сервер необхідний набір агрегованих даних.

5. OLAP-сервер, отримавши дані, виконує фінальні обчислення. Він розраховує складні міри, обчислювані елементи та всі компоненти KPI (Value, Goal, Status, Trend), що були визначені в логіці куба.

6. SSAS повертає на BI-інструмент фінальний, повністю розрахований набір даних (dataset).

7. BI-інструмент, отримавши набір даних, оновлює візуальні елементи – перемальовує графік, оновлює значення в таблиці або змінює колір індикатора KPI.

8. Аналітик бачить на своєму екрані оновлений візуальний результат, що відповідає його запиту.

Архітектуру аналітичної системи та її модулів наведено на рисунку 3.2.

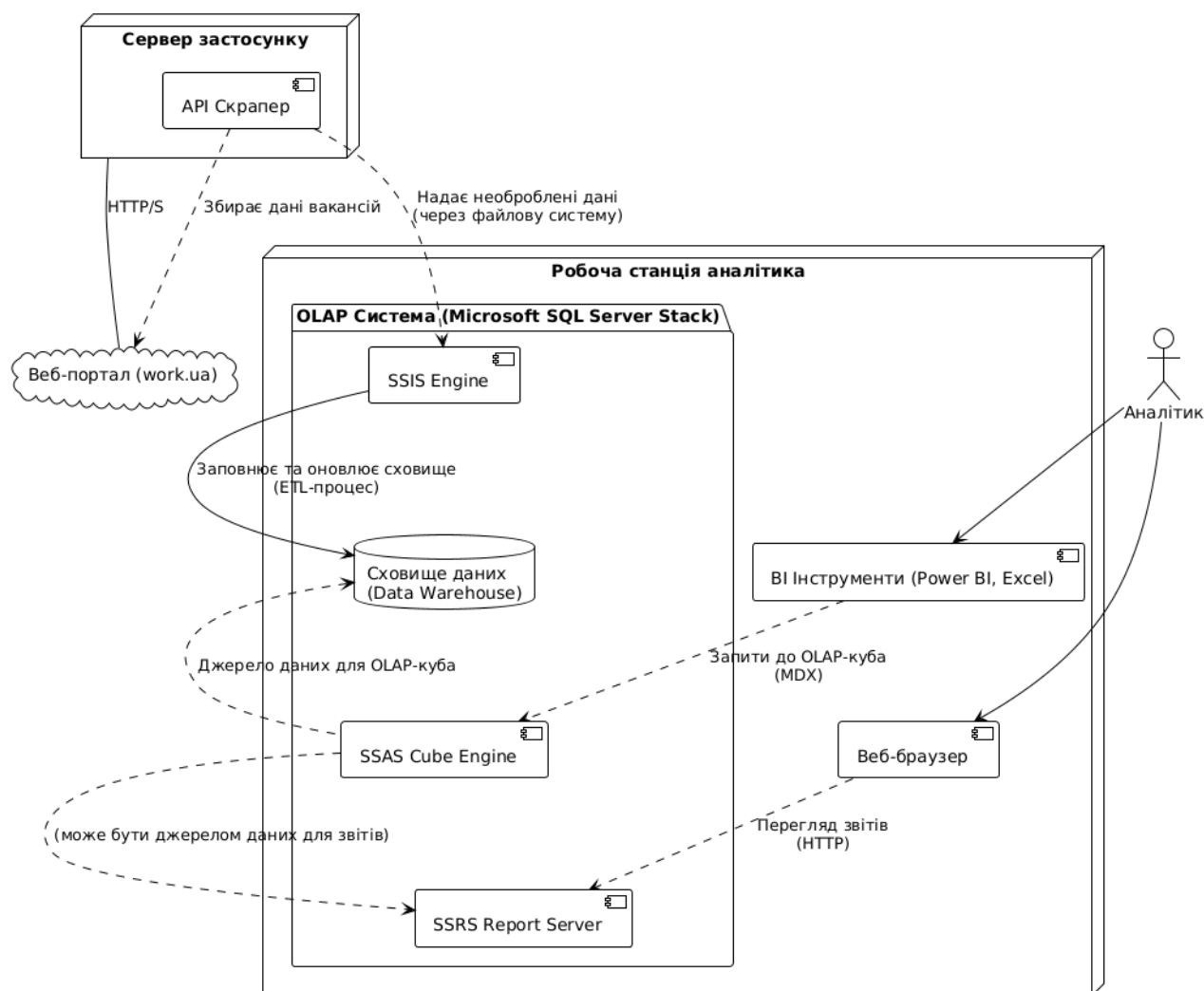


Рис. 3.2 Діаграма розгортання

## 3.2 Компонент збору даних

Компонент збору даних є першою і критично важливою ланкою в архітектурі аналітичної системи. Його основне завдання – автоматизоване вилучення (скрапінг) актуальних даних про вакансії з зовнішнього джерела, веб-порталу work.ua, та їх збереження у проміжному форматі для подальшої обробки ETL-процесами.

**3.2.1 Вибір технології та архітектура.** Для реалізації було обрано технологічний стек Node.js [17] та TypeScript [18]. Цей вибір обґрунтований кількома факторами:

- **Асинхронність.** Ідеально підходить для задач, пов'язаних з великою кількістю мережових запитів, дозволяючи ефективно керувати очікуванням відповідей від сервера.
- **TypeScript** забезпечує строгу типізацію, що значно підвищує надійність коду, спрощує його підтримку та зменшує кількість помилок на етапі розробки.
- **Наявність бібліотек,** таких як axios [19] для виконання HTTP-запитів та cheerio [20] для парсингу HTML-коду, значно прискорює процес розробки.

Скрапер реалізований як серверний додаток на базі фреймворку Express.js, який надає API-ендпоінт (напр., /scrape-and-save). При зверненні до цього ендпоінту запускається процес збору даних. Результати зберігаються у файли формату JSON, які потім слугують джерелом для ETL-паketу SSIS.

**3.2.2 Алгоритм збору даних.** Процес збору даних реалізовано за ітеративним алгоритмом, що дозволяє обходити сторінки з вакансіями та збирати детальну інформацію по кожній з них.

1. Створюється екземпляр axios зі стандартними заголовками (User-Agent) для імітації поведінки реального браузера.

2. Скрапер послідовно обходить сторінки з результатами пошуку (напр., `work.ua/jobs-it/?page=N`).
3. На кожній сторінці відбувається парсинг HTML-коду для вилучення посилань на окремі вакансії.
4. Для кожного знайденого посилання послідовно виконується збір детальної інформації.
5. Для кожної окремої вакансії викликається функція `scrapeWorkUaJobDetails`, яка парсить сторінку вакансії та вилучає всі необхідні атрибути (опис, зарплата, навички тощо).
6. Після завершення обходу всіх сторінок, зібрані дані агрегуються в єдиний масив та зберігаються у JSON-файл.

**3.2.3 Реалізація парсингу сторінок.** Для парсингу HTML-коду використовується бібліотека `Cheerio`, яка надає `jQuery`-подібний синтаксис для навігації та маніпуляції DOM-деревом на стороні сервера.

Приклад фрагменту коду для вилучення навичок наведено на рисунку 3.3.

```
1 // --- Extract Skills ---
2 const skills: string[] = [];
3 $("li.no-style.label.label-skill").each((_i, el) => {
4     const skillText = $(el).text().trim();
5     if (skillText) {
6         skills.push(skillText);
7     }
8 });
9 details.skills = skills.length ? skills : null;
```

Рис. 3.3 Фрагмент коду для вилучення навичок

У цьому прикладі код знаходить усі елементи `<li>` з відповідними CSS-класами, вилучає їх текстовий вміст, очищує від зайвих пробілів та додає до

масиву навичок. Аналогічні селектори використовуються для вилучення назви вакансії, зарплати, локації, опису та інших атрибутів. Особливу увагу приділено обробці дати публікації, для якої парситься атрибут `datetime` тегу `<time>`, що забезпечує точність часових даних.

**3.2.4 Механізми обходу блокувань та обробки помилок.** Для забезпечення стабільної роботи та мінімізації ризику блокування з боку веб-порталу реалізовано наступні механізми:

- **Затримки.** Між запитами до сторінок та окремих вакансій встановлено випадкові затримки (наприклад, від 1 до 3 секунд). Це імітує поведінку людини та знижує навантаження на цільовий сервер.
- **Обробка HTTP-помилки.** Перевіряється статус-код відповіді сервера. У випадку отримання помилки 429 (Too Many Requests), процес автоматично призупиняється на тривалий час (напр., 60 секунд) перед повторною спробою, що дозволяє уникнути перманентного блокування.
- **Обробка помилок.** Код написано з урахуванням можливої відсутності деяких елементів на сторінці. Якщо, наприклад, зарплата не вказана, скрапер не генерує помилку, а коректно записує значення `null`. Це забезпечує стійкість процесу навіть при зміні структури HTML-коду на окремих сторінках.

Ці механізми дозволяють системі працювати надійно та збирати максимально повні дані для подальших етапів аналізу.

### 3.3 Сховище даних та ETL-процеси

**3.3.1 Фінальна логічна модель сховища даних.** Взаємозв'язок між спроектованими таблицею фактів та таблицями вимірів утворює логічну модель даних, що реалізує "схему зірка". Ця модель є візуальним представленням архітектури сховища даних і слугує "кресленням" для подальшої фізичної реалізації в середовищі MS SQL Server.

Фінальна логічна модель представлена на рисунку 3.4.



Рис. 3.4 Логічна модель сховища даних

#### Опис моделі:

- **Центральний елемент.** В центрі моделі знаходиться таблиця Факт\_Вакансії. Вона містить зовнішні ключі до всіх таблиць вимірів (Вакансія, Локація, Роботодавець, Дата, Навичка, Індустрія), а також дві основні кількісні міри: Мінімальна Зарплата та Максимальна Зарплата.
- **Виміри.** Навколо таблиці фактів розташовані таблиці вимірів, що описують контекст кожної події:
  - Вимір Роботодавця - зберігає інформацію про компанії, що розміщують вакансії.
  - Вимір Вакансії - містить унікальні назви та описи вакансій.
  - Вимір Локації - зберігає географічні дані або тип "Дистанційна робота".
  - Вимір Індустрії - описує галузь, до якої належить вакансія.
  - Вимір Навичок - містить унікальний довідник усіх навичок.

- Вимір Дати - надає часовий контекст з атрибутами для аналізу динаміки.
- **Зв'язки.** Кожна таблиця вимірів пов'язана з таблицею фактів зв'язком "один до багатьох". Це означає, що один запис у таблиці вимірів (наприклад, одна локація "Київ") може відповідати багатьом записам у таблиці фактів.

**3.3.2 Створення фізичної моделі сховища даних в MS SQL Server.** На основі логічної моделі, розробленої в попередньому розділі, було створено фізичну базу даних в MS SQL Server. Для визначення таблиць, їх атрибутів, типів даних та зв'язків було використано SQL-скрипти з мовою визначення даних (Data Definition Language, DDL).

Ключовим рішенням при створенні таблиць вимірів (Dim\_Date, Dim\_Location, Dim\_Skill тощо) було використання сурогатних ключів (surrogate keys) – унікальних ідентифікаторів (INT з властивістю IDENTITY), які не залежать від бізнес-ключів із зовнішніх систем. Це забезпечує стабільність зв'язків у сховищі, навіть якщо дані в джерелі змінюються.

Центральна таблиця Fact\_Jobs була створена для зберігання зовнішніх ключів, що посилаються на сурогатні ключі таблиць вимірів, а також для зберігання кількісних показників (мір), таких як SalaryRangeMin та SalaryRangeMax.

Фінальна реалізована структура сховища даних представлена на рисунку 3.5.

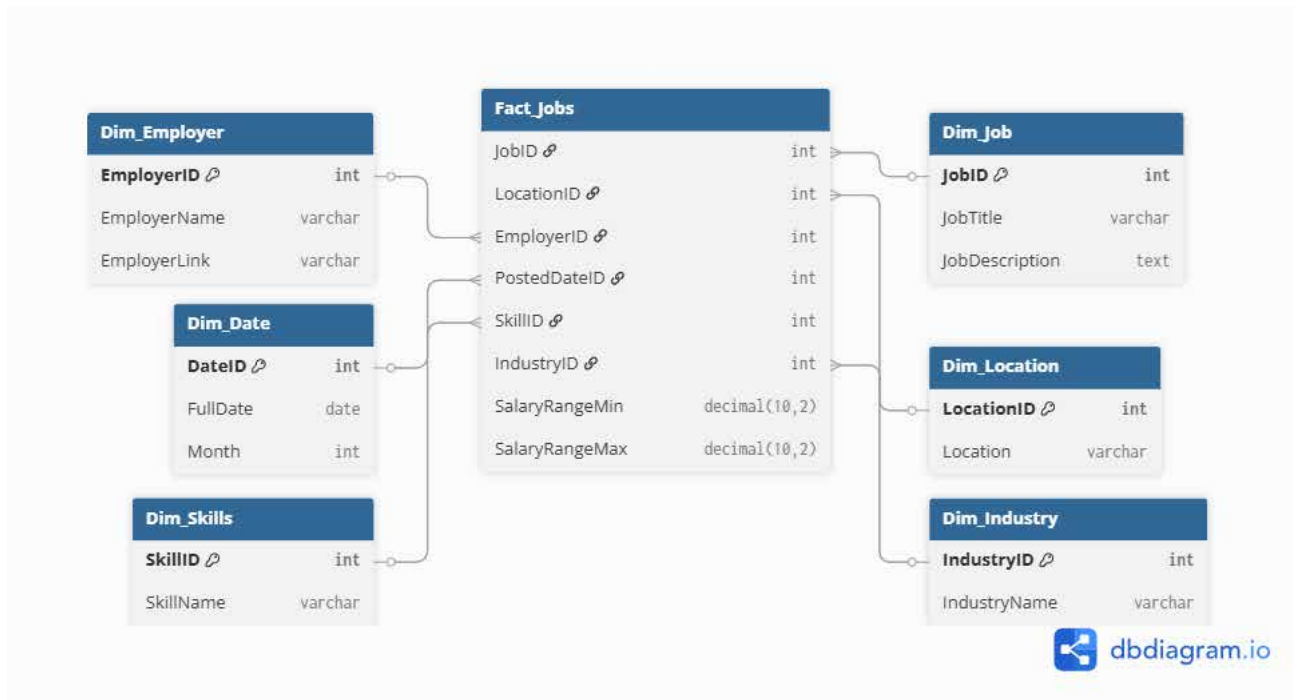


Рис. 3.5 ER-діаграма сховища даних

**3.3.3 Розробка ETL-пакетів.** Процеси вилучення, трансформації та завантаження даних були автоматизовані за допомогою SQL Server Integration Services. Було розроблено комплексний ETL-пакет, логіка роботи якого відповідає діаграмі активності, представленій на рисунку 3.6.

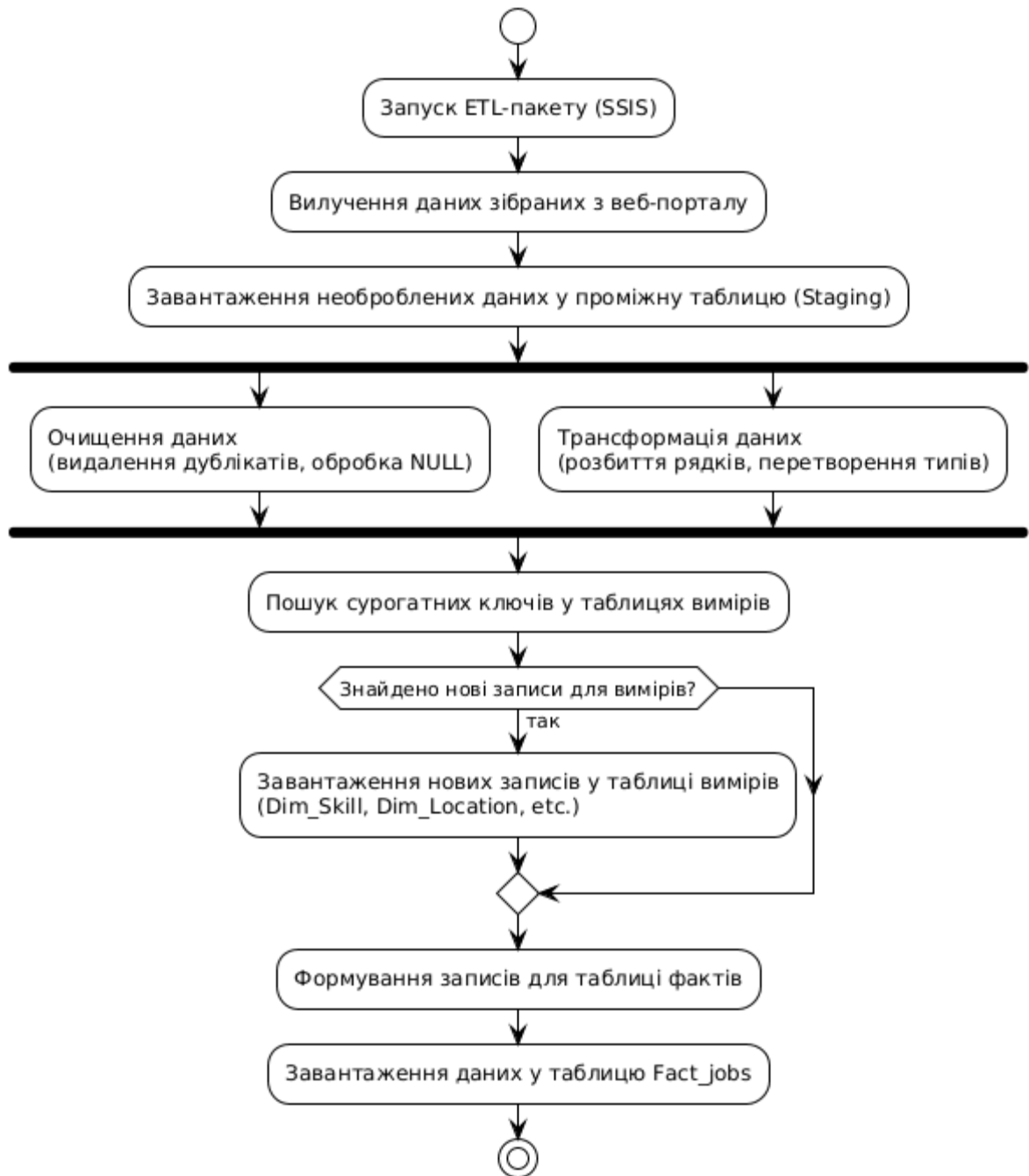


Рис. 3.6 Діаграма активності ETL-процесу

1. **Extract (Вилучення).** Пакет починається з компонента "Excel Source", який зчитує необроблені дані про вакансії з вихідного файлу (.xlsx), що був отриманий на етапі скрапінгу.

2. **Staging (Проміжне завантаження).** Дані завантажуються у проміжну таблицю в базі даних для подальшої обробки, що дозволяє ізолювати процеси трансформації від доступу до зовнішнього файлу.

3. **Transform & Load (Трансформація та Завантаження).** Основна логіка реалізована в компоненті Data Flow Task, який виконує всі необхідні перетворення та завантажує дані у фінальні таблиці сховища.

**3.2.4. Ключові трансформації даних в SSIS.** Потік даних (Data Flow) в рамках ETL-паketу є складним процесом, що включає численні трансформації для забезпечення якості та консистентності даних. Схеми потоків даних представлені на рисунках 3.7 – 3.9.

Код виконання задачі SQL Task 1 наведено в Додатку А.

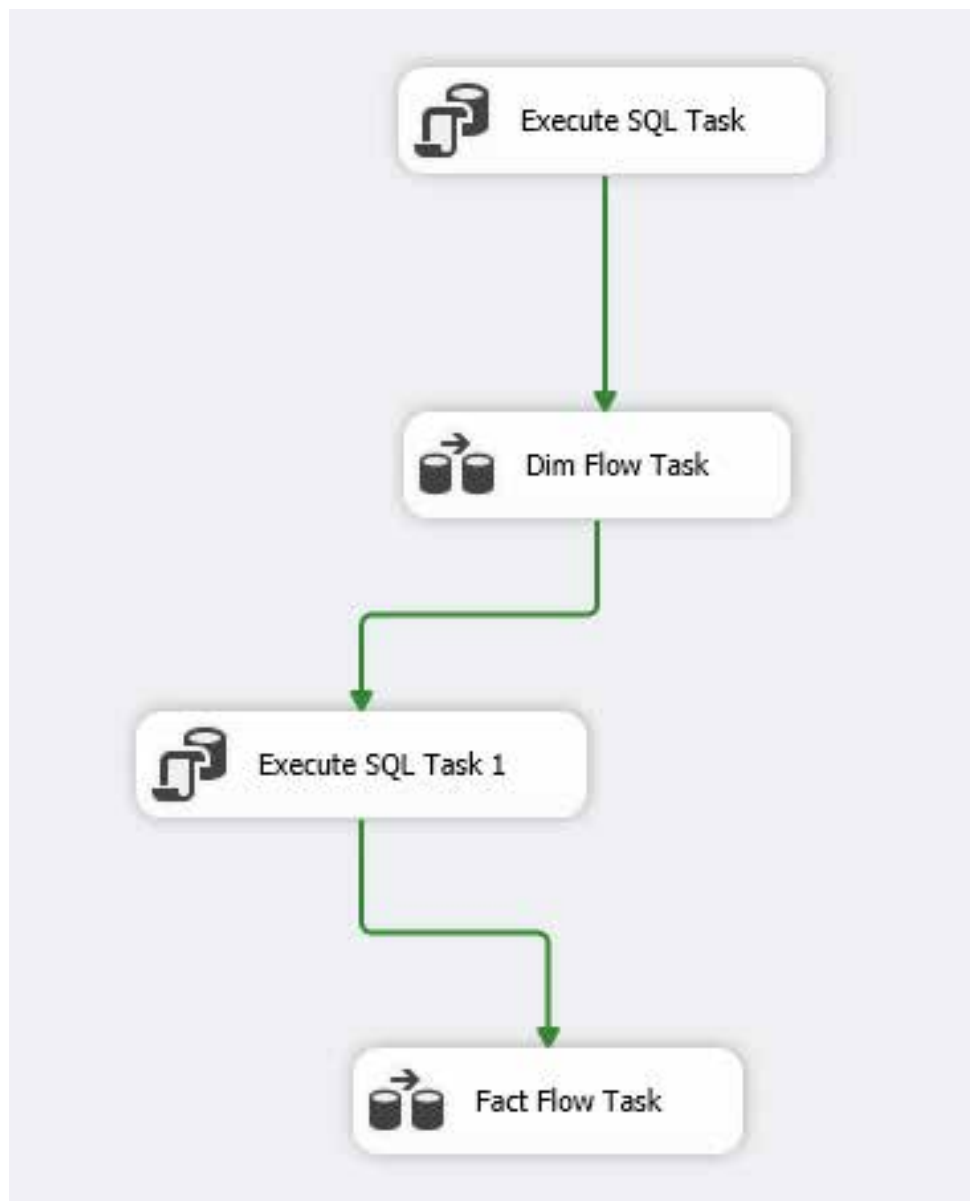


Рис. 3.7 Загальна схема

```
1  SET NOCOUNT ON;  
2  
3  TRUNCATE TABLE dbo.Fact_Jobs;  
4  PRINT 'Fact_Jobs cleared.';  
5  GO  
6  
7  TRUNCATE TABLE dbo.Staging_Jobs;  
8  PRINT 'Staging_Jobs cleared.';  
9  GO  
10  
11 DELETE FROM dbo.Dim_Job;  
12 DBCC CHECKIDENT('dbo.Dim_Job', RESEED, 0);  
13 PRINT 'Dim_Job cleared and reseeded.';  
14 GO  
15  
16 DELETE FROM dbo.Dim_Location;  
17 DBCC CHECKIDENT('dbo.Dim_Location', RESEED, 0);  
18 PRINT 'Dim_Location cleared and reseeded.';  
19 GO
```

Рис. 3.8 Приклад коду для очищення таблиць

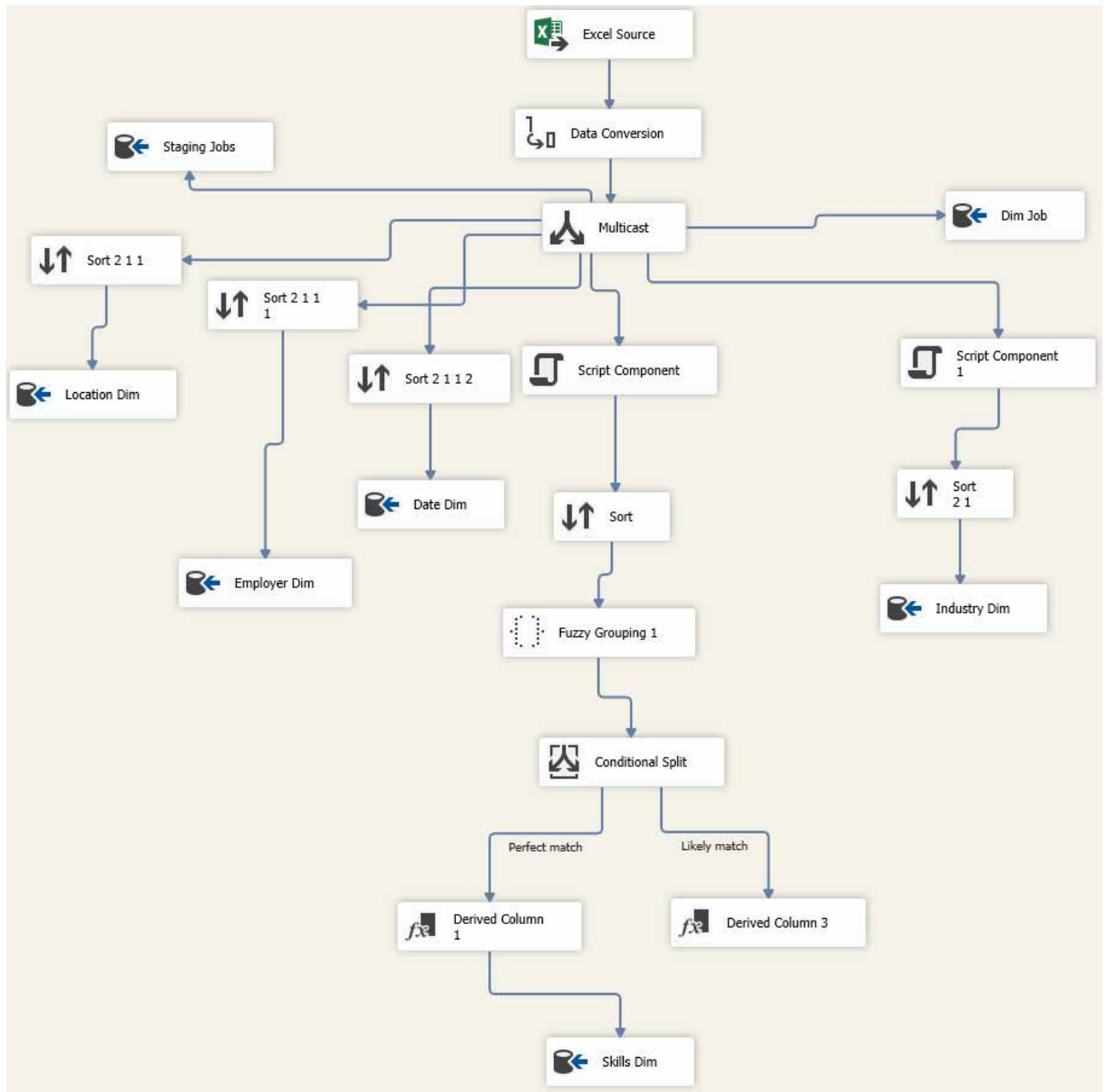


Рис. 3.9 Схема потоку даних в SSIS для завантаження вимірів та фактів

#### Ключові етапи трансформації, реалізовані в потоці:

- Data Conversion.** На самому початку потоку використовується трансформація для приведення текстових полів з датами та числами до відповідних типів даних (DT\_DATE, DT\_NUMERIC), що є необхідним для подальших обчислень.
- Multicast.** Ця трансформація дозволяє розпаралелити(розподілити) потік даних. Один вхідний потік копіюється і направляєтся на кілька

виходів одночасно, що дозволяє паралельно завантажувати різні таблиці вимірів (Dim\_Job, Dim\_Location тощо), які не залежать одна від одної.

- **Sort та Aggregate.** Перед завантаженням у таблиці вимірів дані сортуються та агрегуються (наприклад, за допомогою Sort з опцією "Remove rows with duplicate sort values") для створення унікальних записів вимірів.
- **Fuzzy Grouping.** Для очищення "брудних" даних, таких як назви навичок (Skills), які можуть мати незначні відмінності в написанні, було використано трансформацію Fuzzy Grouping. Вона дозволяє згрупувати схожі за написанням рядки, ідентифікуючи канонічне (основне) значення, що значно підвищує якість даних у вимірі Dim\_Skill.
- **Script Component.** Для реалізації складної бізнес-логіки, такої як розбиття рядка з навичками, розділеними крапкою з комою, на окремі рядки, було використано Script Component. Цей компонент, написаний на C#, створює кілька вихідних рядків з одного вхідного, що є необхідним для реалізації гранулярності "одна навичка на один рядок" у таблиці фактів.
- **Lookup Transformation.** На фінальному етапі, перед завантаженням у таблицю Fact\_Jobs, використовується серія трансформацій Lookup. Вони звертаються до вже завантажених таблиць вимірів (Dim\_Date, Dim\_Location, Dim\_Skill тощо) для пошуку відповідних сурогатних ключів за бізнес-ключами (наприклад, за FullDate, LocationName, SkillName). Це дозволяє коректно сформувати записи для таблиці фактів.

Таким чином, розроблений ETL-пакет забезпечує повний цикл обробки даних, перетворюючи необроблену інформацію з файлу-джерела на структуровані, очищені та консистентні дані, готові для подальшого аналізу в OLAP-кубі.

### 3.4 Реалізація компонентів аналітики

**3.4.1 Розробка OLAP-куба.** На основі спроектованого сховища даних було створено багатовимірний OLAP-куб в середовищі SSAS. Цей куб є центральним аналітичним компонентом, що дозволяє миттєво агрегувати мільйони записів з таблиці фактів та виконувати складні аналітичні запити мовою MDX (Multidimensional Expressions).

На основі спроектованого сховища даних було створено багатовимірний OLAP-куб в середовищі SSAS. Цей куб є центральним аналітичним компонентом, що дозволяє миттєво агрегувати мільйони записів з таблиці фактів та виконувати складні аналітичні запити мовою MDX (Multidimensional Expressions).

**Процес розробки** налічував декілька пунктів.

- Визначення груп мір та вимірів: Таблиця Fact\_Jobs була використана для створення групи мір, що включає такі показники, як [Fact Jobs Count], [Average Min Salary] та [Average Max Salary]. Таблиці вимірів (Dim\_Date, Dim\_Location тощо) були підключені до куба як виміри.
- Створення ієрархій: Для зручності аналізу в часі було створено ієрархію Рік > Місяць у вимірі [Dim Date].
- Реалізація ключових показників ефективності (KPI): Безпосередньо в кубі було налаштовано кілька KPI для моніторингу ключових метрик ринку праці. Приклад реалізованих KPI показано на рисунку 3.10.

Display Structure	Value	Goal	Status	Trend	Weight
KPI AVG Salary	24478.75	0.183.30		↑	
KPI Job City Min Avg Salary	20136.24	379.91		↓	
KPI Market Share by City	1	0.05		↓	
KPI Salary Growth Rate	0.26	5		↑	
KPI Skill Demand	1	1		↑	

Рис. 3.10 Приклад реалізованих KPI в браузері OLAP-куба

Як видно з рисунка, KPI дозволяють в реальному часі відстежувати такі показники, як середня зарплата (KPI AVG Salary), її ріст (KPI Salary Growth Rate)

та частка ринку по містах (KPI Market Share by City), порівнюючи фактичні значення з поставленими цілями та візуалізуючи статус і тренд.

**3.4.2 Розробка аналітичних модулів на Python.** Для вирішення завдань, що виходять за межі можливостей стандартного OLAP-аналізу, таких як кластеризація, класифікація та прогнозування, було розроблено набір аналітичних модулів мовою програмування Python. Цей вибір обґрунтований наявністю потужних спеціалізованих бібліотек та гнучкістю мови для реалізації складних алгоритмів машинного навчання.

Загальний потік роботи для кожного модуля є уніфікованим:

1. Встановлення з'єднання з базою даних за допомогою бібліотеки SQLAlchemy [\[15\]](#).
2. Виконання SQL-запиту та завантаження даних у DataFrame бібліотеки Pandas [\[10\]](#).
3. Попередня обробка та очищення даних.
4. Застосування відповідної моделі з бібліотек Scikit-learn [\[9\]](#) або Statsmodels [\[11\]](#).
5. Візуалізація результатів за допомогою Matplotlib [] та Seaborn.

**Ключові бібліотеки, що використовувались:**

- **Pandas.** Основний інструмент для маніпуляції даними. Використовувався для завантаження даних з SQL, їх очищення, агрегації, створення нових ознак та підготовки до аналізу.
- **Scikit-learn.** Фундаментальна бібліотека для машинного навчання. З неї було використано:
  - KMeans для реалізації кластеризації.
  - BernoulliNB для реалізації класифікатора Наївного Байєса.
  - RandomForestRegressor для визначення важливості факторів.
  - LabelEncoder та StandardScaler для попередньої обробки ознак.

- **Statsmodels.** Спеціалізована бібліотека для статистичного моделювання та аналізу часових рядів. Використовувалась для реалізації моделі ExponentialSmoothing.
- **Mlxtend [12].** Бібліотека, що містить ефективні реалізації алгоритмів інтелектуального аналізу даних. Використовувалась для пошуку асоціативних правил за допомогою **apriori**.
- **Matplotlib** та **Seaborn.** Бібліотеки для статичної візуалізації даних, за допомогою яких було створено всі графіки та діаграми.

Нижче наведено опис реалізації ключових аналітичних модулів.

### Модуль кластеризації вакансій

**Сегментація ринку праці** на групи за рівнем заробітної плати для виявлення прихованих структур та типових посад у кожному сегменті.

**Метод аналізу – кластеризація K-Means.** Це ітеративний алгоритм, який прагне розділити  $n$  спостережень на  $k$  кластерів. Його робота полягає у наступному:

- Випадковим чином обираються  $k$  точок (центроїдів).
- Кожне спостереження (вакансія) відноситься до найближчого центроїда.
- Центроїди перераховуються як центр мас (середнє) всіх спостережень, що потрапили до кластера.
- Кроки 2 і 3 повторюються доти, доки центроїди не перестануть змінюватися.

1. З таблиці **Staging\_Jobs** завантажуються дані про назви вакансій та їх мінімальну й максимальну зарплату.

2. Створюється нова ознака **AvgSalary** як середнє арифметичне між мінімальною та максимальною зарплатою. Рядки без валідних даних про зарплату видаляються.

3. Для підготовки даних до кластеризації ознака **AvgSalary** нормалізується за допомогою **StandardScaler** з бібліотеки **Scikit-learn**, що приводить всі значення до єдиного масштабу.

4. Для визначення оптимальної кількості кластерів ( $k$ ) було використано метод "силуету", який показав, що  $k=3$  є найбільш вдалим значенням, що дозволяє інтерпретувати кластери як "низький", "середній" та "високий" зарплатні сегменти.

5. Навчена модель **KMeans** застосовується до підготовлених даних, і кожній вакансії присвоюється мітка відповідного кластера.

#### Фрагмент коду реалізації кластеризації:

```

1 from sklearn.cluster import KMeans
2 from sklearn.preprocessing import StandardScaler
3
4 # ... завантаження та очищення даних ...
5
6 # Підготовка даних
7 features = df[["AvgSalary"]].values
8 scaler = StandardScaler()
9 scaled_features = scaler.fit_transform(features)
10
11 # Застосування моделі
12 optimal_k = 3
13 kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init="auto")
14 clusters = kmeans.fit_predict(scaled_features)
15 df["Cluster"] = clusters

```

Рис. 3.11 Фрагмент коду кластеризації

#### Модуль класифікації попиту

Імовірність "високого" (H) або "низького" (L) попиту для кожної комбінації "локація-категорія", що дозволяє оцінювати рівень вимог та конкуренції в різних сегментах ринку.

Метод аналізу – Класифікатор Наївного Байєса (BernoulliNB). Цей імовірнісний класифікатор базується на теоремі Байєса і "наївно" припускає, що

всі ознаки (в нашому випадку Location та Category) є незалежними. Він розраховує ймовірність належності до класу ('H' або 'L'), виходячи з того, як часто кожна ознака зустрічалася з кожним класом у навчальних даних.

1. З таблиці **Staging\_Jobs** завантажуються дані про локації, категорії та навички для кожної вакансії.

2. Створюється **цільова змінна Class**. Вакансія відноситься до класу 'H' (High demand), якщо кількість необхідних навичок для неї вища за середнє значення по всій вибірці, і до класу 'L' (Low demand) в іншому випадку.

3. Категоріальні ознаки **Location** та **Category** перетворюються у числовий формат за допомогою **LabelEncoder**.

4. Для класифікації використовується модель **BernoulliNB (Наївний класифікатор Байеса)**, яка добре підходить для бінарної класифікації з дискретними ознаками. Модель навчається на підготовлених даних.

5. На основі навченої моделі розраховуються умовні ймовірності, які і є результатом аналізу попиту.

**Фрагмент коду реалізації класифікатора Наївного Басса:**

```

1 # Імпорт основного алгоритму класифікації
2 from sklearn.naive_bayes import BernoulliNB
3
4 # Визначення факторів (X) та цільової змінної (y)
5 X = df[["LocationEncoded", "CategoryEncoded"]]
6 y = df["Class"]
7
8 # Розділення даних на тренувальний та тестовий набори
9 X_train, X_test, y_train, y_test = train_test_split(
10     X, y, test_size=0.2, random_state=42, stratify=y
11 )
12
13 # Тренування моделі на навчальних даних
14 self.model.fit(X_train, y_train)
15
16 # Прогнозування на тестових даних
17 y_pred = self.model.predict(X_test)

```

Рис. 3.12 Фрагмент коду класифікатора

### Модуль аналізу кількості вакансії

**Прогнозування кількості унікальних вакансій** на майбутні періоди для виявлення загальних трендів.

1. З вихідних даних вилучаються унікальні вакансії за комбінацією **Title** та **Employer**.

2. На основі дат публікації цих унікальних вакансій створюється **часовий ряд** з обраною частотою (наприклад, щомісячно).

3. Критично важливий крок: для забезпечення точності прогнозу з часового ряду видаляється останній запис, якщо він відповідає неповному періоду (наприклад, дані за жовтень, зібрані лише до 13-го числа).

4. До підготовленого часового ряду застосовується модель **ExponentialSmoothing** з бібліотеки **Statsmodels**. Було обрано модель з адитивним трендом без сезонності, оскільки наявний часовий проміжок є недостатнім для виявлення річних сезонних патернів.

5. За допомогою методу **.predict()** навченої моделі генерується прогноз на задану кількість майбутніх періодів.

#### Фрагмент коду реалізації прогнозування:

```

1 from statsmodels.tsa.holtwinter import ExponentialSmoothing
2
3 # ... підготовка часового ряду time_series_data ...
4
5 # Видаляємо останній неповний період
6 time_series_for_forecast = time_series_data[:-1]
7
8 # Навчаємо модель та робимо прогноз
9 model = ExponentialSmoothing(time_series_for_forecast, trend='add')
10 fitted_model = model.fit()
11 forecast = fitted_model.predict(start=len(time_series_for_forecast), ...)

```

Рис. 3.13 Фрагмент коду прогнозування

#### Модуль аналізу тривалості вакансії

Визначення ключових факторів, що впливають на **тривалість перебування вакансії на ринку**.

**Метод аналізу – випадковий ліс (RandomForest)**. Це ансамблевий метод, який будує велику кількість "дерев рішень" і усереднює їхні результати. Його ключовою перевагою є можливість оцінювати **важливість ознак (Feature Importance)**. Він визначає, наскільки сильно кожна ознака (зарплата, кількість навичок, локація) в середньому впливала на зменшення невизначеності при побудові всіх дерев в "лісі".

1. Скрипт (`duration_analysis.py`) завантажує дані та розраховує тривалість життя кожної унікальної вакансії (`ClosingTimeDays`).

2. Формується набір даних, де кожному показнику тривалості відповідають середні значення AvgSalary, SkillCount та Location за цей період.
3. Модель RandomForestRegressor навчається прогнозувати ClosingTimeDays на основі цих трьох факторів.
4. Після навчання з моделі витягується атрибут feature\_importances, який показує відносний внесок кожного фактора у точність прогнозу. Ці значення візуалізуються.

### Фрагмент коду реалізації методу:

```

1 from sklearn.ensemble import RandomForestRegressor
2 from sklearn.preprocessing import LabelEncoder
3
4 # Перетворюємо категоріальну ознаку 'Location' у числову
5 le = LabelEncoder()
6 features_df["LocationEncoded"] = le.fit_transform(features_df["Location"])
7
8 # Визначаємо X (ознаки) та y (цільова змінна)
9 X = features_df[["AvgSalary", "SkillCount", "LocationEncoded"]]
10 y = features_df["ClosingTimeDays"]
11
12 # Навчаємо модель для визначення важливості ознак
13 model = RandomForestRegressor(n_estimators=100, random_state=42)
14 model.fit(X, y)

```

Рис. 3.14 Фрагмент коду аналізу

### Модуль аналізу асоціативних правил

Пошук наборів навичок, які часто вимагаються разом у вакансіях.

**Метод аналізу – алгоритм Apriori.** Це класичний алгоритм для пошуку "частих наборів" у транзакційних даних. Його робота складається з двох етапів:

- **Пошук частих наборів.** Алгоритм ітеративно знаходить набори навичок, які зустрічаються частіше за заданий поріг (min\_support).
- **Генерація правил.** На основі цих частих наборів генеруються правила виду "Якщо є Набір А, то також є Набір Б" з довір'ям (confidence) не нижче заданого порогу.

- Скрипт (`association_rules.py`) завантажує дані з `Dim_Skills` (для валідації) та `Staging_Jobs`.
- Для кожної вакансії створюється "транзакція" — список валідних навичок.
- За допомогою `TransactionEncoder` з бібліотеки `mlxtend` дані перетворюються у формат, придатний для алгоритму.
- Функція `apriori` знаходить часті набори, а функція `association_rules` генерує правила, які потім фільтруються та візуалізуються.

### Фрагмент коду реалізації:

```
1 from mlxtend.preprocessing import TransactionEncoder
2 from mlxtend.frequent_patterns import apriori, association_rules
3
4 te = TransactionEncoder()
5 te_ary = te.fit(transactions).transform(transactions)
6 df_encoded = pd.DataFrame(te_ary, columns=te.columns_)
7
8 frequent_itemsets = apriori(df_encoded, min_support=min_support, use_colnames=True)
9
10 rules = association_rules(
11     frequent_itemsets, metric="confidence", min_threshold=min_confidence
12 )
13 rules["antecedents"] = rules["antecedents"].apply(lambda a: ", ".join(list(a)))
14 rules["consequents"] = rules["consequents"].apply(lambda a: ", ".join(list(a)))
```

Рис. 3.15 Фрагмент коду апріорі

## 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ РИНКУ ПРАЦІ

### 4.1 Аналіз ключових викликів ринку праці

Перше завдання дослідження полягало в ідентифікації дисбалансів на ринку праці, зокрема невідповідності навичок та попиту-пропозиції.

Для виявлення "розриву в навичках" було проведено порівняльний аналіз наборів умінь, що вимагаються для **наймасовіших** та **найбільш високооплачуваних** вакансій (Таблиця 4.1). Результати показали існування **967 навичок**, характерних для масового ринку (напр., 1С, Вміння працювати в команді), та **426 дефіцитних навичок**, що частіше зустрічаються у високооплачуваних вакансіях (напр., .NET Framework, API development, Machine Learning). Це підтверджує гіпотезу про наявність значного розриву між загальними та вузькоспеціалізованими компетенціями.

Таблиця 4.1

#### Наймасовіші та дефіцитні навички

Категорія навичок	Навичка
<b>Масові навички</b>	1С: Управління торгівлею
<b>Масові навички</b>	3D-моделювання
<b>Масові навички</b>	A/B-тестування
<b>Масові навички</b>	Adobe InDesign
<b>Масові навички</b>	Arduino
<b>Масові навички</b>	AutoCAD
<b>Масові навички</b>	Brand management
<b>Масові навички</b>	Burp Suite
<b>Масові навички</b>	CCNA
<b>Масові навички</b>	Ability to work in multitasking mode
<b>Дефіцитні навички</b>	ASP.NET Core
<b>Дефіцитні навички</b>	Accounting management
<b>Дефіцитні навички</b>	Adobe Creative Cloud
<b>Дефіцитні навички</b>	Android
<b>Дефіцитні навички</b>	Architectural visualization
<b>Дефіцитні навички</b>	Automation testing
<b>Дефіцитні навички</b>	Automotive SPICE

Категорія навичок	Навичка
Дефіцитні навички	Azure Security
Дефіцитні навички	BI tools
Дефіцитні навички	API development

Для подальшого аналізу було проведено автоматизовану класифікацію всіх унікальних навичок на "Hard" та "Soft". Розподіл показує значне домінування технічних навичок у вимогах роботодавців (рис. 4.1).

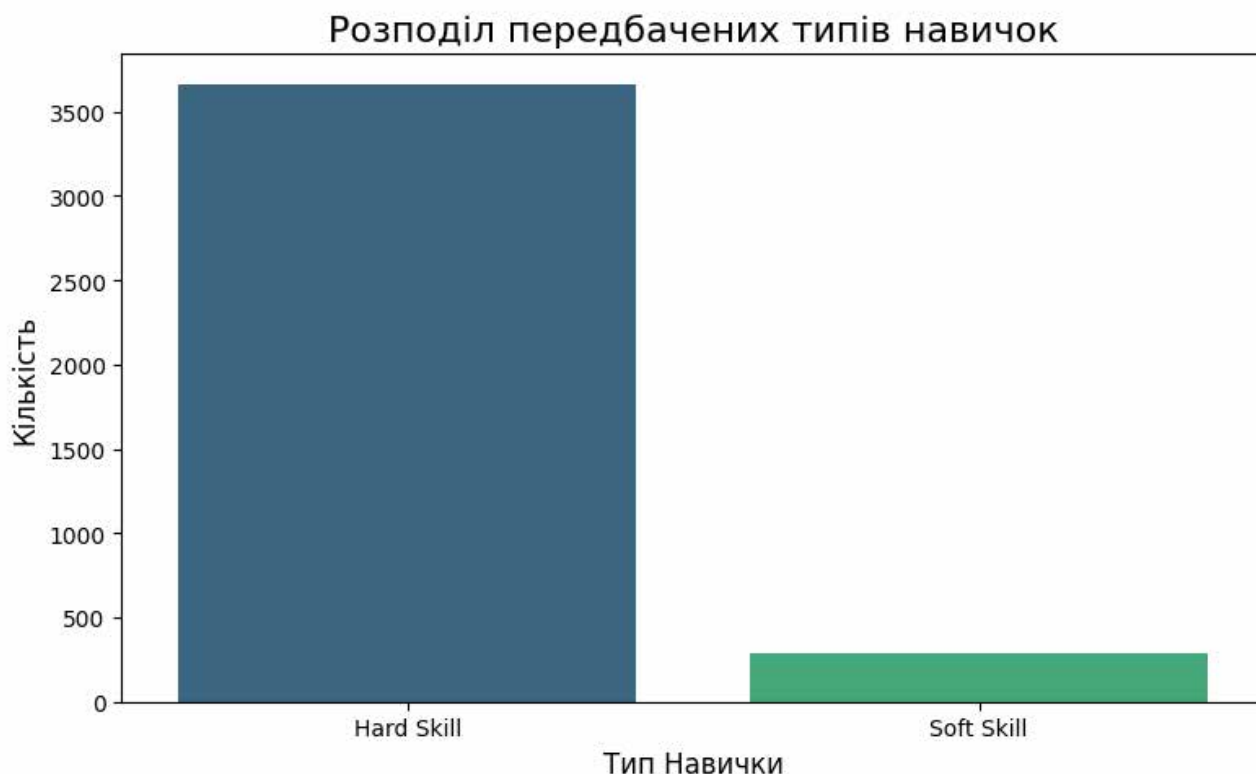


Рис. 4.1 Розподіл навичок за типами (Hard/Soft)

Крім того, за допомогою алгоритму Apriori було знайдено асоціативні правила, що показують, які навички найчастіше вимагаються разом (Таблиця 4.2). Аналіз правил демонструє існування стійких технологічних стеків, наприклад, у сфері SEO (SEMrush, Ahrefs, Google Analytics) та мережевих технологій (DHCP, DNS, VPN).

Таблиця 4.2

## Топ-15 асоціативних правил

Антецедент	Консеквент	support	confidence	lift
<b>Google Ads, Google Tag Manager</b>	Google Shopping	0.0054	0.2478	33.547
<b>Active Directory, Налаштування DNS</b>	DHCP	0.0076	0.8489	32.1922
<b>Ahrefs, Google Analytics, Google Search Console</b>	SEMrush	0.0063	0.4434	31.9815
<b>Ahrefs, Google Analytics</b>	Google Search Console	0.0143	0.8667	30.0587
<b>Ahrefs</b>	SEMrush	0.0135	0.4147	29.9075
<b>LAN, DHCP</b>	Налаштування DNS	0.0062	0.9231	29.7427
<b>IP (Internet Protocol), DHCP</b>	Налаштування DNS	0.0144	0.9098	29.3161
<b>DHCP</b>	Налаштування DNS	0.0235	0.8919	28.7379
<b>Ahrefs, Аналітичне мислення</b>	Google Search Console	0.0056	0.7909	27.4312
<b>Налаштування VPN</b>	VLAN	0.0075	0.2805	27.399
<b>Serpstat</b>	Ahrefs	0.0094	0.8631	26.4306
<b>Налаштування маршрутизаторів</b>	VLAN	0.0053	0.268	26.1766
<b>Google Shopping</b>	Google Tag Manager	0.0055	0.7456	25.6299
<b>Google Ads, Looker Studio</b>	Google Tag Manager	0.006	0.736	25.2994
<b>Налаштування VPN</b>	Налаштування Firewall	0.0058	0.2171	25.1903

За допомогою класифікатора Наївного Байєса було розраховано ймовірність "високого" (H) або "низького" (L) попиту, де попит визначався за середньою кількістю навичок у вакансії.

Графік (рис. 4.2) демонструє значний регіональний дисбаланс. "Дистанційна робота" та "Київ" мають високу ймовірність високого попиту, що вказує на складніші вимоги. Водночас у таких містах, як Харків та Чернівці, значно переважає попит на вакансії з меншою кількістю навичок (клас 'L').

Аналогічний аналіз за категоріями (рис. 4.3) показує, що найвищі вимоги до кількості навичок (високий попит) спостерігаються в категоріях "Маркетинг, реклама, PR" та "ІТ, комп'ютери, інтернет".

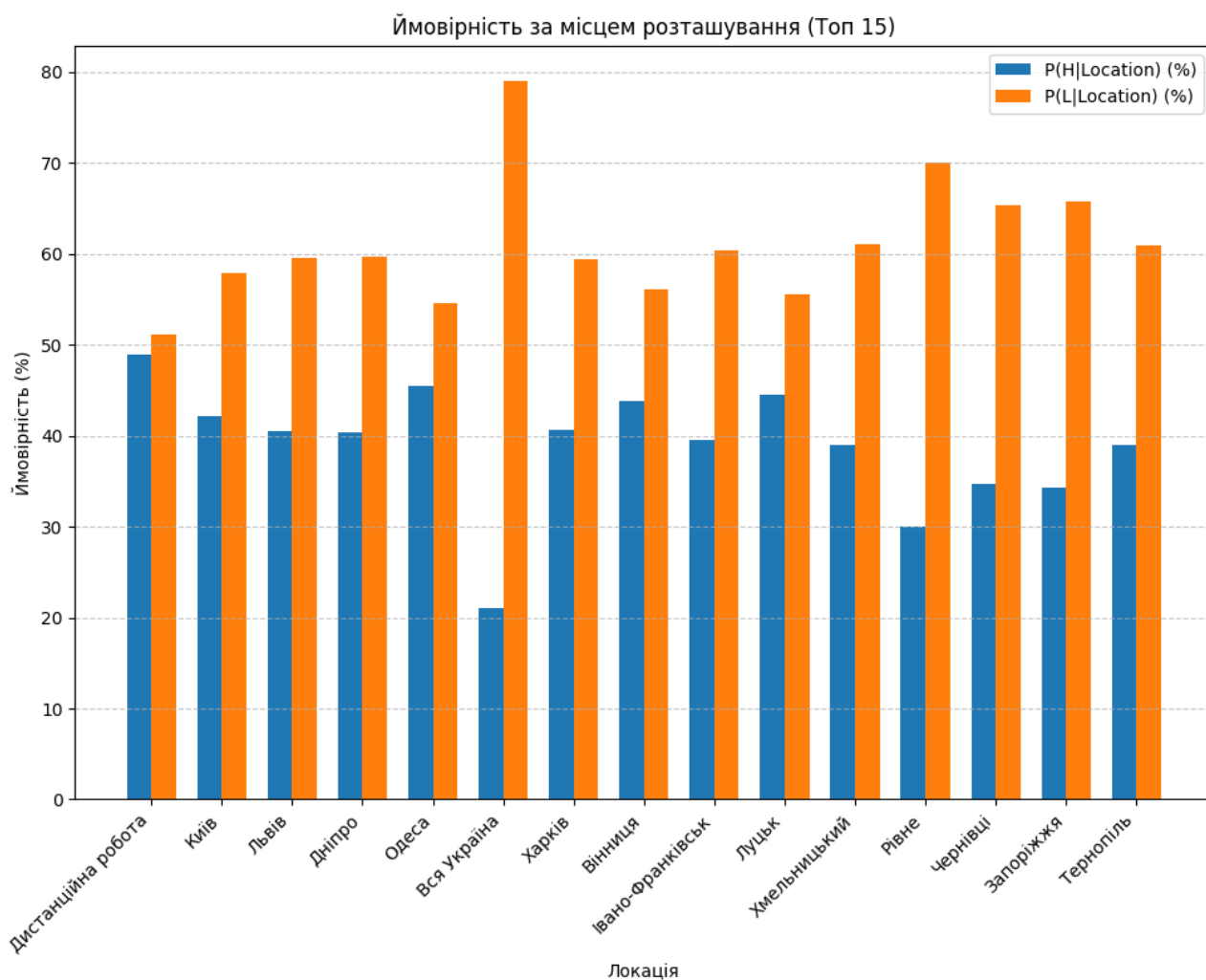


Рис. 4.2 Ймовірність високого (H) та низького (L) попиту за локаціями (Топ-15)

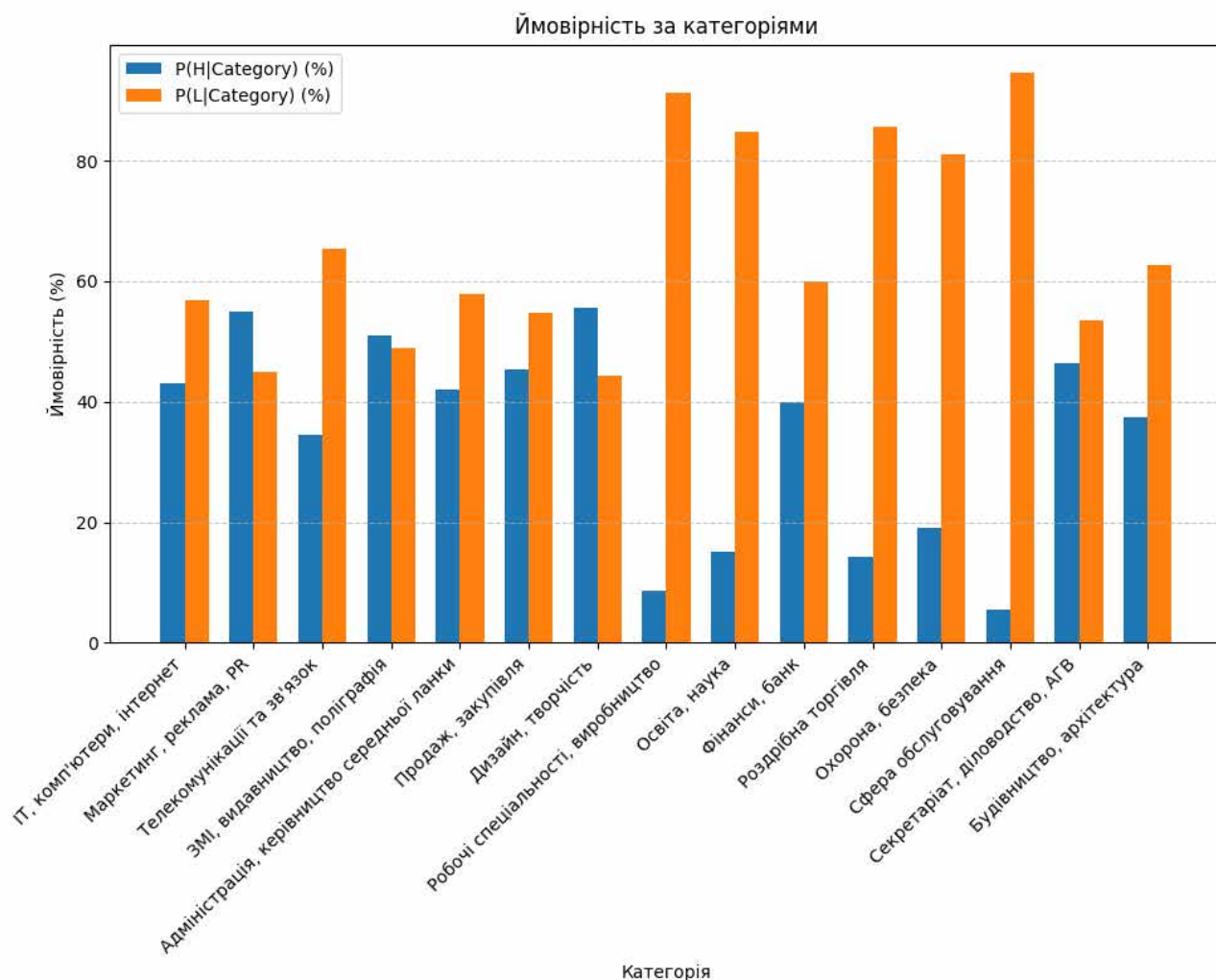


Рис. 4.3 Імовірність високого (H) та низького (L) попиту за категоріями

## 4.2 Дослідження прихованих тенденцій, кластеризація та прогнозування

Друге завдання полягало у виявленні прихованих тенденцій за допомогою кластеризації та прогнозування.

Застосування алгоритму **K-Means** дозволило автоматично сегментувати ринок на три чіткі зарплатні групи.

Графік (рис. 4.4) та таблиця (рис. 4.5) чітко демонструють три зарплатні сегменти: **Сегмент 0** (низький, до ~50 тис. грн), **Сегмент 1** (середній, ~50-100 тис. грн) та **Сегмент 2** (високий, понад 100 тис. грн). Було виявлено, що середній

зарплатний сегмент значною мірою складається з військових спеціальностей, що є яскравим свідченням впливу зовнішніх факторів на ринок праці.

Таблиця 4.3

## Сегменти заробітних плат

Сегмент	Кількість вакансій	Зарплатний діапазон	Найчастіші позиції
Сегмент 0	10822	2,000 - 47,850 грн	SMM-менеджер, Контент-менеджер, Системний адміністратор, Таргетолог, SMM-спеціаліст
Сегмент 1	2498	48,000 - 115,500 грн	SMM-менеджер, Системний адміністратор, Програміст 1С, Таргетолог, Project-менеджер
Сегмент 2	236	118,000 - 600,000 грн	DevOps engineer, Fullstack-розробник, Senior DevOps Engineer, Media Buyer (Facebook), Керівник бізнес-аналітиків (1С)

Розподіл зарплат за сегментами

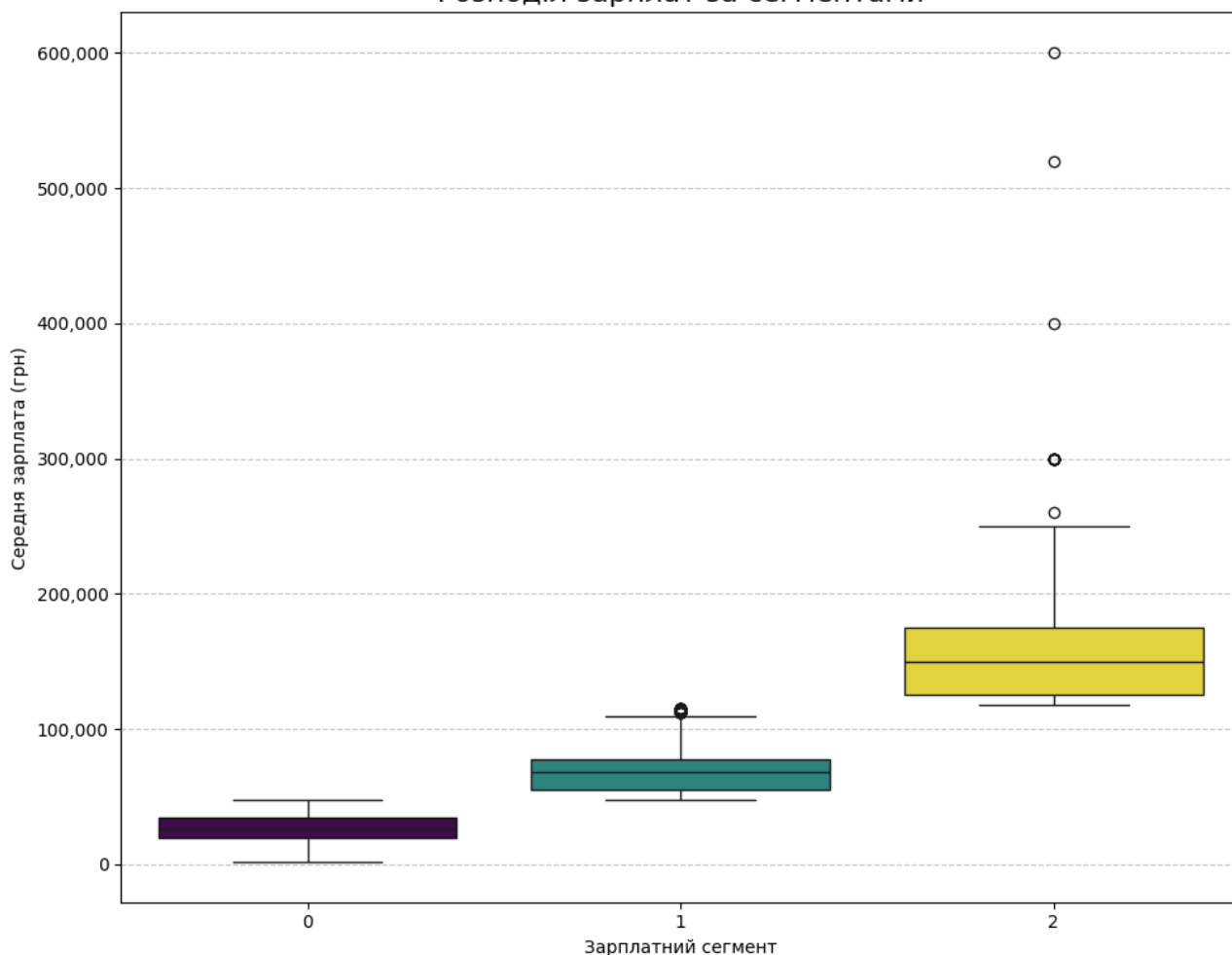


Рис. 4.4 Розподіл заробітних плат за сегментами

На основі часового ряду кількості унікальних вакансій було створено прогнози на майбутні періоди.

Графіки (рис. 4.5-4.6) демонструє сезонні коливання активності ринку. Прогноз, побудований за допомогою моделі Exponential Smoothing, передбачає впевнене зростання кількості нових вакансій наприкінці року, незважаючи на нещодавнє зниження.

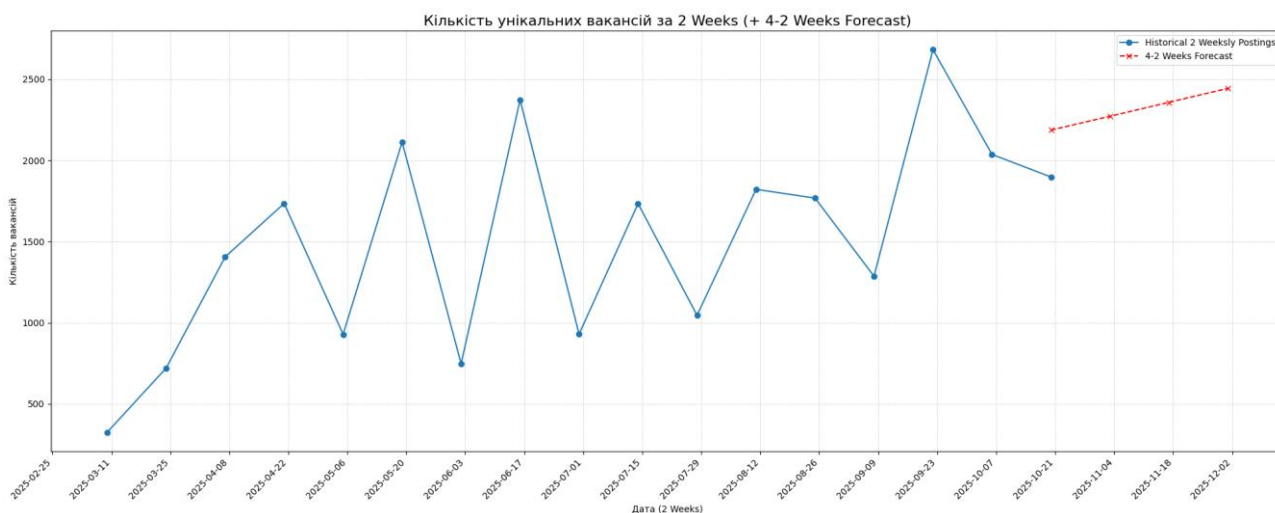


Рис. 4.5 Динаміка та прогноз кількості унікальних вакансій (по 2 тижнях)

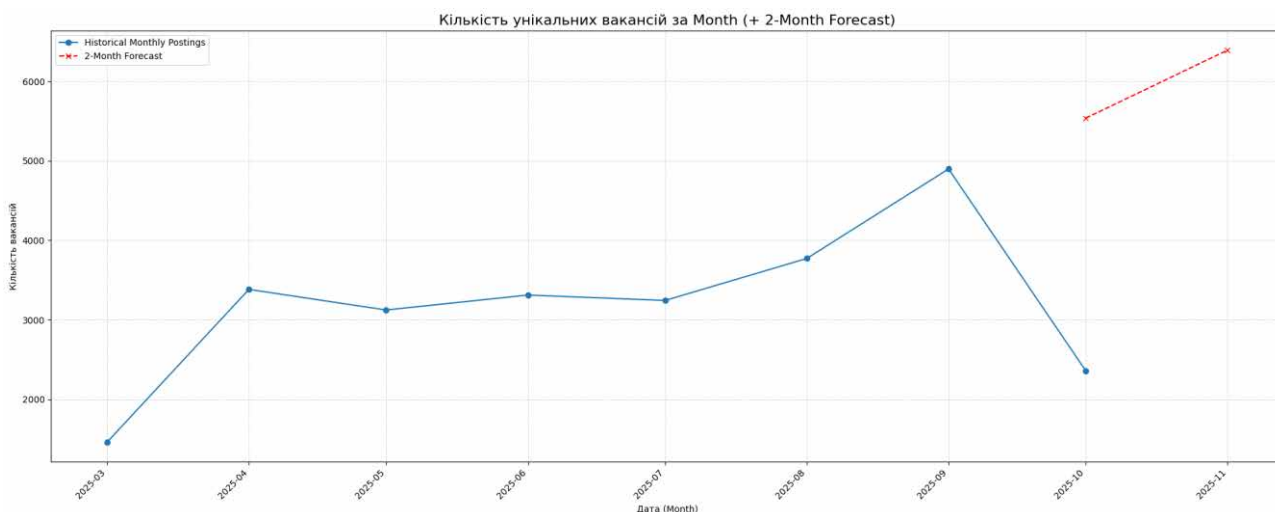


Рис. 4.6 Динаміка та прогноз кількості унікальних вакансій (по місяцях)

### 4.3 Оцінка впливу атрибутів на тривалість вакансії

Для оцінки швидкості закриття вакансій було проаналізовано їх "стійкість" — кількість тижнів, протягом яких вони залишалися активними на ринку.

Гістограма (рис. 4.7) демонструє, що переважна більшість вакансій закриваються протягом першого тижня, що свідчить про високу динаміку ринку. Однак існує "довгий хвіст" вакансій, відкритих понад 10 тижнів. Для виявлення факторів, що впливають на цю тривалість, було застосовано модель RandomForest.

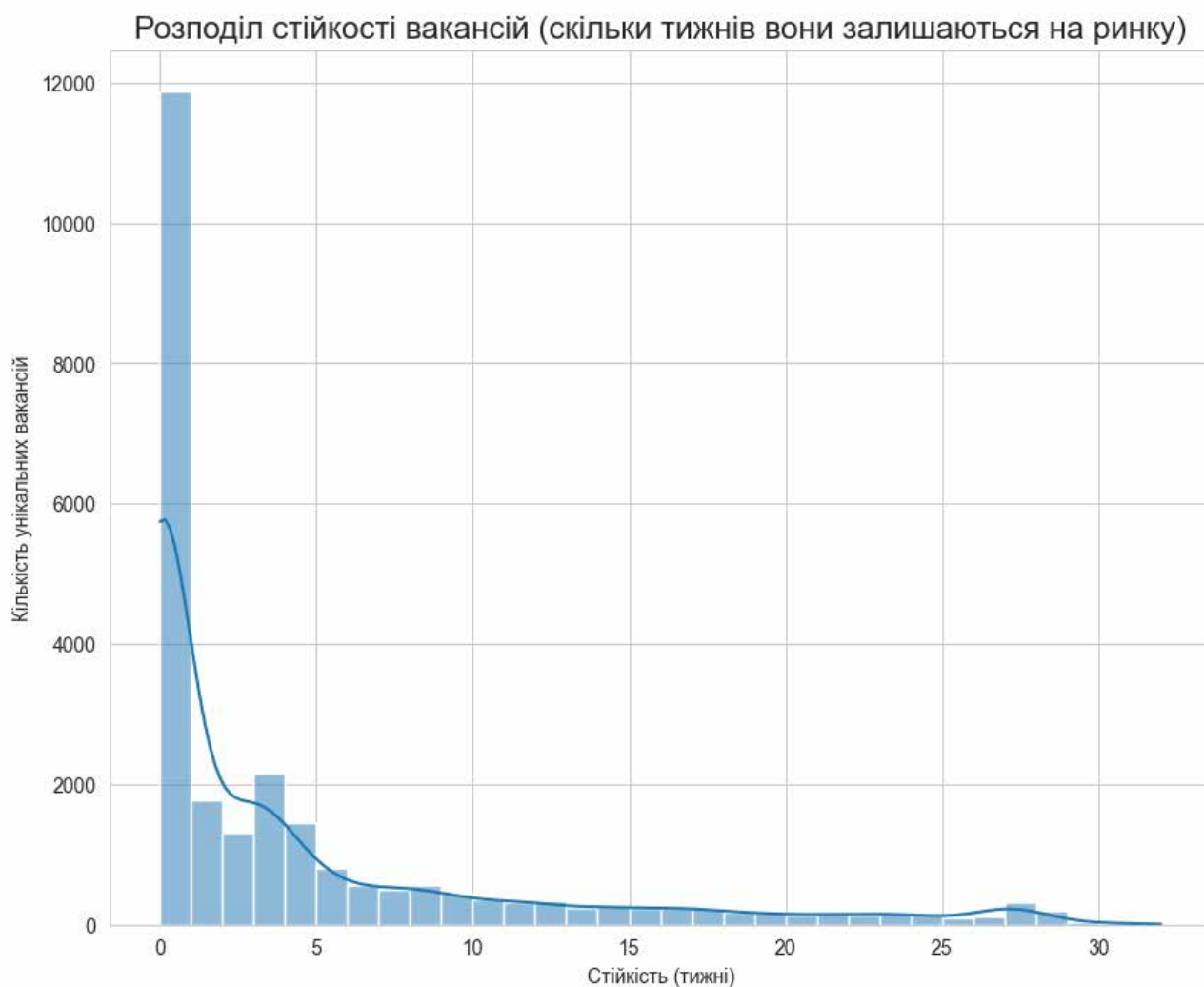


Рис. 4.7 Розподіл стійкості вакансій

Аналіз (рис. 4.8) однозначно показав, що **рівень заробітної плати (AvgSalary)** та **кількість навичок (SkillCount)** є головними факторами, що впливають на тривалість пошуку кандидатів.

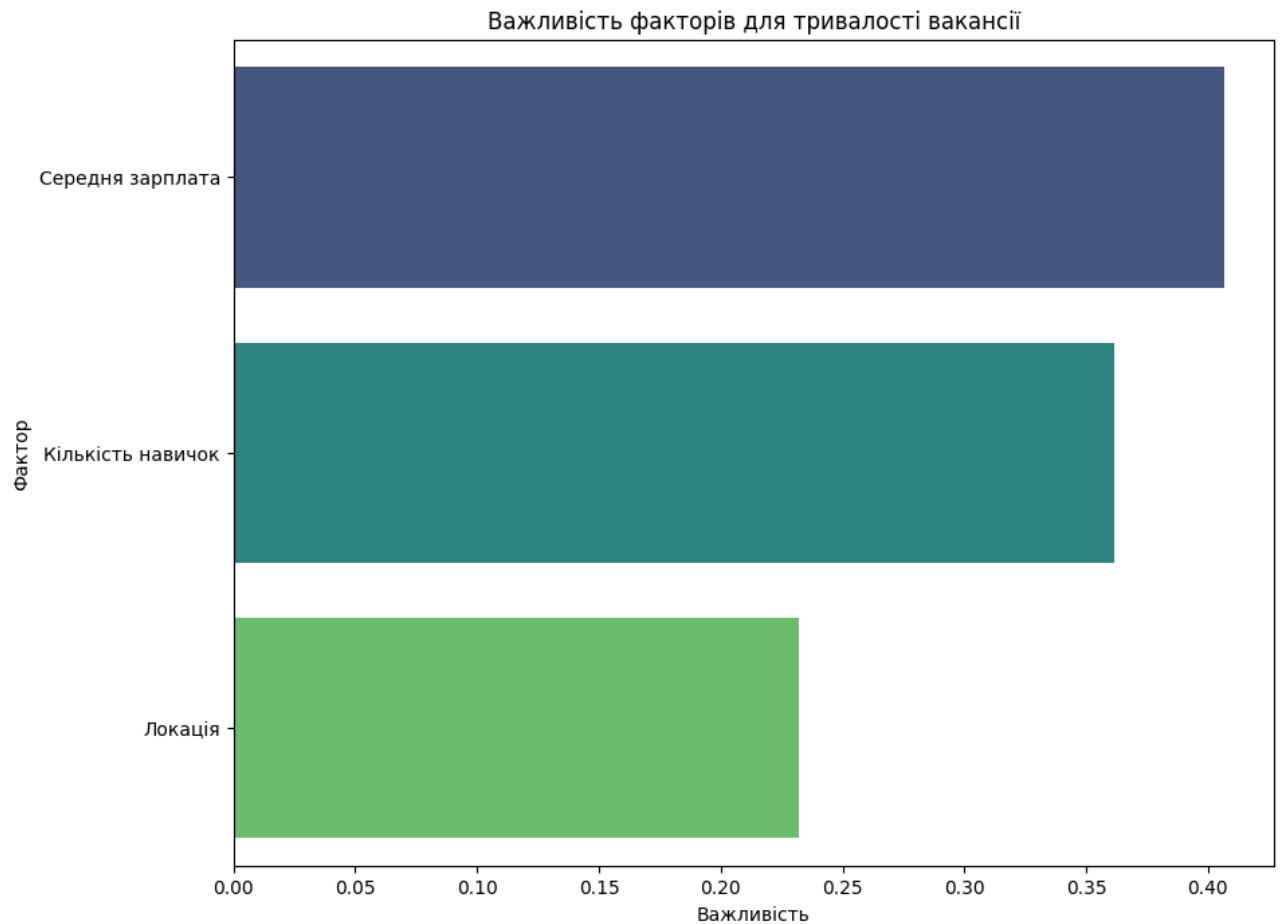


Рис. 4.8 Важливість ключових атрибутів для тривалості вакансії

#### 4.4 Аналіз часових закономірностей за секторами

Для аналізу появи нових вакансій у різних секторах було побудовано теплову карту (рис. 4.9), яка візуалізує сезонні патерни. Чітко видно пік активності в ІТ-секторі у вересні (4897 вакансій), що корелює з початком бізнес-сезону.

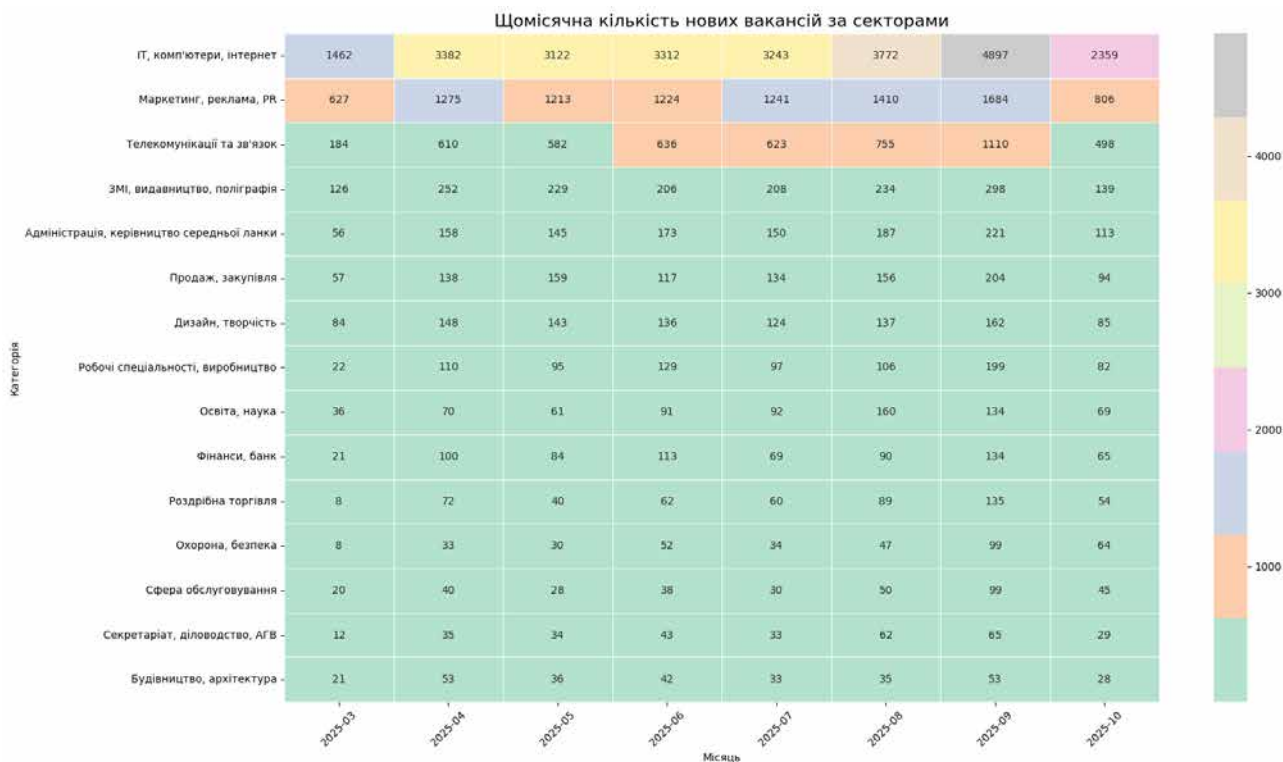


Рис. 4.9 Теплова карта щомісячної активності вакансій за секторами

## 4.5 Інтерактивна візуалізація

Ключовим результатом роботи є створення інтерактивних аналітичних панелей (дашбордів) у середовищі Power BI. На відміну від статичних графіків, дашборди забезпечують можливість глибокого інтерактивного аналізу даних (drill-down, cross-filtering) завдяки прямому підключенню до розробленого OLAP-куба в режимі **Live Connection**. Це означає, що кожна взаємодія користувача з дашбордом (наприклад, вибір певного міста на карті) генерує MDX-запит до сервера SSAS, який миттєво повертає агреговані дані.

Було розроблено два основні дашборди для вирішення ключових бізнес-задач.

Наступний дашборд (рис. 4.10) призначений для швидкої оцінки зарплатних показників та географічного розподілу вакансій.

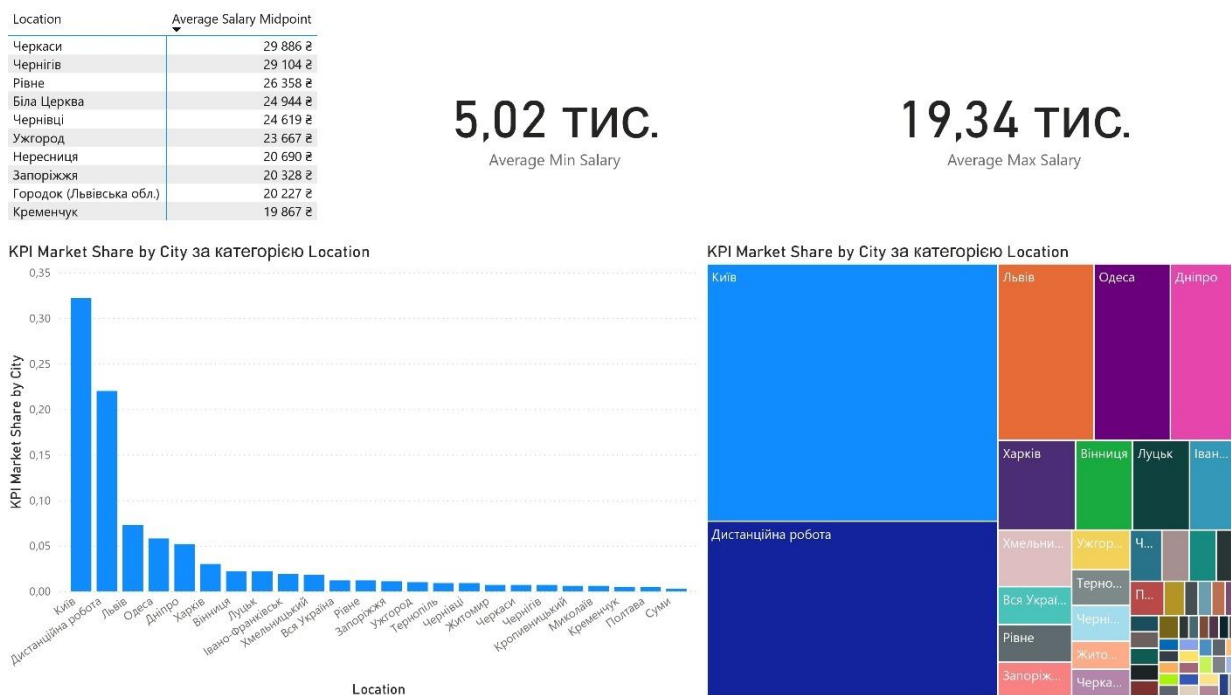


Рис. 4.10 Дашборд для аналізу зарплат та частки ринку за локаціями

Далі наведено компоненти дашборду та їх опис.

- **Карти показників (Card Visuals).** У верхній частині дашборду розташовані дві карти, що відображають **середню мінімальну (5,02 тис. грн)** та **середню максимальну (19,34 тис. грн)** заробітну плату по всій вибірці. Ці показники надають миттєве уявлення про загальний діапазон зарплат на ринку.
- **Стовпчикова діаграма та Treemap ("Частка ринку за локаціями").** Ці два візуальні елементи представляють одну й ту саму метрику — частку кожної локації від загальної кількості вакансій.
  - **Стовпчикова діаграма (зліва)** дозволяє точно порівняти частки. Вона чітко показує, що **Київ** та **Дистанційна робота** є абсолютними лідерами, значно випереджаючи інші регіони.
  - **Treemap-діаграма (справа)** візуалізує домінування за допомогою площі прямокутників. Вона миттєво демонструє, що **Київ** займає найбільшу частину "ринку", за ним слідують **Львів, Одеса, Дніпро** та **Харків**. "Дистанційна робота"

також виділяється як один з найбільших сегментів. Це візуально підтверджує висновок про географічний дисбаланс, зроблений у першому розділі.

- **Таблиця ("Середня зарплата за локаціями").** У верхньому лівому куті розташована таблиця, що дозволяє детально аналізувати середній рівень зарплат у розрізі окремих міст та сортувати їх за цим показником.

Цей інструмент дозволяє HR-аналітику швидко оцінити регіональну конкуренцію, зрозуміти рівень зарплатних очікувань у різних містах та приймати рішення щодо географії пошуку кандидатів.

Дашборд на рисунку 4.11 надає стратегічний огляд ринку, відповідаючи на питання "коли?" та "в яких секторах?" з'являється найбільше вакансій.

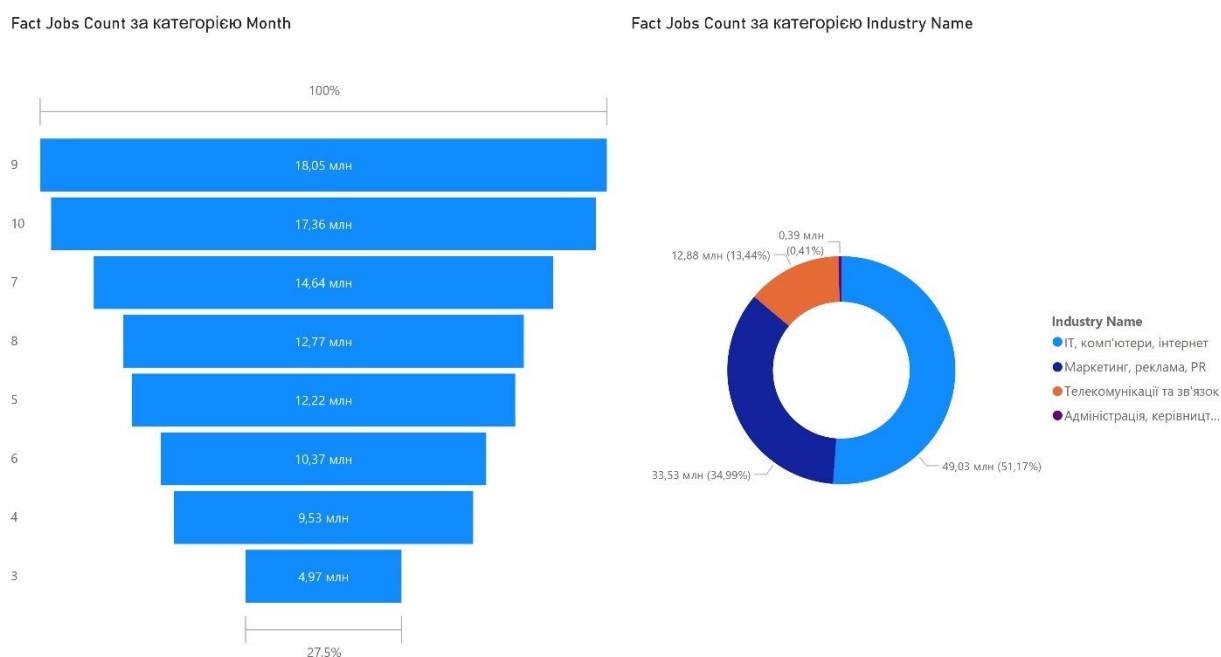


Рис. 4.11 Аналіз розподілу вакансій за місяцями та індустріями

- **Воронкоподібна діаграма ("Розподіл вакансій за місяцями").** Цей візуальний елемент, відсортований за кількістю вакансій, наочно демонструє найактивніші місяці для найму.

- З діаграми видно, що пік активності припадає на **вересень (18,05 млн)** та **жовтень (17,36 млн)**, що корелює з початком бізнес-

сезону після літнього затишшя. Найменш активним місяцем є **березень (4,97 млн)**. Ця інформація є вкрай важливою для планування рекрутингових бюджетів та ресурсів.

- **Кільцева діаграма ("Розподіл вакансій за індустріями")**. Ця діаграма показує структуру ринку в розрізі галузей.
  - Вона однозначно підтверджує домінування двох ключових секторів у досліджуваній вибірці. **"ІТ, комп'ютери, інтернет"** займає **51%** ринку, а **"Маркетинг, реклама, PR"** — **35%**. Разом ці дві галузі складають понад 86% всіх вакансій, що свідчить про високу концентрацію ринку та специфіку джерела даних.

Цей інструмент дозволяє керівникам та HR-директорам зрозуміти макротренди ринку, визначити сезонні патерни та оцінити, в яких галузях спостерігається найбільша активність, що допомагає у прийнятті стратегічних рішень.

Окрім дашбордів у Power BI, для глибокого технічного аналізу було використано вбудований браузер OLAP-кубів у Visual Studio. Цей інструмент дозволяє аналітику безпосередньо взаємодіяти з даними куба, перевіряти коректність розрахунків та аналізувати KPI у різних зрізах (slice and dice).

На рисунку 4.12 представлено приклад такого аналізу, де розраховані KPI відображаються в розрізі вимірів **"Час" (Month)** та **"Географія" (Location)**.

Month	KPI Salary Growth Rate	KPI Salary Growth Rate – ціль	KPI Salary Growth Rate – стан	KPI Salary Growth Rate – тенденція
10	5,79	5,00	●	↑
3	–нескінченність	5,00	○	↓
4	2,99	5,00	○	↑
5	-4,68	5,00	○	↓
6	-0,19	5,00	○	↑
7	0,26	5,00	○	↑
8	-1,82	5,00	○	↓
9	-1,12	5,00	○	↑
<b>Усього</b>	<b>Нескінченність</b>	<b>5,00</b>	<b>●</b>	<b>↑</b>

Month	KPI AVG Salary	KPI AVG Salary – ціль	KPI AVG Salary – стан	KPI AVG Salary – тенденція
3	-1 647,48	8 183,36	○	↓
4	-6 577,41	8 183,36	○	↓
5	24 174,74	8 183,36	●	↑
6	19 497,42	8 183,36	●	↓
7	24 478,75	8 183,36	●	↑
8	-20 193,41	8 183,36	○	↓
9	2 467,21	8 183,36	○	↑
10	16 760,68	8 183,36	●	↑
<b>Усього</b>	<b>14 689,78</b>	<b>8 183,36</b>	<b>●</b>	<b>↑</b>

Location	KPI AVG Salary	KPI AVG Salary – ціль	KPI AVG Salary – стан	KPI AVG Salary – тенденція
Черкаси	29 886,23	6 641,39	●	↑
Чернігів	29 103,79	6 467,51	●	↑
Рівне	26 357,60	5 857,24	●	↑
Біла Церква	24 943,92	6 235,98	●	↑
Чернівці	24 618,69	7 930,30	●	↑
Ужгород	23 667,31	5 259,40	●	↑
Нересниця	20 689,63	20 689,63	●	↑
Запоріжжя	20 327,72	6 976,75	●	↑
Городок (Львівська обл.)	20 226,78	8 090,71	●	↑
Кременчук	19 867,28	4 966,82	●	↑
Дубно	18 405,89	6 135,30	●	↑
<b>Усього</b>	<b>14 689,78</b>	<b>8 183,36</b>	<b>●</b>	<b>↑</b>

Рис. 4.12 Аналіз реалізованих KPI

- **Аналіз у розрізі часу за місяцями:**
  - **KPI Salary Growth Rate:** Цей показник розраховує відсоткове зростання середньої зарплати порівняно з попереднім місяцем. Значення "**нескінченність**" для першого періоду (місяць 3) є очікуваним, оскільки для нього відсутній попередній період для порівняння. Подальші значення (напр., **2,99** для 4-го місяця) показують реальну динаміку. Індикатори тренду візуалізують, чи зростала зарплата порівняно з попереднім місяцем.
    - **KPI AVG Salary:** У цьому зрізі значення показника відображають **відхилення** середньої зарплати за місяць від середньорічної цілі. Наприклад, значення **24 174,74** для 5-го місяця означає, що середня зарплата в цьому місяці була значно вищою за цільовий показник. Негативні значення (напр., **-6 577,41** для 4-го місяця) вказують на те, що зарплата була нижчою за ціль.
- **Аналіз у розрізі географії (Location):**
  - **KPI AVG Salary:** У цьому зрізі показник відображає абсолютне значення середньої зарплати для кожного міста. Наприклад, для **Черкас** середня

зарплата становить **29 886**. Ціль може бути встановлена як середнє значення по всій країні, що дозволяє миттєво оцінити, в яких містах зарплати вищі або нижчі за середній рівень. Індикатори статусу та тренду візуалізують це порівняння.

## ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було розроблено та реалізовано аналітичну систему управління працевлаштуванням. Система поєднує технології сховищ даних та методи інтелектуального аналізу даних для проведення глибокого аналізу сучасного ІТ ринку праці України. За результатами проведеного дослідження було зроблено наступні висновки.

- **Спроектовано та реалізовано аналітичну архітектуру.**

Було успішно розроблено архітектуру системи, що складається з кількох взаємопов'язаних компонентів – **сховища даних**, яке забезпечує високу продуктивність аналітичних запитів; для автоматизації збору, очищення та завантаження даних розроблено **ETL-процеси** за допомогою SQL Server Integration Services; на основі сховища створено **OLAP-куб**, що дозволяє проводити швидкий інтерактивний аналіз та розраховувати ключові показники ефективності (**KPI**); для задач, що виходять за межі OLAP, було реалізовано модулі на Python, що підтверджує ефективність обраного гібридного підходу.

- **Ідентифіковано ключові виклики та дисбаланси на ринку праці.**

Застосовані методи інтелектуального аналізу дозволили кількісно оцінити основні виклики сучасного ринку.

- Аналіз "**розриву в навичках**" підтвердив існування значної невідповідності між навичками, що вимагаються для масових та високооплачуваних вакансій. Було ідентифіковано **967** навичок, характерних для масового ринку, та **426** дефіцитних навичок (вузькоспеціалізовані технології, такі як Golang, Kubernetes, Machine Learning), що вказує на напрямки для професійного розвитку.

- **Класифікація попиту** за допомогою методу Наївного Байєса виявила значний **географічний дисбаланс** – найвищі вимоги до кандидатів спостерігаються для віддалених позицій та у столиці, тоді як в інших регіонах

переважає попит на вакансії з меншим набором навичок. Це підтверджує гіпотезу про концентрацію висококваліфікованих ролей у великих містах та для віддаленого формату.

- **Виявлено приховані тенденції та розроблено прогноз динаміки ринку.**

- Кластеризація вакансій за допомогою алгоритму **K-Means** дозволила автоматично сегментувати ринок на три чіткі зарплатні групи. Найважливішим висновком цього аналізу стало виявлення значного впливу зовнішніх факторів на структуру ринку – у середньому зарплатному сегменті (середня зарплата ~ 48 - 115 тис. грн) 9% вакансій становлять військові спеціальності. Це демонструє, як методи **Data Mining** здатні виявляти неочевидні соціально-економічні тенденції.

- Прогнозування часових рядів за допомогою моделі **Exponential Smoothing** показало наявність сезонних коливань ринку праці та дозволило побудувати короткостроковий прогноз. Модель прогнозує впевнене зростання кількості нових вакансій наприкінці року, що є цінною інформацією для стратегічного планування в HR.

- **Проведено оцінку впливу ключових атрибутів на тривалість вакансії.**

Аналіз тривалості перебування вакансії на ринку показав, що більшість позицій закриваються протягом першого тижня, що свідчить про високу динаміку ринку. Однак, за допомогою моделі регресії **Random Forest** було кількісно визначено фактори, що найбільше впливають на тривалість пошуку. Аналіз показав, що **рівень заробітної плати** та **кількість навичок** є домінуючими факторами, значно випереджаючи вплив локації. Це дозволяє зробити практичний висновок: для прискорення закриття вакансії найбільш ефективними важелями є коригування зарплатної пропозиції та вимог до навичок.

У висновку з вищеописаного було успішно досягнуто поставленої мети. Розроблено та реалізовано програмне забезпечення аналітичної системи, яка довела свою ефективність у вирішенні ключових завдань аналізу ринку праці. Поєднання технологій сховищ даних (DWH, OLAP) та методів інтелектуального аналізу даних (Data Mining, Machine Learning) дозволило не лише структурувати та візуалізувати дані, але й виявити приховані, нетривіальні та практично значущі закономірності.

Отримані результати та розроблена система можуть бути використані HR-відділами компаній для оптимізації кадрової політики, формування конкурентних пропозицій та стратегічного планування. Подальший розвиток системи може включати розширення кількості джерел даних (наприклад інші портали з працевлаштування), реалізацію більш складних моделей машинного навчання (наприклад, аналіз тональності описів вакансій) та створення повноцінного веб-інтерфейсу для кінцевих користувачів.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. Indianapolis: Wiley, 2013. 564 с.
2. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd ed. Morgan Kaufmann, 2012. 703 с.
3. Larose D. T., Larose C. D. Discovering Knowledge in Data: An Introduction to Data Mining. 2nd ed. Wiley, 2014. 336 с.
4. Loshin D. Business Intelligence: The Savvy Manager's Guide. 1st ed. Morgan Kaufmann, 2003. 270 с.
5. Документація Microsoft SQL Server [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/uk-ua/sql> (дата звернення: 28.10.2025).
6. Документація SQL Server Integration Services (SSIS) [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/sql/integration-services/sql-server-integration-services> (дата звернення: 28.10.2025).
7. Документація SQL Server Analysis Services (SSAS) [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/analysis-services/> (дата звернення: 28.10.2025).
8. Документація мови Python [Електронний ресурс]. – Режим доступу: <https://docs.python.org/3/> (дата звернення: 28.10.2025).
9. Документація Scikit-Learn: Machine Learning in Python [Електронний ресурс]. – Режим доступу: <https://scikit-learn.org/stable/> (дата звернення: 28.10.2025).
10. Документація Pandas: Python Data Analysis Library [Електронний ресурс]. – Режим доступу: <https://pandas.pydata.org/docs/> (дата звернення: 28.10.2025).
11. Документація Statsmodels: Statistical modeling and econometrics in Python [Електронний ресурс]. – Режим доступу: <https://www.statsmodels.org/stable/> (дата звернення: 28.10.2025).

12. Документація mlxtend: Machine Learning extensions [Електронний ресурс]. – Режим доступу: <https://rasbt.github.io/mlxtend/> (дата звернення: 28.10.2025).
13. Документація Matplotlib: Visualization with Python [Електронний ресурс]. – Режим доступу: <https://matplotlib.org/stable/users/index.html> (дата звернення: 28.10.2025).
14. Документація Seaborn: statistical data visualization [Електронний ресурс]. – Режим доступу: <https://seaborn.pydata.org/> (дата звернення: 28.10.2025).
15. Документація SQLAlchemy: The Python SQL Toolkit and Object Relational Mapper [Електронний ресурс]. – Режим доступу: <https://www.sqlalchemy.org/> (дата звернення: 28.10.2025).
16. Документація Power BI [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/power-bi/> (дата звернення: 28.10.2025).
17. Документація Node.js [Електронний ресурс]. – Режим доступу: <https://nodejs.org/docs/latest/api/> (дата звернення: 28.10.2025).
18. Документація TypeScript [Електронний ресурс]. – Режим доступу: <https://www.typescriptlang.org/docs/> (дата звернення: 28.10.2025).
19. Документація Axios [Електронний ресурс]. – Режим доступу: <https://axios-http.com/docs/intro> (дата звернення: 28.10.2025).
20. Документація Cheerio [Електронний ресурс]. – Режим доступу: <https://cheerio.js.org/> (дата звернення: 28.10.2025)
21. Using Machine Learning to Predict High School Student Employability – A Case Study [Електронний ресурс]. – Режим доступу: [https://www.researchgate.net/publication/338799931\\_Using\\_Machine\\_Learning\\_to\\_Predict\\_High\\_School\\_Student\\_Employability\\_-\\_A\\_Case\\_Study](https://www.researchgate.net/publication/338799931_Using_Machine_Learning_to_Predict_High_School_Student_Employability_-_A_Case_Study) (дата звернення: 10.11.2025).
22. Machine Learning and Deep Learning Techniques for Recommendation Systems: A Comprehensive Review [Електронний ресурс]. – Режим доступу:

[https://www.researchgate.net/publication/380604199\\_Machine\\_Learning\\_and\\_Deep\\_Learning\\_Techniques\\_for\\_Recommendation\\_Systems\\_A\\_Comprehensive\\_Review](https://www.researchgate.net/publication/380604199_Machine_Learning_and_Deep_Learning_Techniques_for_Recommendation_Systems_A_Comprehensive_Review) (дата звернення: 10.11.2025).

23. Artificial Intelligence Applications in the recruitment process opportunities and challenges [Електронний ресурс]. – Режим доступу:

[https://www.researchgate.net/publication/377570441\\_Artificial\\_Intelligence\\_Applications\\_in\\_the\\_recruitment\\_process\\_opportunities\\_and\\_challenges](https://www.researchgate.net/publication/377570441_Artificial_Intelligence_Applications_in_the_recruitment_process_opportunities_and_challenges)

(дата звернення: 10.11.2025).

24. Analysis of Data Mining Methods for Solving Problems of Labor Market Intellectual Analysis [Електронний ресурс]. – Режим

доступу: [https://www.researchgate.net/publication/364997799\\_ASSESSING\\_THE\\_REGIONAL\\_LABOR\\_MARKET\\_BY\\_USING\\_DATA\\_MINING\\_METHODS\\_WAYS\\_OF\\_EFFECTIVE\\_FUNCTIONING](https://www.researchgate.net/publication/364997799_ASSESSING_THE_REGIONAL_LABOR_MARKET_BY_USING_DATA_MINING_METHODS_WAYS_OF_EFFECTIVE_FUNCTIONING) (дата звернення: 10.11.2025).

25. Веб-портал працевлаштування Work.ua [Електронний ресурс]. – Режим доступу: <https://www.work.ua/> (дата звернення: 20.10.2025).

**ДОДАТОК А**

Код SQL Task 1 виконання заміни Staging таблиці унікальними даними

Сторінок 1

```

1 IF OBJECT_ID('dbo.Staging_Jobs_Unique', 'U') IS NOT NULL
2     DROP TABLE dbo.Staging_Jobs_Unique;
3
4 CREATE TABLE dbo.Staging_Jobs_Unique (
5     [Title] NVARCHAR(MAX),
6     [Employer] NVARCHAR(MAX),
7     [Employer Link] NVARCHAR(MAX),
8     [Location] NVARCHAR(MAX),
9     [Skills] NVARCHAR(MAX),
10    [Description] NVARCHAR(MAX),
11    [Source] NVARCHAR(MAX),
12    [Company Description] NVARCHAR(MAX),
13    [Categories] NVARCHAR(MAX),
14    [Posted Date] DATE,
15    [Max Salary] FLOAT,
16    [Min Salary] FLOAT
17 );
18 GO
19
20 WITH RankedJobs AS (
21     SELECT
22         *,
23         ROW_NUMBER() OVER(
24             PARTITION BY
25                 Title,
26                 Employer
27             ORDER BY
28                 [Posted Date] DESC
29         ) as RowNum
30     FROM
31         dbo.Staging_Jobs
32 )
33 INSERT INTO dbo.Staging_Jobs_Unique
34 SELECT
35     Title,
36     Employer,
37     [Employer Link],
38     Location,
39     Skills,
40     Description,
41     Source,
42     [Company Description],
43     Categories,
44     [Posted Date],
45     [Max Salary],
46     [Min Salary]
47 FROM
48     RankedJobs
49 WHERE
50     RowNum = 1;
51 GO
52
53 BEGIN TRANSACTION;
54     DROP TABLE dbo.Staging_Jobs;
55     EXEC sp_rename 'dbo.Staging_Jobs_Unique', 'Staging_Jobs';
56 COMMIT TRANSACTION;
57 GO

```