

## Магістр комп'ютерних наук

(Спеціалізація штучний інтелект і машинне навчання)

### Опис навчальної програми

#### Рівень 1

Назва дисципліни	ECTS / Години	Тривалість за розкладом, тижні
<a href="#">Програмування на Python: Основи та кращі практики</a>	5/125	7
<a href="#">Математика для комп'ютерних наук та вступ до методів розв'язування задач</a>	5/125	5
<a href="#">Базові алгоритми та структури даних</a>	5/125	5

## Рівень 2

Назва дисципліни	ECTS / Години	Тривалість за розкладом, тижні
<a href="#">Чисельне програмування на Python</a>	5 / 125	6
<a href="#">Реляційні бази даних: Концепції та технології</a>	5 / 125	5
<a href="#">Машинне навчання: Основи та застосування</a>	5 / 125	8
<a href="#">Візуальна аналітика</a>	5 / 125	7
<a href="#">Вступ до поглибленого навчання (Deep Learning)</a>	5 / 125	7
<a href="#">Генеративний та агентний ШІ</a>	5 / 125	7
<a href="#">Прикладне машинне навчання: бізнес-кейси</a>	5 / 125	5
<a href="#">Інженерія даних</a>	5 / 125	6
<a href="#">Продуктова аналітика та прикладна статистика</a>	5 / 125	7

## Рівень 3

Назва дисципліни	ECTS / Години	Тривалість за розкладом, тижні
<a href="#">Хмарні обчислення для ML/AI та попередньої обробки даних</a>	5 / 125	6
<a href="#">MLOps CI/CD</a>	5 / 125	7
<a href="#">Гнучке управління продуктами для команд розробників програмного забезпечення</a>	5 / 125	4
<a href="#">Кар'єрні стратегії та м'які навички для IT-спеціалістів</a>	5 / 125	2

## Опис дисципліни та очікувані результати навчання

### Рівень 1

#### Програмування на Python: Основи та кращі практики

*Опис:* Ця дисципліна дає практичне та детальне розуміння популярних парадигм програмування та типів зберігання даних. Студенти, які опанують цю дисципліну, зможуть писати та вирішувати проблеми програмування. Дисципліна починається з основ про функції, різні вбудовані функції та як їх кодувати, визначені користувачем. Потім студенти дізнаються про різні типи сховищ даних, дізнаються про списки та різні маніпуляції зі списками (як, наприклад, розбиття списку на частини), а також розглянуть приклади двовимірних списків.

Вивчаючи створення функцій, студенти повинні дізнатися, як можна зберігати різні результати та вхідні дані, використовуючи різні типи даних. Після вступу та обговорення списків, студенти розглянуть множини, кортежі, словники та рядки.

Студент повинен бути добре підготовлений до застосування цих концепцій та створення алгоритмів і програмного забезпечення, використовуючи те, що він вивчив на цьому курсі.

#### **Основні результати навчання:**

- Оцінювати, аналізувати та критикувати різні стратегії зберігання даних у комп'ютерній програмі.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах, що стосуються розв'язання задач з 2D-списками.
- Пропонувати відповідні рішення для складних і мінливих проблем зберігання даних, програмування функцій та алгоритмів.

<b>Пререквізити (умови для входу до навчання)</b>	- VSCode - Git&GitHub
<b>Інструменти, технології, які опановуються на дисципліні</b>	Python Git PEP8 Pickle
<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	- Основи програмування на Python та найкращі практики його використання. - Вміння розробляти програми на Python, використовуючи різноманітні структури даних та алгоритми. - Вміння писати чистий, модульний та ефективний код на Python, дотримуючись найкращих практик програмування.
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	Фінальний командний проєкт. Розробити персонального помічника з інтерфейсом командного рядка для управління контактами та нотатками, а також навчитись співпрацювати, обмінюватися ідеями, вирішувати конфлікти, планувати та координувати процес розробки.

## **Математика для комп'ютерних наук та вступ до методів розв'язування задач**

*Опис:* Математика та комп'ютерні науки тісно пов'язані між собою. Проблеми в комп'ютерних науках часто формалізуються і розв'язуються за допомогою математичних методів. Цілком ймовірно, що багато важливих проблем, які зараз стоять перед ІТ працівниками, будуть вирішені дослідниками, які мають досвід в алгебрі, аналізі, комбінаториці, логіці та/або теорії ймовірностей, а також в комп'ютерних науках.

Ця дисципліна охоплює дискретну математику для комп'ютерних наук

та інженерії. Теми можуть включати асимптотичний запис і зростання функцій; перестановки і комбінації; принципи підрахунку; дискретна ймовірність. Також можуть бути розглянуті інші вибрані теми, такі як рекурсивне означення та структурна індукція; машина Тюрінга та інваріанти; рекурентність; породжуючі функції.

Студенти зможуть пояснювати та застосовувати основні методи дискретної (неперервної) математики в інформатиці. Вони зможуть використовувати ці методи в наступних курсах з проєктування та аналізу алгоритмів, теорії обчислюваності, програмної інженерії та комп'ютерних систем. У центрі уваги дисципліни – реальні проблеми та застосунки, які часто зустрічаються в бізнесі та промисловості.

Крім того, студенти дізнаються про різні стратегії вирішення проблем і про те, коли їх використання дасть хороший старт. Вирішення проблем – це процес. Більшість стратегій передбачають кроки, які допомагають ідентифікувати проблему та обрати найкраще рішення.

Створення інструментарію стратегій вирішення проблем покращить навички вирішення цих проблем. Практикуюсь, студенти зможуть розпізнавати та обирати з-поміж кількох стратегій найбільш підходящу для вирішення складних проблем. Дисципліна буде зосереджений на розробці стратегій вирішення проблем, таких як абстрагування, модульність, рекурсія, ітерація, бісекція та вичерпне перерахування.

Дисципліна також ознайомить вас з масивами та деякими з їхніх реальних застосувань, таких як префіксна сума, перенос, підмасиви та двовимірні матриці. Приклади включатимуть галузеві проблеми та глибоке занурення в побудову їх рішень за допомогою різних підходів, визнаючи обмеження кожного з них (наприклад, коли слід використовувати структуру даних, а коли не слід).

### **Основні результати навчання:**

- Оцінювати, аналізувати та критикувати різні стратегії оцінки алгоритмічних витрат, що виникають у контексті розв'язання обчислювальних задач та вирішення питань, що виникають у контексті структурованих даних.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах, що стосуються того, як менеджери повинні оцінювати ефективність алгоритмів та вирішувати проблеми зі структурованими даними.
- Пропонувати відповідні рішення для складних та мінливих проблем, пов'язаних з вирішенням проблем у розробці програмного забезпечення.

<p><b>Пререквізити (умови для входу до навчання)</b></p>	<p>- Знання основ Python, включаючи роботу зі змінними, списками, словниками та функціями.</p> <p>Nice-to-have</p> <ul style="list-style-type: none"> <li>- Google Colab</li> </ul> <p>Базові знання бібліотек:</p> <ul style="list-style-type: none"> <li>- NumPy</li> <li>- Pandas</li> <li>- Matplotlib</li> <li>- SciPy</li> </ul>
<p><b>Інструменти, технології, які опановуються на дисципліні</b></p>	<p>Математичний апарат та його програмна реалізація за допомогою бібліотек Python.</p>
<p><b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b></p>	<ul style="list-style-type: none"> <li>- Теоретичні і професійно-практичні компетенції в області математичних методів розв'язання інженерних задач.</li> <li>- Оволодіння відповідним математичним апаратом, який повинен бути достатнім для опрацювання математичних моделей, пов'язаних з подальшою практичною діяльністю фахівців.</li> <li>- Оволодіння основними математичними методами, необхідними для аналізу і моделювання процесів і явищ, пошуків оптимальних рішень з метою підвищення ефективності програмування і вибору найкращих способів реалізації цих рішень, опрацювання і аналізу результатів експериментів.</li> </ul>
<p><b>Фінальний проєкт, який виконує студент та опис проєкту</b></p>	<p>Фіналізуюче домашнє завдання. Маючи вхідний набір даних, опрацювати його, вирішити задачу прогнозування ціни нерухомості (з потоку MCS07, раніше — прогнозування цільової змінної — якості сну) у такому порядку:</p> <ol style="list-style-type: none"> <li>1. Аналіз та підготовка даних (статистичний аналіз).</li> <li>2. Моделювання (створення регресійних моделей для прогнозування).</li> <li>3. Аналіз результатів (візуалізація та порівняння якості моделей).</li> </ol> <p>За бажанням студент може оформити проєкт на платформі Kaggle у вигляді публічного ноутбука.</p>

## **Базові алгоритми та структури даних**

*Опис:* Ця дисципліна спрямована на формування стійких фундаментальних знань про структури даних (DS), які широко використовуються в обчисленнях. Модуль починається з представлень часової та просторової складності та оцінки фрагментів коду. Це допомагає студентам знаходити компроміси між різними структурами даних під час розв'язання реальних обчислювальних задач. Модуль знайомить з найбільш поширеними базовими структурами даних, такими як динамічні масиви,

багатовимірні масиви, списки, рядки, хеш-таблиці, бінарні дерева, збалансовані бінарні дерева, черги пріоритетів та графіки. У модулі обговорюються різні варіанти реалізації для кожної з перерахованих вище структур даних, а також компроміси в просторі і часі для кожної реалізації. На цьому курсі студенти реалізують ці структури даних з нуля, щоб отримати глибоке розуміння їхньої внутрішньої роботи. Студенти також дізнаються, як використовувати вбудовані структури даних, доступні в різних мовах програмування/бібліотеках, таких як Python/NumPy/JavaScript. Студенти розв'язують реальні задачі, в яких вони повинні використовувати оптимальну структуру даних для розв'язання обчислювальної задачі.

### Основні результати навчання:

- Оцінювати, аналізувати та критикувати різні стратегії вирішення питань, що виникають у контексті структур даних.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах щодо того, як менеджери повинні впроваджувати структури даних.
- Пропонувати відповідні рішення для складних і мінливих проблем, пов'язаних з різними підходами до застосування структур даних.

<b>Пререквізити (умови для входу до навчання)</b>	<ul style="list-style-type: none"> <li>- Знання основ Python, включаючи роботу зі змінними, списками, словниками та функціями.</li> <li>- Основи математики та статистики.</li> </ul>
<b>Інструменти, технології, які опановуються на дисципліні</b>	Базові алгоритми та структури даних, жадібні та рандомізовані алгоритми, динамічне та лінійне програмування, програмна реалізація алгоритмів та структур даних за допомогою бібліотек Python.
<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	<ul style="list-style-type: none"> <li>- Широкий спектр сучасних алгоритмів і структур для ефективного збереження та швидкої обробки великих статистичних даних.</li> <li>- Способи, області та прийоми опрацювання та використання структур даних.</li> <li>- Сучасні методи побудови та аналізу ефективних алгоритмів та способи їх реалізації в конкретних застосуваннях.</li> </ul> <p>Вміти:</p> <ul style="list-style-type: none"> <li>- Критично підходити до вибору структур даних та алгоритмів при реалізації конкретних інформаційних проєктів (аналізувати, порівнювати та обирати).</li> <li>- Застосовувати методи розробки алгоритмів конструювання програмного забезпечення та структур даних і знань.</li> </ul>

	<ul style="list-style-type: none"> <li>- Будувати структурований алгоритм обробки базових структур даних, переконавшись в його працездатності та отриманні потрібного результату.</li> <li>- Оцінювати складові ефективності алгоритмів функціонування комп'ютеризованих систем.</li> </ul>
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	<p>Фінальний індивідуальний проєкт. 7 незалежних завдань:</p> <ol style="list-style-type: none"> <li>1. Реалізувати алгоритми обробки однозв'язного списку.</li> <li>2. Візуалізувати фрактал «дерево Піфагора» за допомогою рекурсії.</li> <li>3. Реалізувати алгоритм Дейкстри.</li> <li>4. Реалізувати візуалізацію бінарної купи.</li> <li>5. Виконати програмну реалізацію методів DFS та BFS для візуалізації обходу дерева.</li> <li>6. Розробити функції жадібного алгоритму та алгоритму динамічного програмування для оптимізації бюджету.</li> <li>7. Використати метод Монте-Карло для моделювання кидання двох ігрових кубиків.</li> </ol>

## Рівень 2

### Чисельне програмування на Python

*Опис:* Ця дисципліна допомагає студентам перевести математичні/статистичні/наукові концепції в код. Це базова дисципліна для написання коду для вирішення проблем Data Science ML та AI. Він вводить базові концепції програмування (такі як керуючі структури, рекурсія, класи та об'єкти) з нуля, не припускаючи ніяких попередніх умов, щоб зробити цю дисципліну доступною для студентів з некомп'ютерних наукових галузей, таких як біологія, фізика, медицина, хімія, будівництво та машинобудування тощо. Після побудови міцного фундаменту, дисципліна переходить до занурення в основні математичні бібліотеки, такі як NumPy, Scipy та Pandas. Студенти також дізнаються, коли і як використовувати вбудовані структури даних, такі як списки, словники, набори та кортежі. Модуль вводить поняття обчислювальної складності, щоб допомогти студентам писати оптимізований код, використовуючи відповідні структури даних та методи алгоритмічного проєктування. У цьому модулі не розглядаються структури даних та методи проєктування алгоритмів – вони доступні в модулі «Структури даних та алгоритми». Ця дисципліна є цінною для всіх студентів, які спеціалізуються в математичних підгалузях CS, таких як ML, Data Science, наукові обчислення тощо.

## Основні результати навчання:

- Оцінювати, аналізувати та критикувати різні стратегії вирішення питань, що виникають в контексті чисельного програмування на Python.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах, що стосуються того, як менеджери повинні працювати з числовим програмуванням на Python.
- Пропонувати відповідні рішення для складних та мінливих проблем, що стосуються чисельного програмування на Python.

<b>Пререквізити (умови для входу до навчання)</b>	<ul style="list-style-type: none"><li>- Знання мови програмування Python на базовому рівні.</li><li>- Основи роботи з бібліотеками для наукового обчислення в Python, такими як NumPy та SciPy.</li><li>- Розуміння основних концепцій чисельних методів та алгоритмів.</li></ul>
<b>Інструменти, технології, які опановуються на дисципліні</b>	NumPy Pandas Matplotlib SciPy
<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	<ul style="list-style-type: none"><li>- Знання та вміння перекладати математичні, статистичні та наукові концепції у код з використанням мови програмування Python.</li><li>- Знання основних концепцій програмування, включно зі структурами управління, рекурсією, класами та об'єктами, що дозволить вам ефективно писати код для завдань із машинним навчанням та штучного інтелекту.</li><li>- Вміння використовувати основні математичні бібліотеки, зокрема NumPy, SciPy і Pandas, а також вбудовані структури даних: списки, словники, множини та кортежі.</li><li>- Знання про концепції обчислювальної складності та вміння писати оптимізований код із використанням відповідних структур даних та методів алгоритмічного проектування.</li></ul>
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	Фіналізуюче домашнє завдання. Завантаження та візуалізація даних. Кластеризація даних для виявлення груп. Зменшення розмірності даних за допомогою методу PCA. Побудова та оцінка моделі класифікації методом логістичної регресії. Оцінка якості моделей класифікації та узагальнення результатів.

## **Реляційні бази даних: Концепції та технології**

*Опис:* Це основна та фундаментальна дисципліна, яка має на меті навчити студента моделювати, проєктувати, впроваджувати та запитувати системи реляційних баз даних для реальних потреб зберігання та обробки даних. Студенти почнуть з діаграмних інструментів (ER-діаграми) для відображення реальної проблеми зберігання даних у вигляді сутностей, зв'язків та ключів. Потім вони навчаться переводити ER-діаграму в реляційну модель з таблицями. Після цього учасники знайомляться з мовою SQL як фактичним інструментом для створення, модифікації, додавання, видалення, запитів та маніпулювання даними в реляційній базі даних. Зважаючи на популярність SQL, дисципліна приділяє значну увагу формуванню вміння писати оптимізовані та складні запити для різних завдань маніпулювання даними.

Модуль знайомить студентів з різними реальними прикладами SQL, щоб сформувані сталі практичні знання. Потім студенти переходять до розуміння різних компромісів у сучасних реляційних базах даних, наприклад, між обсягом пам'яті та затримкою. Проєктування бази даних потребує глибокого розуміння нормальних форм для мінімізації дублювання даних, індексування для прискорення та вирівнювання таблиць, щоб уникнути складних об'єднань в середовищах з низькою затримкою. Ці реальні стратегії проєктування баз даних обговорюються на практичних прикладах з різних галузей. Більша частина дисципліни використовує базу даних MySQL з відкритим вихідним кодом та хмарні реляційні бази даних (наприклад, Amazon RDS), щоб допомогти студентам застосувати вивчені концепції на реальних базах даних за допомогою завдань.

### **Основні результати навчання:**

- Оцінювати, аналізувати та критикувати різні стратегії вирішення питань, що виникають в контексті реляційних баз даних.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах щодо того, як менеджери повинні працювати з реляційними базами даних.
- Пропонувати відповідні рішення для складних і мінливих проблем, що стосуються реляційних баз даних.

<b>Пререквізити (умови для входу до навчання)</b>	- Базові знання математики
<b>Інструменти, технології, які опановуються на дисципліні</b>	MySQL
<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	<ul style="list-style-type: none"> <li>- Знання основних концепцій та технік реляційних баз даних.</li> <li>- Вміння проєктувати, створювати та керувати реляційними базами даних.</li> <li>- Вміння виконувати складні запити та оптимізувати їх для ефективної обробки даних.</li> </ul>
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	Фінальний індивідуальний проєкт. Завантажити дані, нормалізувати таблицю, проаналізувати дані, побудувати колонку з використанням вбудованих SQL функцій, побудувати та використати власну функцію

## Машинне навчання: Основи та застосування

*Опис:* Ця дисципліна зосереджується на побудові базових класифікаційних та регресійних моделей та їх чіткому розумінні як з математичної, так і з прикладної точки зору. Модуль починається з базового вступу до багатовимірної геометрії точок, метрики відстаней, гіперплощин і гіперсфер. На основі цього ми вводимо математичне формулювання логістичної регресії для знаходження розділяючої гіперплощини. Студенти навчаться розв'язувати задачу оптимізації за допомогою векторного числення та алгоритмів на основі градієнтного спуску (ГС). Модуль знайомить з обчислювальними варіаціями градієнтного спуску, такими як міні-пакетний та стохастичний градієнтний спуск. Студенти також вивчать інші популярні методи класифікації та регресії, такі як k-найближчих сусідів, наївний Байєс, дерева рішень, лінійна регресія тощо. Студенти також дізнаються, як кожен з цих методів застосовується в різних реальних ситуаціях, таких як наявність викидів, незбалансованих даних, багатокласова класифікація тощо. Студенти вивчать компроміс між зміщенням і дисперсією, а також різні методи уникнення надмірного та недостатнього припасування. Студенти також вивчають ці алгоритми з точки зору Байєса та геометричної інтуїції. Цей модуль є практичним, і студенти застосують всі ці класичні методи до реальних кейсів.

## Основні результати навчання:

- Оцінювати, аналізувати та критикувати різні стратегії вирішення питань, що виникають у контексті машинного навчання.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах, що стосуються того, як менеджери повинні працювати з машинним навчанням.
- Пропонувати відповідні рішення для складних і мінливих проблем, пов'язаних з машинним навчанням.

<b>Пререквізити (умови для входу до навчання)</b>	<ul style="list-style-type: none"><li>- Впевнений рівень володіння Python, знання структур даних, досвід роботи з бібліотеками Pandas, NumPy, Matplotlib/Seaborn.</li><li>- Основи теорії ймовірностей та статистики, лінійна алгебра, методи оптимізації.</li><li>- Основи роботи з Git.</li></ul> <p>Nice-to-have:</p> <ul style="list-style-type: none"><li>- Практичний досвід підготовки аналітичних звітів та візуалізації даних.</li><li>- Знання базових концепцій візуалізації даних (графіки, діаграми, характеристики візуалізацій).</li></ul>
<b>Інструменти, технології, які опановуються на дисципліні</b>	Scikit-Learn statsmodels SciPy statsmodels Conda Spyder Kaggle
<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	<ul style="list-style-type: none"><li>- Теоретичні основи ML</li><li>- Підготовка даних / EDA</li><li>- Прийоми feature engineering</li><li>- Побудова й оцінка моделей</li><li>- Прийоми візуалізації даних</li></ul>
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	Фінальний індивідуальний проєкт. Індивідуальний проєкт. Змагання на платформі Kaggle: побудова моделі для прогнозування на основі реального датасету. Реалізувати повний цикл вирішення задачі машинного навчання, що включає в себе такі етапи: <ul style="list-style-type: none"><li>- збір і обробку даних,</li><li>- навчання моделі,</li><li>- пакування моделі,</li><li>- публікацію моделі.</li></ul>

## Візуальна аналітика

*Опис:* Ця дисципліна спрямована на формування міцних фундаментальних знань про інструменти аналізу даних, які широко використовуються в галузі Data Science. Зараз існують потужні інструменти візуалізації даних, які використовуються в галузі бізнес-аналітики для обробки та візуалізації необроблених бізнес-даних у дуже презентабельному та зрозумілому форматі. Хорошим прикладом є Tableau, який використовується усіма відділами аналізу даних компаній та компаніями з аналізу даних у різних галузях завдяки простоті використання та ефективності. Tableau використовує реляційні бази даних, Online Analytical Processing Cubes, електронні таблиці, хмарні бази даних для створення візуалізацій графічного типу. Дисципліна починається з візуалізацій і переходить до поглибленого вивчення різних функцій діаграм і графіків, обчислень, картографування та інших функціональних можливостей. Слухачів навчають швидким табличним розрахункам, опорним лініям, різним типам візуалізацій, діапазонам і розподілам, параметрам, діаграмам руху, трендам і прогнозуванню, форматуванню, історіям, запису показників і розширеному картографуванню.

### Основні результати навчання:

- Оцінювати, аналізувати та критикувати різні стратегії розповіді історій за допомогою даних.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах, що стосуються компромісів у науці про дані.
- Пропонувати відповідні рішення для складних і мінливих проблем, що стосуються науки про дані в реальних інженерних і наукових контекстах.

<b>Пререквізити (умови для входу до навчання)</b>	<ul style="list-style-type: none"><li>- Основи аналітики даних.</li><li>- Знання базових концепцій візуалізації даних (графіки, діаграми, характеристики візуалізацій).</li></ul> <p>Nice-to-have:</p> <ul style="list-style-type: none"><li>- Знання основ дизайну та візуальної комунікації.</li><li>- Досвід роботи з графічним редактором для створення власних графічних елементів.</li></ul>
<b>Інструменти, технології, які опановуються на дисципліні</b>	PowerBI AI-інструменти для візуалізацій

<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	<p>Володіння навичками використання потужних інструментів візуалізації даних, наприклад PowerBI, для обробки та представлення сирових даних бізнесу у зрозумілому та доступному форматі.</p> <p>Глибоке розуміння різних функцій, розрахунків, технік мапінгу та інших можливостей у межах інструментів візуалізації даних.</p> <p>Знання передових функцій і техніки візуалізації даних, включно зі швидкими розрахунками таблиць, використанням параметрів, рухомими графіками, тенденціями та прогнозуванням, форматуванням тощо.</p>
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	<p>Фінальний індивідуальний проєкт. Побудувати інтерактивний дашборд з даними для прийняття управлінських рішень для топ-менеджерів компанії з використанням застосунку PowerBI.</p>

## **Вступ до поглибленого навчання (Deep Learning)**

*Опис:* Цей курс надає ґрунтовний математичний та прикладний вступ до глибокого навчання. Модуль починається з моделі перцептрона як надто спрощеного наближення до біологічного нейрона. Ми мотивуємо потребу в мережі нейронів і те, як їх можна з'єднати, щоб сформувати багат шаровий перцептрон (MLPs). Після цього студенти отримують чітке розуміння алгоритмів зворотного поширення та його обмежень, починаючи з 1980-х років. Студенти вивчають, як сучасне глибоке навчання почало розвиватися завдяки вдосконаленим обчислювальним інструментам та наборам даних. Ми вивчаємо більш сучасні блоки активації (такі як ReLU і SeLU) і те, як вони долають проблеми з більш класичними блоками сигмоїда і танха. Студенти вивчають методи вагової ініціалізації, регуляризацію відсівом, пакетну нормалізацію тощо, щоб забезпечити успішне навчання глибоких МНК. Модуль вивчає варіанти градієнтного спуску, які були спеціально розроблені для роботи з системами глибокого навчання, такими як ADAM, AdaGrad, RMSProp тощо. Студенти також вивчають AutoEncoders, VAEs та Word2Vec як некеровані архітектури кодування для глибокого навчання. Ми застосовуємо всю вивчену теорію до різних реальних проблем, використовуючи TensorFlow 2 та Keras. Студенти також розуміють, як TensorFlow 2 працює зсередини, приділяючи особливу увагу обчислювальній обробці графів.

### **Основні результати навчання:**

- Створювати синтетичні контекстуальні дискусії з ключових питань, пов'язаних з глибоким навчанням
- Застосовувати професійний та науковий підхід до дослідження проблем, пов'язаних з глибоким навчанням

- Ефективно управляти міждисциплінарними питаннями, що виникають у зв'язку з глибоким навчанням
- Демонструвати самостійність у дослідженнях та оригінальність рішень, розроблених для глибокого навчання
- Діяти автономно у визначенні дослідницьких проблем та рішень, пов'язаних з глибоким навчанням
- Вирішувати проблеми та бути готовим до прийняття управлінських рішень, пов'язаних з методами та принципами глибокого навчання

## **Генеративний та агентний ШІ**

*Опис:* Цей курс зосереджується на моделюванні послідовностей (текст, музика, часові ряди) за допомогою моделей глибокого навчання. Ми починаємо з простої рекурентної нейронної мережі та її обмежень при роботі з довгими послідовностями. Студенти вивчають LSTM та GRU, які можуть обробляти значно довші послідовності для моделювання послідовних даних, таких як текст, музика, генні послідовності та часові ряди. Ми вивчаємо варіації LSTM, такі як двонаправлені LSTM та архітектури кодер-декодер. Після цього ми детально вивчаємо механізм уваги та моделі на основі трансформаторів, які на сьогоднішній день є найсучаснішими для НЛП та моделювання послідовностей. Модуль вивчає трансформатори кодер-декодер, BERT, BERT-варіації, моделі GPT-1,2 і 3 як з архітектурної та математичної точок зору, так і з практичної точки зору. Студенти навчатимуться реалізовувати багато з цих складних моделей з нуля (використовуючи TensorFlow 2 і Keras), щоб отримати більш глибоке розуміння того, як вони працюють зсередини. Студенти вивчатимуть популярні застосування глибокого навчання в NLP, такі як тегування частин мови, системи запитань-відповідей, розмовні системи (чат-боти), семантичний пошук з низькою затримкою тощо. Для кожної з цих задач студенти вивчатимуть найсучасніші моделі глибокого навчання разом із програмними реалізаціями.

Ця дисципліна охоплює сучасні підходи до глибокого навчання, зокрема роботу з генеративним штучним інтелектом (GenAI) та великими мовними моделями (LLM). Студенти навчатимуться створювати пайплайни на основі LLM та векторних баз даних, розробляти інтелектуальних агентів і чат-ботів, а також працювати з генеративними моделями для створення зображень (Stable Diffusion), музики (Suno AI) та впровадження семантичного пошуку. Курс поєднує

теоретичні знання з практичними навичками для роботи з передовими технологіями AI.

**Основні результати навчання:**

- Створювати синтетичні контекстуальні дискусії з ключових питань, пов'язаних з глибоким навчанням для NLP (Natural Language Processing, обробки природньої мови)
- Застосовувати професійний та науковий підхід до дослідницьких проблем, пов'язаних з глибоким навчанням для NLP
- Ефективно управляти міждисциплінарними питаннями, які виникають у зв'язку з глибоким навчанням для NLP
- Демонструвати самостійність у дослідженнях та оригінальність рішень, розроблених для глибокого навчання для NLP
- Діяти автономно у визначенні дослідницьких проблем та рішень, пов'язаних з глибоким навчанням для NLP
- Вирішувати проблеми та бути готовим до прийняття управлінських рішень, пов'язаних з методами та принципами глибокого навчання для NLP

## Прикладне машинне навчання: бізнес-кейси

*Опис:* Цей курс знайомить з більш просунутими методами ML, такими як ансамблі: bagging, boosting, каскадні та стекові класифікатори та регресори. Він охоплює як теоретичні основи, так і прикладні деталі цих методів, а також популярні реалізації бустінгу, такі як LightGBM, CatBoost та XGBoost. Студенти також заглиблюються в ядерні методи з особливим акцентом на SVM для класифікації та регресії. Студенти вивчатимуть сучасні методи агностичної важливості ознак моделі та інтерпретованості моделі, такі як LIME та SHAP. Студенти також вивчають класичні методи кодування тексту на основі NLP, такі як Bag-of-words, TF-IDF тощо. Модуль вивчає різні класичні методи аналізу та прогнозування часових рядів, такі як ARMA, ARIMA тощо. Студенти також вчать ставити задачі прогнозування часових рядів як задачі регресії та класифікації для використання добре вивчених методів ML. Після цього студенти вивчають різні методи Feature engineering, які часто стають у нагоді при вирішенні реальних проблем. Студенти вивчатимуть такі методи, як аналіз помилок, абстрактний аналіз тощо, щоб налагодити і зрозуміти, чому і де модель працює добре, а де погано. Розгляд feature engineering передбачено також у розрізі детальних кейсів з практики (з якими даними стикаємось на практиці, у яких форматах, де вони беруться, як обираємо алгоритм) й розкриває продуктове застосування власне написаних алгоритмів машинного навчання.

Одна тема присвячена UML діаграмам по різних практичних ML/AI проектах. Пізніше в цьому курсі ми розглянемо, як створювати системи рекомендацій, використовуючи методи фільтрації на основі контенту та спільної роботи.

### Основні результати навчання:

- Створювати синтетичні контекстуальні дискусії з ключових питань, пов'язаних з передовим машинним навчанням
- Застосовувати професійний та науковий підхід до дослідницьких проблем, пов'язаних з передовим машинним навчанням
- Ефективно управляти міждисциплінарними питаннями, які виникають у зв'язку з передовим машинним навчанням
- Демонструвати самостійність у дослідженнях та оригінальність у рішеннях, розроблених для вдосконаленого машинного навчання

- Діяти автономно у визначенні дослідницьких проблем та рішень, пов'язаних з Machine Learning
- Вирішувати проблеми та бути готовим до прийняття управлінських рішень, пов'язаних з методами та принципами поглибленого машинного навчання

## Інженерія даних

*Опис:* Дані – це паливо, що рухає всі великі організації. Ця дисципліна допоможе вам зрозуміти, як обробляти дані в масштабі.

Від розуміння основ розподіленої обробки до проектування сховищ даних і написання конвеєрів ETL (Extract Transform Load) для обробки пакетних і потокових даних.

Студенти отримують комплексне уявлення про повний життєвий цикл інженерії даних.

### Основні результати навчання:

- Оцінювати, аналізувати та критикувати різні стратегії управління великими масивами даних у режимі реального часу.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах, що стосуються життєвого циклу інженерії даних.
- Пропонувати відповідні рішення для складних і мінливих проблем обробки великих даних в реальних інженерних контекстах.

<p><b>Пререквізити (умови для входу до навчання)</b></p>	<ul style="list-style-type: none"> <li>- Знання та розуміння основ програмування, зокрема мови Python.</li> <li>- Знання базових алгоритмів і структур даних.</li> <li>- Основи роботи з базами даних і мовою SQL.</li> <li>- Знання Linux (основи bash/shell).</li> <li>- Знання Docker.</li> <li>- Базові знання математики.</li> </ul> <p>Nice-to-have:</p> <ul style="list-style-type: none"> <li>- Загальне уявлення про сховища даних (data warehouses) і Data Lakes.</li> <li>- Pandas.</li> </ul>
<p><b>Інструменти, технології, які опановуються на дисципліні</b></p>	<ul style="list-style-type: none"> <li>- Apache (py)spark (зокрема Spark Streaming),</li> <li>- Apache Kafka,</li> <li>- Apache Airflow,</li> <li>- Slowly Changing Dimensions,</li> <li>- Change Data Capture,</li> </ul>

	<ul style="list-style-type: none"> <li>- databricks,</li> <li>- Apache Flink,</li> <li>- DBT,</li> <li>- Apache Beam,</li> <li>- Cloud Native ETL Tools,</li> <li>- Snowflake.</li> </ul>
<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	<ul style="list-style-type: none"> <li>- Базові знання необхідні для розуміння інженерної сторони справи, та практичні навички з Data Engineering, які можна буде застосувати в реальних проєктах.</li> <li>- Знання передових технологій та інструментів, які є в арсеналі сучасного дата-інженера, і вміння впроваджувати їх для підвищення ефективності ваших аналітичних досліджень.</li> </ul> <p>Вміння працювати з такими інструментами:</p> <ul style="list-style-type: none"> <li>- Apache (py)spark (зокрема Spark Streaming),</li> <li>- Apache Kafka,</li> <li>- Apache Airflow.</li> </ul> <p>Розуміння важливих концептів роботи з даними:</p> <ul style="list-style-type: none"> <li>- Slowly Changing Dimensions,</li> <li>- Change Data Capture.</li> </ul> <p>Уявлення про:</p> <ul style="list-style-type: none"> <li>- databricks,</li> <li>- Apache Flink,</li> <li>- DBT,</li> <li>- Apache Beam,</li> <li>- Cloud Native ETL Tools,</li> <li>- Snowflake.</li> </ul>
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	<p>Фінальний індивідуальний проєкт.</p> <ol style="list-style-type: none"> <li>1. Вирішити реальну задачу обробки потоків даних у режимі реального часу і реалізувати стрімінгове рішення, яке зможе оновлювати аналітику та моделі на основі поточних даних.</li> <li>2. Побудувати трирівневу архітектуру обробки даних: від початкового збереження (landing zone), через оброблені та очищені дані (bronze/silver), до фінального аналітичного набору (gold). Автоматизувати цей процес за допомогою Apache Airflow.)</li> </ol>

## **Продуктова аналітика та прикладна статистика**

*Опис:* Ця дисципліна спочатку вчить студентів аналізувати способи взаємодії користувачів з сервісом. Такий метод, який називається

продуктовою аналітикою, допомагає бізнесу відстежувати та аналізувати дані користувачів. Слухачі дізнаються більше про те, що потрібно для просування продукту від ідеї до реалізації, запуску, а потім до ітеративних вдосконалень. Дисципліна вчить, як вимірювати прогрес, підтверджувати або оновлювати гіпотези продукту та презентувати отримані знання про продукт. Крім того, студенти отримають досвід прийняття обґрунтованих рішень, а також дізнаються, як презентувати отримані результати та створювати бізнес-кейс на основі аналітики, щоб отримати підтримку для продукту.

Друга частина дисципліни охоплює основи теорії ймовірностей, статистичні методи та обчислювальні алгоритми для математично строгого аналізу даних. Студенти дізнаються про популярні дискретні та неперервні розподіли, такі міри, як середнє значення та дисперсія, а також методи визначення кореляцій у даних. Модуль також охоплює основи непараметричної статистики та її застосування до реальних проблем в охороні здоров'я. Студенти отримають прикладне та математичне розуміння статистики, що лежить в основі експериментів контроль-лікування (A/B) та перевірки гіпотез, використовуючи обчислювальні інструменти, такі як Bootstrapping, методи Монте-Карло та RANSAC.

### Основні результати навчання:

- Аналізувати та оцінювати різні стратегії та методології, рекомендовані в наукових джерелах для підвищення успіху продукту та вирішення питань в контексті прикладної статистики.
- Пропонувати відповідні рішення для складних і мінливих проблем, пов'язаних з успіхом або невдачею продукту та прикладною статистикою в реальних умовах.
- Критикувати та оцінювати сильні та слабкі сторони різних методологій і стратегій вимірювання залученості користувачів та застосування теорії ймовірностей в управлінському контексті.

<p><b>Пререквізити (умови для входу до навчання)</b></p>	<ul style="list-style-type: none"> <li>- Основи аналітики даних.</li> <li>- Знання базових статистичних понять (середнє значення, медіана, стандартне відхилення), елементарне розуміння теорії ймовірностей.</li> <li>- Вміння працювати з аналітичними інструментами.</li> </ul> <p>Nice-to-have:</p> <ul style="list-style-type: none"> <li>- Досвід роботи з візуалізацією даних і побудовою графіків.</li> </ul>
<p><b>Інструменти, технології, які опановуються</b></p>	<p>Optimizely</p>

<p><b>на дисципліні</b></p>	<p>Amplitude Jira Google Analytics Google Tag Manager</p>
<p><b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b></p>	<ul style="list-style-type: none"> <li>- Володіння навичками продуктового аналітика, включно з аналізом даних про залучення користувачів та їх використання для відстеження, аналізу та покращення продуктів протягом їхнього життєвого циклу.</li> <li>- Здатність вимірювати прогрес, перевіряти або оновлювати гіпотези щодо продукту та представляти отримані результати навчання.</li> <li>- Забезпечення прийняття обґрунтованих рішень та навички презентації результатів, а також розроблення аналітикою підтриманого бізнес-кейсу для залучення підтримки продукту.</li> <li>- Розуміння основ теорії ймовірності, статистичних методів та обчислювальних алгоритмів для математично обґрунтованого аналізу даних.</li> <li>- Знання прикладних і математичних аспектів статистики, що лежать в основі A/B тестування та перевірки гіпотез, із використанням інструментів обчислювальної статистики, як-от Bootstrapping, методи Монте-Карло та RANSAC.</li> </ul>
<p><b>Фінальний проєкт, який виконує студент та опис проєкту</b></p>	<p>Фінальний індивідуальний проєкт. Перевірка гіпотез, ANOVA-аналіз, аналіз результатів A/B тестування, презентація отриманих результатів.</p>

## Рівень 3

### Хмарні обчислення для ML/AI та попередньої обробки даних

*Опис:* Це курс, який фокусується як на архітектурному дизайні, так і на практичному вивченні передових хмарних сервісів. Ми почнемо з моделі безсерверних обчислень і того, як вона досягається більшістю хмарних провайдерів. Ми навчимося використовувати її для створення веб-додатків, додатків для обробки даних і файлів та аналітики. Потім ми дізнаємося про архітектуру розподілених черг повідомлень і про те, як їх можна використовувати для організації складних хмарних систем з великою кількістю компонентів і сервісів. Моніторинг ресурсів у вашій хмарній системі - це ключ до забезпечення низьких витрат і високої доступності, а також безперебійного функціонування всієї системи в цілому. Ми навчимося використовувати AWS CloudWatch для відстеження різних ключових показників, запуску тривоги, виявлення аномальної поведінки та реагування на них майже в режимі реального часу. Ми вивчимо архітектуру та дизайн балансувальників навантаження і те, як вони відіграють ключову роль у більшості горизонтально масштабованих веб-додатків. Студенти також дізнаються про архітектуру та дизайн мереж доставки контенту (CDN) від Akamai та Amazon. Ми дізнаємося, як CDN можна використовувати для швидкої доставки прямих трансляцій та контенту веб-сайтів, використовуючи глобально розподілені сервери та кешування. Більша частина курсу передбачає вивчення внутрішньої архітектури різних хмарних систем та використання їх для вирішення реальних інженерних завдань.

Вона занурюватиме у хмарні платформи для саме AI-проектів, тренування власних моделей на хмарі (збереження, процесинг даних, засоби побудови Machine Learning), ознайомлення як з AWS так і з Google Cloud й Azure. Піднімає питання розрахунку витрат і порівняння on-premises та Cloud інфраструктури.

## MLOps CI/CD

*Опис:* Ця дисципліна спрямована на формування основної компетенції побудови реальних комплексних систем машинного навчання та їх розгортання у виробництві для вирішення різноманітних проблем і сценаріїв. Студенти вивчатимуть різноманітні системи ML, починаючи від високопродуктивних систем з низькою затримкою в Інтернеті до пристроїв Інтернету речей з низькою обчислювальною потужністю та обмеженим енергоспоживанням (як, наприклад, розумні годинники). Студенти детально вивчатимуть життєвий цикл ML та різні компоненти. Ми також використовуємо реальні платформи ML, такі як KubeFlow від Google, TensorFlow Lite та SageMaker від Amazon, щоб реалізувати реальні системи та зрозуміти інженерні компроміси та виклики. Студенти також вивчатимуть відповідні технології та інструменти, такі як контейнеризація (Docker) та оркестрування контейнерів (Kubernetes), а також Git, які часто широко використовуються в реальних масштабованих системах ML. На цьому практичному курсі ми вирішуватимемо численні реальні кейси та обговорюватимемо рішення, створені різними компаніями та організаціями, щоб надати студентам всебічне розуміння різноманітних систем та варіантів дизайну.

### **Основні результати навчання:**

- Оцінювати, аналізувати та критикувати різні стратегії вирішення питань, що виникають у контексті постановки машинного навчання на виробництво.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах, що стосуються того, як менеджери повинні працювати з постановкою машинного навчання.
- Пропонувати відповідні рішення для складних і мінливих проблем, пов'язаних з виробничою реалізацією машинного навчання.

<b>Пререквізити (умови для входу до навчання)</b>	<ul style="list-style-type: none"> <li>- Знання та розуміння основ машинного навчання.</li> <li>- Основи роботи з системами контролю версій, такими як Git.</li> </ul> <p>Nice-to-have:</p> <ul style="list-style-type: none"> <li>- Розуміння концепцій контейнеризації та оркестрації (наприклад, Docker, Kubernetes).</li> </ul>
<b>Інструменти, технології, які опановуються на дисципліні</b>	Kubeflow Jenkins GitLab CI/CD

<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	<ul style="list-style-type: none"> <li>- Володіння навичками створення повних ML-систем та їх розгортання, з урахуванням різних сценаріїв та обмежень.</li> <li>- Глибоке розуміння життєвого циклу ML, включно з етапами дослідження, розробки, навчання, оцінки та розгортання, з фокусом на інженерні компроміси та виклики.</li> <li>- Знання ML-платформ і технологій та вміння реалізувати системи ML реального світу з їх використанням.</li> <li>- Ознайомленість із технологіями та інструментами Docker, Kubernetes та Git, які часто використовуються в масштабованих системах ML.</li> </ul>
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	Фінальний індивідуальний проєкт. Побудова застосунку на основі мікросервісної архітектури.

## **Гнучке управління продуктами для команд розробників програмного забезпечення**

*Опис:* Кожна організація створює продукти для вирішення проблем своїх клієнтів. Продакт-менеджери є критично важливою частиною організації, яка стежить за тим, щоб потреби клієнтів, які постійно змінюються, та ринкові тенденції були враховані та трансформовані в чудові рішення, які допомагають бізнесу досягти бажаних результатів.

На цій дисципліні студенти отримують фундаментальне розуміння практик управління продуктом.

Це дасть їм комплексне уявлення про повний життєвий цикл управління продуктом.

### **Основні результати навчання:**

- Оцінювати, аналізувати та критикувати різні стратегії вдосконалення продукту після запуску.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах, що стосуються вимірювання залученості користувачів.
- Пропонувати відповідні рішення для складних і мінливих проблем успіху або невдачі продукту в реальному інженерному та науковому контекстах.

<b>Пререквізити (умови для входу до навчання)</b>	Відсутні
<b>Інструменти, технології, які опановуються на дисципліні</b>	Product Vision Value Proposition CusDev Persona Framework Agile-методології CJM Product Backlog North Star Framework Модель AARRR Business Model Canvas Юніт-економіка Pitch Deck
<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	<ul style="list-style-type: none"> <li>- Здатність ефективно працювати в команді.</li> <li>- Здатність придумати ідею стартапу, розробити бачення продукту та його ціннісну пропозицію.</li> <li>- Здатність дослідити цільовий ринок, визначити цільового споживача та основних конкурентів.</li> <li>- Здатність застосовувати на практиці принципи та цінності Agile у рамках управління командою.</li> <li>- Здатність описувати беклог продукту, створювати MVP продукту.</li> <li>- Здатність визначити систему метрик продукту та розробити конверсійну воронку.</li> <li>- Здатність розробляти бізнес модель стартапу.</li> <li>- Здатність застосовувати концепцію unit-економіки та аналізувати показники для оптимізації продукту.</li> <li>- Здатність підготувати презентацію власного проєкту (Pitch Deck) для пошуку фінансування, підготувати свій стартап.</li> </ul>
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	Фінальний командний проєкт. Груповий проєкт покликаний надати навички «упаковувати» та презентувати продуктові ідеї свого стартапу. Отже, у фінальному проєкті команда має надати результати проведених досліджень споживачів, конкурентів, аналізу ринку, прописаної воронки AARRR з метриками на кожному етапі та найголовнішою метрикою North Star, пояснити ідею, бачення, цінність та функціональність продукту, його бізнес-модель.

## **Кар'єрні стратегії та м'які навички для ІТ-спеціалістів**

*Опис:* Ця дисципліна покликана навчити ІТ-спеціалістів м'яким навичкам та кар'єрним стратегіям, необхідних для досягнення успіху в технологічній галузі. Дисципліна базується на проєктній роботі та охоплює низку тем, таких як комунікативні навички, робота в команді, тайм-менеджмент, лідерство, нетворкінг та кар'єрний розвиток.

Дисципліна охоплює весь життєвий цикл технологічного проєкту, від збору вимог до доставки та обслуговування. Студенти дізнаються, як ефективно спілкуватися із зацікавленими сторонами, раціонально розподіляти свій час, керувати командою та ефективно співпрацювати в командному середовищі.

Дисципліна також охоплює такі аспекти розвитку кар'єри, як нетворкінг та побудова професійних відносин, створення особистого бренду та розробка кар'єрного плану. Студенти дізнаються, як визначити свої сильні та слабкі сторони, а також як використовувати свої навички та досвід для просування кар'єри в технологічній індустрії.

### **Основні результати навчання:**

- Розвивати та демонструвати навички ефективної комунікації.
- Ефективно співпрацювати в командному середовищі.
- Розвивати та демонструвати лідерські навички.
- Будувати та підтримувати професійні стосунки.
- Розробляти та виконувати кар'єрний план.

<b>Пререквізити (умови для входу до навчання)</b>	Відсутні
<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	- Ефективні стратегії кар'єрного росту в галузі ІТ та засоби для досягнення своїх професійних цілей. - Вміння комунікувати ефективно, працювати в команді та управляти часом, що покращить твою продуктивність і сприятиме успіху в роботі. - Вміння розвивати особисті навички, впевненість в собі та лідерські якості, що допоможе тобі побудувати успішну кар'єру в галузі ІТ.
<b>Фінальний проєкт, який виконує студент та опис проєкту</b>	Фіналізуюче домашнє завдання

## **Прикладні комп'ютерні науки: Дипломний проєкт**

*Опис:* Це дисципліна, що базується на проєктній роботі і має на меті формування необхідних навичок для створення програмних систем на основі вебтехнологій. Дисципліна охоплює весь життєвий цикл

створення програмних проєктів, від збору вимог та визначення обсягу робіт з документації на продукт від проєктування архітектури системи, і аж до доставки та обслуговування програмної системи.

Дисципліна охоплює як фронтенд (створення інтерфейсів на основі браузера для користувачів з використанням фронтенд вебфреймворків), так і бекенд (тобто сервер, на якому функціонує API для надання інформації фронтенду), і який працює на основі SQL або подібної системи управління базами даних для зберігання інформації.

Дисципліна охоплює всі аспекти реалізації програмного проєкту, включаючи безпеку, автентифікацію та авторизацію користувачів, моніторинг та аналітику, а також підтримку проєкту. Дисципліна також охоплює такі аспекти супроводу проєкту, як використання системи контролю версій, налаштування конвеєрів безперервної інтеграції та розгортання, а також системи відстеження помилок.

#### **Основні результати навчання:**

- Оцінювати, аналізувати та критикувати різні стратегії організації та розгортання програмного забезпечення в хмарі.
- Порівнювати та оцінювати різні методології, рекомендовані в наукових джерелах, що стосуються дизайну схем даних.
- Пропонувати відповідні рішення для складних і мінливих проблем супроводу програмного забезпечення в реальних інженерних і наукових контекстах.

<b>Пререквізити (умови для входу до навчання)</b>	Знання та навички усіх пройдених попередньо дисциплін
<b>Знання, вміння, навички, які отримує студент після вивчення дисципліни</b>	<ul style="list-style-type: none"><li>- Створення та реалізація власного проєкту з комп'ютерних наук, застосовуючи знання та навички, які здобули під час навчання.</li><li>- Застосування теоретичних знань на практиці, вирішуючи реальні проблеми та використовуючи сучасні технології та методів.</li><li>- Знання про процес розробки програмного забезпечення від постановки завдання до впровадження та підтримки проєкту, що допоможе вам розвинути вміння керування проєктами та виконання комплексних завдань.</li></ul>

--	--