

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ



ЗАТВЕРДЖУЮ

Декан факультету інформаційних технологій

Тетяна ГОВОРУЩЕНКО

1 вересня 2026 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Основи машинного навчання

Призначення Робочої програми

Рівень вищої освіти

Мова навчання

Обсяг дисципліни, кредитів ЄКТС

Статус дисципліни

Факультет

Кафедра

Для освітніх програм різних спеціальностей

Перший (бакалаврський)

Українська

8

Вибіркова

Інформаційних технологій

Інженерії програмного забезпечення

Форма здобуття освіти	Обсяг дисципліни		Кількість годин						Форма семестрового контролю
			Аудиторні заняття					Самостійна робота (в т.ч. ІРС)	Залік
	Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття		
Д	8	240	82	16	34	32		158	+

Робоча програма складена на основі освітніх програм підготовки бакалавра та стандарту вищої освіти спеціальності.

Робоча програма складена  д-р фіз.-мат. наук проф. Леонід БЕДРАТЮК

Схвалена на засіданні кафедри Інженерії програмного забезпечення

Протокол від 31 серпня 2026 № 1. Зав. кафедри  Леонід БЕДРАТЮК

Хмельницький 2026

Посада	Назва кафедри	Підпис	Ім'я, ПРІЗВИЩЕ
Зав. кафедри, <u>д-р фіз.-мат. наук, проф.</u>	Інженерії програмного забезпечення		<u>Леонід БЕДРАТЮК</u>
Гарант ОП, <u>д-р фіз.-мат. наук, проф.</u>	Інженерії програмного забезпечення		<u>Леонід БЕДРАТЮК</u>

ОСНОВИ МАШИННОГО НАВЧАННЯ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	-
Кількість встановлених кредитів ЄКТС	8
Форми навчання, для яких викладається дисципліна	Очна(Денна)

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен: *володіти* професійною термінологією та основними поняттями машинного навчання; *пояснювати* принципи навчання моделей на даних, узагальнювальну здатність, причини недонавчання і перенавчання, роль навчальної, валідаційної та тестової вибірок; *формулювати* прикладні задачі як задачі машинного навчання, визначати об'єкти спостереження, ознаки, цільову змінну та критерії оцінювання; *аналізувати* набори даних, їх структуру, якість, розподіли, пропущені значення, аномалії та взаємозв'язки між ознаками; *виконувати* попередню обробку даних, кодування категоріальних ознак, масштабування, відбір та конструювання ознак із дотриманням вимог щодо запобігання витоків даних; *проектувати* коректні та відтворювані ML-експерименти, використовувати baseline, валідацію і крос-валідацію; *будувати та застосовувати* класичні моделі машинного навчання для задач регресії, класифікації, кластеризації, зниження розмірності та виявлення аномалій; *оцінювати, порівнювати та обґрунтовувати* вибір моделей за відповідними метриками, стабільністю результатів, характером похибок та інженерними обмеженнями; *інтерпретувати* результати роботи моделей і аналізувати причини помилкових передбачень; *використовувати* бібліотеки Pandas, NumPy, scikit-learn, Matplotlib та інші програмні засоби для аналізу даних, побудови, оцінювання і порівняння моделей; *використовувати* засоби AutoML для автоматизованого пошуку моделей і гіперпараметрів, аналізувати отримані конфігурації та перевіряти коректність проведених експериментів; *застосовувати* системи генеративного штучного інтелекту як інструмент для пошуку ідей, побудови та пояснення програмного коду, пошуку помилок, проектування експериментів, аналізу результатів і підготовки документації; *критично оцінювати та верифікувати* код, пояснення, моделі, експериментальні схеми та висновки, отримані за допомогою систем штучного інтелекту, шляхом виконання програмного коду, проведення експериментів, перевірки метрик та аналізу результатів; *організовувати* відтворюваність ML-проектів шляхом версіонування коду, даних, конфігурацій, моделей та результатів експериментів; *інтегрувати* навчений ML-компонент у програмну систему, *тестувати* його функціональні та нефункціональні характеристики, *оцінювати* швидкість inference, використання ресурсів та надійність; *пояснювати* необхідність моніторингу ML-компонента після розгортання та повторного навчання при зміні даних або погіршенні якості моделі; *самостійно обґрунтовувати* прийняті рішення щодо даних, моделей, метрик, експериментів та способу інтеграції ML-компонента і нести відповідальність за остаточний результат.

Зміст навчальної дисципліни. Машинне навчання в інженерії програмного забезпечення. Постановка задач машинного навчання. Дані, їх аналіз та попередня обробка. Розвідувальний аналіз даних. Ознаки та цільова змінна. Якість і репрезентативність даних. Поділ даних на навчальну, валідаційну та тестову вибірки. Data leakage. Pipeline та відтворювана попередня обробка. Оцінювання, порівняння та вибір моделей машинного навчання. Baseline. Метрики регресії та класифікації. Крос-валідація. Аналіз похибок. Регресійні моделі. Лінійна та поліноміальна регресія. Складність моделі, недонавчання, перенавчання та регуляризація. Лінійні та ймовірнісні методи класифікації. Логістична регресія. Наївний басівський класифікатор. Метричні методи. Метод k -найближчих сусідів. Метод опорних векторів. Ядерні методи. Дерева рішень. Ансамблеве навчання. Bagging, випадкові ліси та boosting. Навчання без учителя. Кластеризація. Метод головних компонент та зниження розмірності. Виявлення аномалій. Автоматизація машинного навчання та AutoML. Генеративний штучний інтелект як інструмент ML-інженера. Відтворюваність ML-експериментів, версіонування даних, коду,

конфігурацій і моделей. Основи MLOps. Інтеграція, тестування, розгортання, моніторинг і супровід ML-компонента програмної системи. Межі класичного машинного навчання та основи нейронних мереж..

Запланована навчальна діяльність: Мінімальний обсяг навчальних занять в одному кредиті ЄКТС навчальної дисципліни становить 10 годин

Форми (методи) навчання: лекції (з використанням методів проблемного навчання, візуалізації та аналізу прикладних ситуацій); лабораторні заняття; практичні заняття; самостійна робота; виконання ML-експериментів і розроблення програмних прототипів; виконання навчальних завдань із контрольованим використанням систем генеративного штучного інтелекту та AutoML для генерації і модифікації програмного коду, пошуку варіантів розв'язання, проєктування експериментів, аналізу результатів, виявлення помилок і підготовки документації з обов'язковою перевіркою отриманих результатів студентом.

Форми оцінювання результатів навчання: захист лабораторних і практичних робіт; тестування усне опитування; оцінювання результатів ML-експериментів, їх відтворюваності та обґрунтованості висновків; оцінювання програмних прототипів із ML-компонентами; завдання з використанням AutoML і систем генеративного штучного інтелекту з перевіркою коректності отриманого коду, моделей та результатів; усний захист прийнятих інженерних рішень, пояснення роботи використаного програмного коду та аналізу отриманих результатів.

Вид семестрового контролю: залік

Навчальні ресурси:

1. Jung, A. (2022). Machine Learning: The Basics. Springer Nature Singapore. – 751 p
2. Kharwal, A. (2023). Machine Learning Algorithms: Handbook. Clever Fox Publishing. – 234 p
3. Bhasin, H. (2023). Machine Learning for Beginners - 2nd Edition: Build and Deploy Machine Learning Systems Using Python (English Edition). BPB Publications – 384 p
4. Geetha, T. V., Sendhilkumar, S. (2023). Machine Learning: Concepts, Techniques and Applications. CRC Press. – 478 p
5. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
6. Інституційний репозиторій ХНУ. URL : <https://elar.khmnu.edu.ua/home>
7. Модульне середовище для навчання. <https://msn.khmnu.edu.ua/course/view.php?id=8670>

Викладач: д-р фіз.-мат. наук, проф. Бедратюк Л.П., ст. викладач Бедратюк Г.І.

3. ПОЯСНЮВАЛЬНА ЗАПИСКА

Дисципліна «Основи машинного навчання» відноситься до циклу вибіркових дисциплін і сприяє поглибленню фахових умінь з побудови, тренування та впровадження моделей класичного машинного навчання для вирішення прикладних задач у галузі розробки програмного забезпечення. Вивчення дисципліни формує розуміння машинного навчання як експериментального та інженерного процесу, що охоплює постановку прикладної задачі, аналіз і підготовку даних, побудову та порівняння моделей, оцінювання їх узагальнювальної здатності, інтерпретацію результатів, інтеграцію ML-компонента у програмну систему та контроль його роботи під час експлуатації. Опанування дисципліни передбачає практичну роботу з бібліотеками Python для аналізу даних і машинного навчання, зокрема NumPy, Pandas, Matplotlib та scikit-learn, застосування засобів автоматизації експериментів і AutoML, організацію відтворюваності результатів, версіонування коду, даних, конфігурацій і моделей. Значна увага приділяється задачам, характерним для інженерії програмного забезпечення: прогнозуванню дефектів, оцінюванню трудомісткості та часу виконання задач, класифікації повідомлень про помилки, аналізу логів і телеметрії, виявленню аномалій, прогнозуванню ризиків релізів та оцінюванню характеристик якості програмного забезпечення. У навчальному процесі передбачено контрольоване використання систем генеративного штучного інтелекту як професійного інструмента ML-інженера. Студенти можуть використовувати такі системи для пошуку ідей, генерації та пояснення програмного коду, аналізу альтернативних рішень, проєктування експериментів, пошуку помилок, інтерпретації результатів і підготовки документації. Результати, отримані за допомогою систем штучного інтелекту, підлягають обов'язковій перевірці шляхом виконання програмного коду, проведення експериментів, аналізу метрик та обґрунтування прийнятих рішень. Такий підхід формує здатність ефективно використовувати сучасні засоби автоматизації та зберігати відповідальність за коректність кінцевого інженерного результату. Навчання передбачає роботу з технічною документацією та іншими професійними джерелами англійською мовою. Під час виконання лабораторних і практичних завдань студенти розвивають уміння формулювати та перевіряти гіпотези, аргументовано порівнювати альтернативні рішення, критично оцінювати результати, документувати експерименти та обґрунтовувати прийняті інженерні рішення.

Мета дисципліни. Сформувати у здобувачів системне розуміння основ класичного машинного навчання та його місця в інженерії програмного забезпечення; сформувати теоретичні знання і практичні вміння з аналізу та підготовки даних, побудови, оцінювання, порівняння й інтерпретації моделей машинного навчання; виробити навички проєктування коректних і відтворюваних ML-експериментів, використання бібліотек Python, AutoML і систем генеративного штучного інтелекту; сформувати здатність інтегрувати, тестувати, розгортати та супроводжувати ML-компоненти програмних систем і обґрунтовувати прийняті інженерні рішення.

Предмет дисципліни. Методи, алгоритми, моделі, дані та програмні засоби класичного машинного навчання, принципи побудови і проведення ML-експериментів, оцінювання та порівняння моделей, автоматизація процесів машинного навчання, а також інженерні підходи до інтеграції, тестування, відтворюваності, розгортання і супроводу ML-компонентів програмних систем.

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен: *володіти* професійною термінологією та основними поняттями машинного навчання; *пояснювати* принципи навчання моделей на даних, узагальнювальну здатність, причини недонавчання і перенавчання, роль навчальної, валідаційної та тестової вибірок; *формулювати* прикладні задачі як задачі машинного навчання, визначати об'єкти спостереження, ознаки, цільову змінну та критерії оцінювання; *аналізувати* набори даних, їх структуру, якість, розподіли, пропущені значення, аномалії та взаємозв'язки між ознаками; *виконувати* попередню обробку даних, кодування категоріальних ознак, масштабування, відбір та конструювання ознак із дотриманням вимог щодо запобігання витоку

даних; *проектувати* коректні та відтворювані ML-експерименти, використовувати baseline, валідацію і крос-валідацію; *будувати та застосовувати* класичні моделі машинного навчання для задач регресії, класифікації, кластеризації, зниження розмірності та виявлення аномалій; *оцінювати, порівнювати та обґрунтовувати* вибір моделей за відповідними метриками, стабільністю результатів, характером похибок та інженерними обмеженнями; *інтерпретувати* результати роботи моделей і аналізувати причини помилкових передбачень; *використовувати* бібліотеки Pandas, NumPy, scikit-learn, Matplotlib та інші програмні засоби для аналізу даних, побудови, оцінювання і порівняння моделей; *використовувати* засоби AutoML для автоматизованого пошуку моделей і гіперпараметрів, аналізувати отримані конфігурації та перевіряти коректність проведених експериментів; *застосовувати* системи генеративного штучного інтелекту як інструмент для пошуку ідей, побудови та пояснення програмного коду, пошуку помилок, проектування експериментів, аналізу результатів і підготовки документації; *критично оцінювати та верифікувати* код, пояснення, моделі, експериментальні схеми та висновки, отримані за допомогою систем штучного інтелекту, шляхом виконання програмного коду, проведення експериментів, перевірки метрик та аналізу результатів; *організовувати* відтворюваність ML-проектів шляхом версіонування коду, даних, конфігурацій, моделей та результатів експериментів; *інтегрувати* навчений ML-компонент у програмну систему, *тестувати* його функціональні та нефункціональні характеристики, *оцінювати* швидкість inference, використання ресурсів та надійність; *пояснювати* необхідність моніторингу ML-компонента після розгортання та повторного навчання при зміні даних або погіршенні якості моделі; *самостійно обґрунтовувати* прийняті рішення щодо даних, моделей, метрик, експериментів та способу інтеграції ML-компонента і нести відповідальність за остаточний результат.

Політика використання систем штучного інтелекту. У межах дисципліни «Основи машинного навчання» використання систем генеративного штучного інтелекту дозволяється і розглядається як складова професійної діяльності ML-інженера, якщо викладачем для конкретного завдання не встановлено інший режим. Системи ШІ можуть використовуватися для пошуку ідей, генерації та модифікації програмного коду, пояснення алгоритмів, пошуку помилок, підбору моделей і гіперпараметрів, проектування експериментів, аналізу результатів та підготовки документації. Допускається отримання за допомогою ШІ повного програмного коду для виконання навчального завдання.

Результат, отриманий за допомогою ШІ, не вважається автоматично правильним і підлягає перевірці студентом шляхом виконання коду, проведення експериментів, аналізу метрик і результатів. Студент несе відповідальність за коректність даних, preprocessing, експериментальної схеми, програмного коду, моделей та зроблених висновків. Під час захисту студент повинен пояснити прийняті рішення, продемонструвати розуміння використаного коду та методів, уміти виявляти й виправляти помилки, а також за вимогою викладача модифікувати або доповнити отримане рішення.

Використання систем штучного інтелекту відповідно до встановлених вимог дисципліни **не є порушенням академічної доброчесності**. Використання ШІ у завданнях або під час форм контролю, для яких викладач окремо встановив режим самостійного виконання без ШІ, не допускається.

4. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва розділу (теми)	Кількість годин, відведених на:			
	лекції	лабораторні	практичні	СРС
Тема 1. Основи машинного навчання, дані та побудова коректного ML-експерименту	6	0	8	34
Тема 2. Класичні методи навчання з учителем	8	16	12	56
Тема 3. Навчання без учителя	1	4	4	24
Тема 4. Автоматизація, відтворюваність та інженерія ML-систем	1	14	8	44
Разом:	16	34	32	158

5. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

5.1. Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
1	Лекція 1. Машинне навчання в інженерії програмного забезпечення. Машинне навчання як спосіб побудови програмної поведінки. Місце ML у програмній інженерії. Основні типи задач: регресія, класифікація, кластеризація, зниження розмірності, виявлення аномалій. Задачі ML у розробленні, тестуванні, релізі та експлуатації програмних систем. Постановка ML-задачі, роль даних, узагальнювальна здатність, експериментальний характер машинного навчання. Життєвий цикл, якість, обмеження та автоматизація ML-компонента. Роль інженера програмного забезпечення. <i>Літ.: [1] с.18-47, [2] с.34-65, [3] с.8-24</i>	2
2	Лекція 2. Дані, попередня обробка та відтворювані pipeline EDA, типи ознак, пропущені значення, кодування, масштабування, конструювання ознак, витік даних, Pipeline, ColumnTransformer, базове поняття контракту даних. <i>Літ.: [1] с. 48-83, [3] 25-74</i>	2
3	Лекція 3. Проєктування та оцінювання ML-експерименту Baseline як початкова точка експерименту. Навчальна, валідаційна і тестова вибірки. Крос-валідація. Метрики регресії та класифікації. Аналіз похибок і стабільності результатів. Порівняння кандидатних моделей, налаштування гіперпараметрів та принципи обґрунтованого вибору моделі. Відтворюваність експериментального порівняння.	2
4	Лекція 4. Регресійні моделі, складність і регуляризація. Лінійна та поліноміальна регресія. Функція втрат, метод найменших квадратів і градієнтний спуск. Багатовимірна регресія. Складність моделі, недонавчання та перенавчання. L1- та L2-регуляризація, Lasso, Ridge та Elastic Net. Вплив гіперпараметрів регуляризації на узагальнювальну здатність моделі. <i>Літ.: [5] С. 110-127;</i>	2

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
5	Лекція 5. Лінійні та ймовірнісні методи класифікації. Задача класифікації та границя рішення. Логістична регресія, ймовірність класу, порогове рішення та регуляризація. Формула Баєса. Наївний баєсівський класифікатор, гаусівська та бернуллівська моделі. Бінарна і багатокласова класифікація. Інтерпретація результатів класифікації. <i>Літ.: [2] с.636-674, [10] с.309-378</i>	2
6	Лекція 6. Метричні методи та метод опорних векторів. Метричні методи класифікації. Відстані у просторі ознак, метод k-найближчих сусідів та роль масштабування. Лінійна машина опорних векторів, максимальна межа та опорні вектори. Нелінійна класифікація, ядерний підхід, поліноміальне та гаусівське ядра. Параметри C і gamma, особливості застосування SVM. <i>Літ.: [1] с.699-796, [10] с.379-391</i>	2
7	Лекція 7. Дерева рішень та ансамблеве навчання. Побудова дерев рішень. Ентропія, інформаційний виграш та індекс Джині. Контроль складності, обмеження глибини, стрижка та важливість ознак. Дерева для класифікації і регресії. Ідея ансамблевого навчання. Bagging, bootstrap, випадкові ліси та градієнтний boosting. Порівняння окремих моделей і ансамблів. <i>Літ.: [1] с.204-303, [10] с.179-218</i>	2
8	Лекція 8. Навчання без учителя: кластеризація, зниження розмірності та пошук аномалій. Навчання на нерозмічених даних. Кластеризація та метод k-середніх. Оцінювання структури кластерів. Проблема високої розмірності. Метод головних компонент (РСА) та геометрична інтерпретація зниження розмірності. Використання зниження розмірності для візуалізації та аналізу даних. Основні підходи до виявлення аномалій. <i>Літ.: [1] с.699-796, [5] с.379-391</i>	2
Разом		16

5.2 Зміст практичних занять

Перелік практичних занять

№ з/п	Тема практичного заняття	Кількість годин
1	Аналіз даних та їх попередня обробка <i>Літ.: [2] С. 44-50; [3] С. 65-160</i> <i>Літ.: [2] С. 44-50; [4] С. 65-160</i>	4
2	Побудова та оцінювання коректного ML-експерименту <i>Літ.: [2] С. 12-32; [5] С. 31-67;</i>	4
3	Перенавчання, складність моделі та регуляризація <i>Літ.: [2] С. 50-59; [4] С. 18-62;</i>	4
4	Пороги, вартість помилок і прийняття рішень у класифікації. <i>Літ.: [2] С. 12-32; [5] С. 31-67;</i>	4

№ з/п	Тема практичного заняття	Кількість годин
5	Багатокритеріальний вибір ML-моделі за інженерними обмеженнями <i>Лім.: [2] С. 59-75;</i>	4
6	Інтерпретація та перевірка корисності результатів навчання без учителя <i>Лім.: [5] С. 75-104;</i>	4
7	Критичний аудит ML-рішення, створеного за допомогою ШІ або AutoML <i>[9] С. 18-62;</i>	4
8	Архітектура, тестування та експлуатаційна якість ML-компонента <i>[5] С. 209-279</i>	4
Разом		32

5.3 Зміст лабораторних занять

Перелік лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
1	Регресійне моделювання даних: від baseline до обґрунтованої моделі <i>Лім.: [2] С. 44-50; [4] С. 65-160</i>	4
2	Класифікація та аналіз помилок моделі <i>Лім.: [1] С. 84-110; [3] С. 161-180</i>	4
3	Порівняння метричних методів і машин опорних векторів <i>Лім.: [2] С. 44-50</i>	4
4	Дерева рішень та ансамблі: якість, стійкість та інтерпретація <i>Лім.: [2] С. 122-138;</i>	4
5	Кластеризація, зниження розмірності та виявлення аномалій у редукованому просторі ознак <i>Лім.: [5] С. 75-104;</i>	4
6	AutoML і ручне моделювання: контрольоване порівняння <i>Лім.: [2] С. 50-59; [3] С. 18-62; [4] С. 209-279</i>	4
7	Відтворюваний ML-проект: конфігурації, трекінг і версії <i>Лім.: [2] С. 59-75; [9] С. 18-62;</i>	4
8	Розроблення програмного прототипу з ML-компонентом <i>Лім.: [5] С. 110-127;</i>	6
Разом		34

5.4 ЗМІСТ САМОСТІЙНОЇ РОБОТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Самостійна робота студентів полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації та технічної документації, підготовці до практичних і лабораторних занять, виконанні індивідуальних завдань, роботі з наборами даних, проведенні ML-експериментів, аналізі та інтерпретації отриманих результатів, а також підготовці до захисту виконаних робіт. У межах самостійної роботи студенти виконують аналіз і попередню обробку даних, будують та порівнюють моделі машинного навчання, досліджують вплив ознак і гіперпараметрів, аналізують похибки моделей, перевіряють відтворюваність експериментів, працюють із бібліотеками машинного навчання, засобами AutoML, версіонування та організації

ML-проектів. Окремі завдання передбачають опрацювання технічної документації та інших професійних джерел англійською мовою. Під час самостійної роботи допускається використання систем генеративного штучного інтелекту відповідно до політики дисципліни. Студент може застосовувати їх для пошуку ідей, пояснення методів, генерації та модифікації програмного коду, пошуку помилок, підбору моделей і гіперпараметрів, планування експериментів, аналізу результатів та підготовки документації. Для окремих завдань або форм контролю викладач може встановлювати режим виконання без використання систем штучного інтелекту. Результати, отримані за допомогою систем штучного інтелекту, підлягають обов'язковій перевірці студентом. Студент повинен виконати та протестувати отриманий код, перевірити коректність підготовки даних і схеми експерименту, проаналізувати метрики та похибки, виявити можливі помилки або необґрунтовані висновки, за потреби виправити їх та самостійно обґрунтувати остаточне інженерне рішення. Під час захисту студент повинен продемонструвати розуміння використаних методів, програмного коду та отриманих результатів. Для організації самостійної роботи студентам надається сторінка навчальної дисципліни у Модульному середовищі для навчання, де розміщені Робоча програма дисципліни, навчально-методичні матеріали, завдання до практичних і лабораторних робіт, матеріали для самостійного опрацювання, вимоги до оформлення та захисту робіт, а також рекомендації щодо використання систем штучного інтелекту, AutoML та інших програмних засобів

Номер тижня	Вид самостійної роботи	К-ть годин
1-2	Опрацювання лекційного матеріалу. Підготовка до ЛР1 та ПЗ1. Підготовка до захисту ЛР1.	17
3-4	Опрацювання лекційного матеріалу. Підготовка до ЛР2 та ПЗ2. Підготовка до захисту ЛР2 та тестування.	17
5-6	Опрацювання лекційного матеріалу. Підготовка до ЛР3 та ПЗ3. Підготовка до захисту ЛР3	17
7-8	Опрацювання лекційного матеріалу. Підготовка до ЛР4 та ПЗ4. Підготовка до захисту ЛР4 та тестування.	17
9-10	Опрацювання лекційного матеріалу. Підготовка до ЛР5 та ПЗ5. Підготовка до захисту ЛР5	17
11-12	Опрацювання лекційного матеріалу. Підготовка до ЛР6 та ПЗ6. Підготовка до захисту ЛР6 та тестування.	18
13-14	Опрацювання лекційного матеріалу. Підготовка до ЛР7 та ПЗ7. Підготовка до захисту ЛР7.	18
15-16	Опрацювання лекційного матеріалу. Підготовка до ЛР8 та ПЗ8. Підготовка до захисту ЛР8	18
17	Опрацювання лекційного матеріалу. Підготовка до тестування.	19
	Разом	158

*ПЗ – практичне заняття

6. ТЕХНОЛОГІЇ ТА МЕТОДИ НАВЧАННЯ

Процес навчання дисципліни «Основи машинного навчання» спрямований на формування у здобувачів вищої освіти теоретичного розуміння принципів класичного машинного навчання та практичних умінь з аналізу даних, побудови, оцінювання, порівняння й інтеграції моделей у програмні системи. Навчання організовується як послідовний експериментальний та інженерний процес: від постановки прикладної задачі й підготовки даних до перевірки моделей, інтерпретації результатів, забезпечення відтворюваності та створення програмного прототипу з ML-

компонентом. Лекційні заняття проводяться з використанням мультимедійних презентацій, візуалізацій, демонстрацій програмних прикладів та аналізу прикладних ситуацій з інженерії програмного забезпечення. Застосовуються методи пояснення, ілюстрування, проблемного викладу та аналізу експериментальних результатів. Математичні поняття розглядаються у зв'язку з поведінкою алгоритмів і моделей, що сприяє розумінню причин отриманих результатів та формуванню інженерного мислення. Лабораторні заняття орієнтовані на побудову і експериментальну перевірку ML-рішень із використанням Python та бібліотек машинного навчання. Студенти працюють з наборами даних, створюють baseline, будують і порівнюють моделі, налаштовують гіперпараметри, аналізують похибки та перевіряють узагальнювальну здатність моделей. У наступних роботах увага приділяється автоматизації експериментів засобами AutoML, відтворюваності, роботі з конфігураціями та версіями експериментів, а також розробленню програмного прототипу з ML-компонентом. Практичні заняття спрямовані на формування здатності приймати та обґрунтовувати інженерні рішення. Студенти аналізують якість даних і способи їх попередньої обробки, проєктують схеми ML-експериментів, досліджують перенавчання та регуляризацию, обирають метрики і моделі відповідно до прикладної задачі, аналізують результати навчання без учителя, виконують критичний аудит рішень, створених за допомогою AutoML або систем штучного інтелекту, та розглядають питання якості ML-компонента програмної системи. Використовуються аналіз кейсів, дискусії, порівняння альтернативних рішень, постановка і перевірка гіпотез та усне обґрунтування результатів. У межах дисципліни застосовуються системи генеративного штучного інтелекту як професійний інструмент підтримки ML-розроблення. Вони можуть використовуватися для генерації та модифікації програмного коду, пояснення алгоритмів, пошуку помилок, формулювання гіпотез, планування експериментів, підбору можливих моделей і гіперпараметрів, аналізу результатів та підготовки документації. Отримані за допомогою ШІ рішення підлягають перевірці студентом шляхом виконання коду, проведення експериментів, аналізу метрик і похибок та обґрунтування остаточного рішення. Для організації навчальної та експериментальної роботи використовуються Python, Jupyter Notebook або Google Colab, бібліотеки NumPy, Pandas, Matplotlib, scikit-learn, засоби AutoML, системи контролю версій Git та інші програмні інструменти. Результати робіт оформлюються таким чином, щоб забезпечувати можливість повторення експерименту, простеження використаних даних, конфігурацій, моделей та отриманих результатів.

Поєднання лекційних, лабораторних і практичних занять забезпечує поступовий перехід від розуміння принципів машинного навчання до самостійного проєктування, перевірки та обґрунтування ML-рішень у задачах програмної інженерії.

7. МЕТОДИ КОНТРОЛЮ

Поточний контроль здійснюється під час практичних і лабораторних занять, захисту виконаних робіт, а також у межах контрольних заходів, передбачених робочою програмою та графіком освітнього процесу, у тому числі з використанням Модульного середовища для навчання.

Для поточного контролю використовуються такі методи:

- оцінювання результатів роботи на практичних заняттях: аналіз даних, обґрунтування способів рrерrocessing, проєктування ML-експериментів, вибір метрик і моделей, аналіз результатів та участь в обговоренні інженерних рішень;
- тестовий та усний контроль засвоєння теоретичного матеріалу, основних понять, методів і принципів машинного навчання;
- оцінювання виконання та захисту лабораторних робіт з урахуванням коректності програмної реалізації, схеми експерименту, якості моделей, аналізу похибок, відтворюваності та обґрунтованості висновків;
- оцінювання здатності студента пояснити використаний програмний код, логіку роботи моделі, вибір ознак, гіперпараметрів, метрик та способу валідації, а також виконати запропоновану викладачем модифікацію рішення під час захисту;

- оцінювання завдань із використанням AutoML і систем генеративного штучного інтелекту з обов'язковою перевіркою студентом отриманого коду, моделей, експериментальних результатів і висновків;
- оцінювання програмного прототипу з ML-компонентом, його інтеграції, тестування, відтворюваності, якості та готовності до подальшого супроводу;
- контроль дотримання встановленої для конкретного завдання політики використання систем штучного інтелекту.

Під час захисту робіт основна увага приділяється здатності студента самостійно пояснити отриманий результат, виявити можливі помилки, аргументувати прийняті рішення та продемонструвати розуміння використаних методів машинного навчання.

Підсумкова семестрова оцінка виставляється за результатами поточного контролю. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, вважається таким, який *має* академічну заборгованість. Ліквідація академічної заборгованості із семестрового контролю здійснюється у період екзаменаційної сесії або за графіком, встановленим деканатом відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ».

8. ПОЛІТИКА ДИСЦИПЛІНИ

Політика навчальної дисципліни загалом визначається системою вимог до здобувача вищої освіти, що передбачені чинними положеннями Університету про організацію і навчально-методичне забезпечення освітнього процесу. Зокрема, проходження інструктажу з техніки безпеки; відвідування занять з дисципліни є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання за погодженням із лектором може відбуватись в он-лайн режимі. Успішне опанування дисципліни і формування фахових компетентностей і програмних результатів навчання передбачає необхідність підготовки до практичних та лабораторних занять, вивчення теоретичного матеріалу з теми, активно працювати на занятті, брати участь у дискусіях щодо прийнятих рішень при виконанні здобувачами лабораторних робіт тощо.

Здобувачі вищої освіти мають дотримуватися встановлених термінів виконання всіх видів навчальної роботи відповідно до робочої програми навчальної дисципліни. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті (це речення пишите, якщо є лабораторні). Пропущене практичне і лабораторне заняття студент зобов'язаний відпрацювати у встановлений викладачем термін, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється за результатами опитування під час практичних занять, тестування.

Здобувач вищої освіти, виконуючи самостійну роботу або індивідуальну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, плагіат (в т.ч. із використанням мобільних девайсів)). У разі виявлення порушення політики академічної доброчесності в будь-яких видах навчальної роботи здобувач вищої освіти отримує незадовільну оцінку і має повторно виконати завдання з відповідної теми (виду роботи), що передбачені робочою програмою. Будь-які форми порушення академічної доброчесності *не допускаються*.

У межах вивчення навчальної дисципліни здобувачам вищої освіти передбачено визнання і зарахування результатів навчання, набутих шляхом неформальної освіти, що розміщені на доступних платформах (<https://prometheus.org.ua>), які сприяють формування компетентностей і поглибленню результатів навчання, визначених робочою програмою дисципліни, або забезпечують вивчення відповідної теми та/або виду робіт з програми навчальної дисципліни (детальніше у Положенні про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ).

9 ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ СТУДЕНТІВ У СЕМЕСТРІ

При оцінюванні результатів навчання здобувачів вищої освіти з будь-якого виду навчальної роботи (структурної одиниці) рекомендується використовувати наведені нижче узагальнені критерії:

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих ПРН та сформованих компетентностей	Узагальнений зміст критерія оцінювання
Відмінно (високий)	Здобувач вищої освіти глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає логічний виклад відповіді мовою викладання (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними приладами та інструментами, прикладними програмами. Здобувач не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки, демонструє практичні навички з вирішення фахових завдань. При відповіді допустив дві–три несуттєві <i>похибки</i> .
Добре (середній)	Здобувач вищої освіти виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання правил, закономірностей тощо. Відповідь здобувача вищої освіти будується на основі самостійного мислення. Здобувач вищої освіти у відповіді допустив дві–три <i>несуттєві помилки</i> .
Задовільно (достатній)	Здобувач вищої освіти виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь здобувача вищої освіти будується на рівні репродуктивного мислення, здобувач вищої освіти має слабкі знання структури навчальної дисципліни, допускає неточності і <i>суттєві помилки</i> у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
Незадовільно (недостатній)	Здобувач вищої освіти виявив розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка «незадовільно» виставляється здобувачеві вищої освіти, який не може продовжити навчання без додаткової роботи з вивчення навчальної дисципліни.

Структурування дисципліни за видами навчальної роботи і оцінювання результатів навчання студентів

Аудиторна робота																Контрольні заходи	Семестровий контроль
Лабораторні заняття								Практичні заняття								Тестовий контроль	Залік
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	Т 1-2	
Кількість балів за кожний вид навчальної роботи (мінімум-максимум)																	
																6-10	За

3-5	3-5		рейтингом
24-40	24-40	12-20	60-100**

Примітки: *За набрану з будь-якого виду навчальної роботи з дисципліни кількість балів, нижче встановленого мінімуму, здобувач отримує незадовільну оцінку і має її перездати у встановлений викладачем (деканом) термін. Інституційна оцінка встановлюється відповідно до таблиці «Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС».

Оцінювання на практичних заняттях

Оцінка за практичне заняття формується з урахуванням таких складових: розуміння теоретичного матеріалу з теми; правильність аналізу даних і постановки ML-задачі; обґрунтованість вибору способів preprocessing, метрик, моделей та схеми експерименту; здатність інтерпретувати отримані результати, аналізувати похибки та порівнювати альтернативні рішення; уміння професійно аргументувати прийняті інженерні рішення.

Для завдань, що передбачають використання систем генеративного штучного інтелекту або AutoML, додатково оцінюється здатність студента критично перевірити запропонований код, модель, експериментальну схему та висновки, виявити можливі помилки, за потреби виправити їх і самостійно обґрунтувати остаточний результат.

При оцінюванні практичного заняття викладач керується узагальненими критеріями, наведеними у таблиці «**Критерії оцінювання навчальних досягнень здобувача вищої освіти**» (мінімальний позитивний бал – 3 бали, максимальний – 5 балів).

Оцінювання результатів захисту лабораторної роботи

Виконана й оформлена відповідно до встановлених методичними рекомендаціями вимог лабораторна робота комплексно оцінюється викладачем під час її захисту з урахуванням таких критеріїв: коректність постановки задачі та роботи з даними; правильність програмної реалізації; обґрунтованість вибору моделі, метрик і способу валідації; якість проведеного експерименту; аналіз похибок і отриманих результатів; відтворюваність роботи; здатність пояснити використаний програмний код і принципи роботи моделі; уміння оцінити придатність отриманого рішення для поставленої прикладної задачі. Якщо під час виконання лабораторної роботи використовувалися AutoML або системи генеративного штучного інтелекту, студент повинен продемонструвати, що отриманий результат був ним перевірений. Оцінюється здатність пояснити згенерований або модифікований код, відтворити основні етапи експерименту, обґрунтувати прийняті рішення, виявити можливі помилки та за вимогою викладача внести зміни до програмної реалізації або експериментальної схеми.

При оцінюванні практичного заняття викладач керується узагальненими критеріями, наведеними у таблиці «**Критерії оцінювання навчальних досягнень здобувача вищої освіти**» (мінімальний позитивний бал – 3 бали, максимальний – 5 балів).

У випадку виявлення здобувачем рівня знань, нижчого ніж 60 відсотків від максимального балу, встановленого Робочою програмою для кожної структурної одиниці, лабораторна робота йому **не зараховується** і для її захисту він має детальніше опрацювати матеріал з теми роботи, методику її виконання, виправити грубі помилки та повторно вийти на її захист у призначений для цього викладачем час.

Оцінювання результатів тестового контролю

Кожний з тестів, передбачених Робочою програмою, складається із 30 тестових завдань, кожне з яких є рівнозначним. Відповідно до таблиці структурування видів робіт за тестовий контроль здобувач залежно від кількості правильних відповідей може отримати від 3 до 5 балів.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

На тестування відводиться 30 хвилин. Студент проходить тестування в он-лайн режимі у Модульному середовищі для навчання. При отриманні негативної оцінки тест слід перездати до терміну **наступного** контролю.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість правильних відповідей	1-17	18-23	24-26	27-30
Відсоток правильних відповідей	0-59	60-79	80-89	90-100
Кількість балів	-	3	4	5

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС визначається в автоматизованому режимі після внесення викладачем результатів оцінювання у балах з усіх видів навчальної роботи до електронного журналу. Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені нижче у таблиці «Співвідношення».

Семестровий залік виставляється на останньому занятті за умови якщо загальна сума балів, яку накопичив здобувач з дисципліни (іншого освітнього компонента) за результатами **поточного** контролю, знаходиться у межах від 60 до 100 балів. При цьому за інституційною шкалою ставиться оцінка «**зараховано**», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів відповідно до таблиці Співвідношення. Присутність здобувача у цьому випадку не є обов'язковою.

Таблиця – Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Рейтингова шкала балів	Інституційна оцінка (рівень досягнення здобувачем вищої освіти запланованих результатів навчання з навчальної дисципліни)	
		Залік	Іспит/диференційований залік
A	90-100	Зараховано	Відмінно/Excellent – високий рівень досягнення запланованих результатів навчання з навчальної дисципліни, що свідчить про безумовну готовність здобувача до подальшого навчання та/або професійної діяльності за фахом
B	83-89		Добре/Good – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом
C	73-82		Задовільно/Satisfactory – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
D	66-72		
E	60-65		
FX	40-59	Незараховано	Незадовільно/Fail – Низка запланованих результатів навчання з навчальної дисципліни відсутня. Рівень набутих результатів навчання є недостатнім для подальшого навчання та/або професійної діяльності за фахом
F	0-39		Незадовільно/Fail – Результати навчання відсутні

10. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ ЗДОБУТИХ СТУДЕНТАМИ ЗНАНЬ

1. Що таке машинне навчання та яке його місце в інженерії програмного забезпечення?
2. Які основні типи задач розв'язуються методами машинного навчання?
3. Чим відрізняються навчання з учителем і навчання без учителя?
4. Що таке об'єкт спостереження, ознака та цільова змінна?
5. Як перейти від прикладної проблеми до формалізованої ML-задачі?
6. Чому одна прикладна проблема може мати кілька різних ML-формалізацій?
7. Які задачі машинного навчання характерні для інженерії програмного забезпечення?
8. Що означає узагальнювальна здатність ML-моделі?
9. У чому полягають недонавчання та перенавчання?
10. Чому висока якість на навчальній вибірці не гарантує хорошої роботи моделі на нових даних?
11. Що таке набір даних та якими типами ознак можуть описуватися об'єкти?

12. У чому полягає розвідувальний аналіз даних (EDA)?
13. Які статистичні характеристики використовуються під час початкового аналізу даних?
14. Як аналізуються розподіли числових ознак?
15. Як виявляються викиди та аномальні значення?
16. Як аналізуються категоріальні ознаки?
17. Що показує кореляція між ознаками?
18. Чому кореляція не є доказом причинно-наслідкового зв'язку?
19. Які способи обробки пропущених значень використовуються у машинному навчанні?
20. Для чого виконується масштабування ознак?
21. Чим відрізняються стандартизація та нормалізація?
22. Як кодуються категоріальні ознаки?
23. Що таке feature engineering та відбір ознак?
24. Що таке data leakage і чому він спотворює результати ML-експерименту?
25. Для чого використовуються Pipeline та ColumnTransformer?
26. Для чого дані поділяються на навчальну, валідаційну та тестову вибірки?
27. Що таке baseline і яку роль він відіграє в ML-експерименті?
28. Що таке крос-валідація?
29. Як забезпечити коректне порівняння кількох моделей?
30. Які метрики використовуються для оцінювання регресійних моделей?
31. Які метрики використовуються для оцінювання моделей класифікації?
32. Що таке матриця помилок класифікації?
33. Що характеризують Precision, Recall та F1-score?
34. Як вибір метрики залежить від прикладної задачі та вартості помилок?
35. У чому полягає аналіз похибок моделі?
36. Що таке гіперпараметри моделі та як вони відрізняються від параметрів?
37. Які принципи використовуються для обґрунтованого вибору фінальної моделі?
38. У чому полягає задача лінійної регресії?
39. Який зміст має функція втрат у задачі регресії?
40. У чому полягає ідея градієнтного спуску?
41. Що таке поліноміальна регресія та як складність моделі впливає на її поведінку?
42. Для чого використовується регуляризація?
43. Чим відрізняються L1- та L2-регуляризація?
44. У чому полягає логістична регресія?
45. Яку роль відіграє поріг класифікації?
46. На яких припущеннях ґрунтується наївний баєсівський класифікатор?
47. У чому полягає метод k -найближчих сусідів?
48. Як вибір метрики відстані та масштабування ознак впливає на метричні методи?
49. У чому полягає геометрична ідея методу опорних векторів?
50. Що таке опорні вектори та максимальна межа?
51. Для чого в SVM використовуються ядра?
52. Яку роль відіграють параметри C та γ ?
53. Як будується дерево рішень?
54. Що характеризують критерій Джині, ентропія та інформаційний виграш?
55. Як контролюється складність дерева рішень?
56. Що таке важливість ознак у деревоподібних моделях?
57. У чому полягає ідея ансамблевого навчання?
58. Що таке bagging і bootstrap?
59. Як працює випадковий ліс?
60. У чому полягає загальна ідея boosting?
61. Які особливості навчання без учителя?
62. У чому полягає задача кластеризації?
63. Як працює метод k -середніх?
64. Як можна оцінювати якість кластеризації?
65. У чому полягає «прокляття розмірності»?

66. Для чого використовується зниження розмірності?
67. У чому полягає метод головних компонент (РСА)?
68. Що характеризує пояснена дисперсія в РСА?
69. У чому полягає задача виявлення аномалій?
70. Які приклади виявлення аномалій характерні для програмних систем?
71. Що таке AutoML?
72. Які етапи ML-експерименту можуть бути автоматизовані?
73. Як автоматизується пошук гіперпараметрів?
74. Чому модель, автоматично вибрана AutoML, потребує додаткової перевірки?
75. Як генеративний штучний інтелект може використовуватися в роботі ML-інженера?
76. Які ризики пов'язані з використанням програмного коду та висновків, згенерованих системами ШІ?
77. Як студент або ML-інженер повинен верифікувати результати, отримані за допомогою ШІ?
78. Які рішення в ML-проекті залишаються відповідальністю інженера незалежно від рівня автоматизації?
79. Що означає відтворюваність ML-експерименту?
80. Які артефакти ML-проекту необхідно версіювати?
81. Яку роль відіграють конфігураційні файли та журналювання експериментів?
82. Що таке MLOps і які задачі він вирішує?
83. Як ML-модель інтегрується у програмну систему?
84. Які аспекти ML-компонента необхідно тестувати?
85. Чим якість ML-моделі відрізняється від якості ML-системи?
86. Які нефункціональні характеристики важливі для ML-компонента?
87. Що таке inference і які характеристики його продуктивності потрібно контролювати?
88. Для чого потрібний моніторинг ML-моделі після розгортання?
89. Як зміна розподілу даних у часі може впливати на якість моделі?
90. За яких умов виникає потреба у повторному навчанні моделі?
91. Які основні обмеження машинного навчання?
92. Чому передбачення ML-моделі не може розглядатися як гарантовано правильне для кожного окремого об'єкта?
93. Чому для критичних рішень може бути необхідною перевірка людиною?
94. Яка роль інженера програмного забезпечення протягом життєвого циклу ML-компонента?
95. У яких випадках класичні методи машинного навчання можуть бути недостатніми?
96. Якою є загальна ідея штучного нейрона та багатошарового перцептрона?
97. Чим підхід нейронних мереж концептуально відрізняється від розглянутих у курсі класичних ML-моделей?

11. НАВЧАЛЬНО-МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Освітній процес з дисципліни «Основи машинного навчання» забезпечений необхідними навчально-методичними матеріалами, що розміщені в Модульному середовищі для навчання MOODLE:

Курс «Основи машинного навчання»: <https://msn.khmnu.edu.ua/course/view.php?id=8670>

12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Jung, A. (2022). Machine Learning: The Basics. Springer Nature Singapore. – 751 p
2. Kharwal, A. (2023). Machine Learning Algorithms: Handbook. Clever Fox Publishing. – 234 p
3. Bhasin, H. (2023). Machine Learning for Beginners - 2nd Edition: Build and Deploy Machine Learning Systems Using Python (English Edition). BPB Publications – 384 p
4. Geetha, T. V., Sendhilkumar, S. (2023). Machine Learning: Concepts, Techniques and

Applications. CRC Press. – 478 p

5. Харченко В. О. Основи машинного навчання : навч. посіб. / В. О. Харченко. – Суми : Сумський державний університет, 2023. – 264 с.

Допоміжна

6. Кононова К. Ю. Машинне навчання: методи та моделі: підручник для бакалаврів, магістрів та докторів філософії спеціальності 051 «Економіка» / К. Ю. Кононова. – Харків: ХНУ імені В. Н. Каразіна, 2020. – 301 с.
7. Биков, М. М. Основи інтелектуальних технологій. Частина 2. Технології машинного навчання : електронний навчальний посібник комбінованого (локального та мережного) використання [Електронний ресурс] / Биков М. М., Ковтун В. В., Гришук Т. В. – Вінниця : ВНТУ, 2024. – 153 с.
8. Машинне навчання : навчальний посібник / Т. М. Басюк, В. В. Литвин, Л. М. Захарія, Н. Е. Кунанець ; за науковою редакцією д.т.н., проф., В. В. Пасічника. – 3-тє видання, стереотипне. – Львів : «Новий Світ – 2000», 2025. – 330 с.
9. Л.М. Олещенко, Машинне навчання: комп'ютерний практикум з дисципліни «Машинне навчання» [Електронний ресурс]: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» –Київ: КПІ ім. Ігоря Сікорського, 2022. – 92 с.
10. S. Raschka, V. Mirjalili, Python Machine Learning - Second Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow, Packt, 2017.
11. F. Chollet, Deep Learning with Python - Second Edition, Manning Publications Co, 2021
12. I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016
13. A. Müller, S. Guido, Introduction to Machine Learning with Python, O'Reilly Media, Inc, 2017.
14. Karella, T., Suk, T., Košík, V., Bedratyuk, L., Kerepecký, T., & Flusser, J. (2024). 3D Non-separable Moment Invariants and Their Use in Neural Networks. *SN Computer Science*, 5(8), 1-16.
15. Бедратюк Л.П., Бедратюк Г.І., Гурман І.В., Метод збільшення окремого зображення на основі багатовиходової регресії та моментної метрики, “Вимірювальна та обчислювальна техніка в технологічних процесах”, 2023, №1, с.149-157
16. Jan Flusser, Tomáš Suk, Leonid Bedratyuk, Tomáš Karella, Non-separable moments in 3D, Computer Analysis of Images and Patterns: 20th International Conference, CAIP 2023, Limassol, Cyprus, September 25–28, 2023, Proceedings, Part I Sep 2023 Pages 295–305 https://doi.org/10.1007/978-3-031-44237-7_28

11. ІНФОРМАЦІЙНІ РЕСУРСИ

8. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
9. Інституційний репозитарій ХНУ. URL : <https://elar.khmnu.edu.ua/home>
10. Модульне середовище для навчання. <https://msn.khmnu.edu.ua/course/view.php?id=8670>
11. Prometheus: Машинне навчання. <https://prometheus.org.ua/prometheus-free/machine-learning/>
12. Машинне навчання. <https://w3schoolsua.github.io/ai/index.html#gsc.tab=0>