

Міністерство освіти і науки України
Луцький національний технічний університет



МАШИННЕ НАВЧАННЯ

Методичні вказівки до виконання самостійної роботи
для здобувачів
першого (бакалаврського) рівня вищої освіти
освітньої програми «Штучний інтелект та аналіз масивів даних»
спеціальності Прикладна математика
денної форми навчання

Луцьк 2025

УДК 004.85(076)

М10

Електронна копія друкованого видання передана для внесення в репозитарій ЛНТУ

Директор бібліотеки _____ Н. Поліщук

Рекомендовано до видання вченою радою факультету архітектури, будівництва та дизайну ЛНТУ,

протокол № ___ від «___» _____ 2025 року.

Голова вченої ради факультету ФАБД _____ О. Андрійчук

Розглянуто і схвалено на засіданні кафедри прикладної математики та механіки ЛНТУ,

протокол № ___ від «___» _____ 2025 року.

Завідувачка кафедри прикладної математики та механіки _____ О. Мікуліч

Укладач: _____ О. Приходько, кандидат технічних наук, доцент кафедри прикладної математики та механіки ЛНТУ

Рецензент: _____ доктор технічних наук, професор О. Мікуліч

Відповідальний за випуск: _____ О. Мікуліч, доктор технічних наук, професор, завідувачка кафедри прикладної математики та механіки ЛНТУ

Машинне навчання: методичні вказівки до виконання самостійної роботи для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Штучний інтелект та аналіз масивів даних» спеціальності «Прикладна математика» денної форми навчання / уклад. О. Приходько Луцьк: ЛНТУ, 2025. – 20 с.

Методичні вказівки містять розгорнуті рекомендації до виконання самостійної роботи з навчальної дисципліни «Машинне навчання». Видання побудовано у форматі «дорожньої карти», що включає інструкції до теоретичного вивчення матеріалу, опис практичних та індивідуальних завдань, перелік контрольних запитань і рекомендовану літературу.

Призначено для студентів денної форми навчання спеціальності «Прикладна математика» за освітньою програмою «Штучний інтелект та аналіз масивів даних»

Зміст

ВСТУП.....	4
Тема 1. Лінійні моделі та градієнтний алгоритм	5
Тема 2. Імовірнісний підхід у машинному навчанні	7
Тема 3. Метод опорних векторів, РСА та показники якості.....	9
Тема 4. Метричні методи та кластеризація	11
Тема 5. Регресійні дерева та ансамблі.....	13
ПІСЛЯМОВА ТА АВТОРСЬКА ПРИМІТКА.....	17
Список рекомендованої літератури.....	18

ВСТУП

Дисципліна «Машинне навчання» (Machine Learning) є ключовим етапом професійної підготовки бакалаврів за спеціальністю «Прикладна математика» (ОП «Штучний інтелект та аналіз масивів даних»). Цей курс, що викладається у 5-му семестрі, виконує роль моста між фундаментальними математичними дисциплінами, які ви вивчали перші два роки (математичний аналіз, лінійна алгебра, теорія ймовірностей), та спеціалізованими курсами, що чекають на вас попереду.

Важливість цього предмету є критичною. Це єдина дисципліна у вашому навчальному плані, яка системно охоплює «класичні» алгоритми машинного навчання (від лінійної регресії до градієнтного бустингу). В той час як наступний, 6-й семестр буде повністю присвячений нейронним мережам, саме тут ви повинні засвоїти універсальні принципи: що таке функція втрат, як працює регуляризація, чому виникає перенавчання (overfitting) і як правильно валідувати моделі. Без глибокого розуміння цих концепцій робота зі складнішими архітектурами Deep Learning перетвориться на «сліпе» перебирання параметрів. Крім того, на практиці – у банківському секторі, ритейлі чи промисловості – саме класичні методи (зокрема ансамблі дерев) часто виявляються ефективнішими для роботи з табличними даними, ніж нейромережі.

Ці методичні вказівки розроблені у форматі «дорожньої карти» (roadmap). Їх мета – не замінити підручники, а структурувати вашу самостійну роботу. Курс насичений математикою: ви будете виводити формули градієнтів, працювати з матричними розкладами та імовірнісними розподілами. Методичка допоможе вам розставити пріоритети: на чому зосередити увагу при читанні літератури, як пов'язати суху теорію з практичною реалізацією алгоритмів та які питання ставити собі для самоперевірки. Сприймайте цей документ як навігатор, що проведе вас від простих лінійних моделей до state-of-the-art алгоритмів бустингу, які сьогодні перемагають на змаганнях Kaggle.

Тема 1. Лінійні моделі та градієнтний алгоритм

Розпочніть роботу над темою з глибокого усвідомлення поняття **ознакового простору**. Ви повинні розуміти, що успіх моделі залежить не лише від алгоритму, а й від того, як ви представите дані. Розберіться, чому для лінійних моделей критично важливою є масштабованість ознак (StandardScaler, MinMaxScaler) і як саме геометрично змінюється поверхня функції втрат, якщо ознаки мають різний масштаб. Опрацюйте методи кодування категоріальних змінних (One-Hot Encoding, Label Encoding), звертаючи увагу на проблему мультиколінеарності, яку вони можуть спровокувати.

Наступним кроком є вивчення **аналітичного розв'язку лінійної регресії**. Ви маєте вміти вивести формулу нормального рівняння через мінімізацію суми квадратів помилок. Важливо не просто запам'ятати формулу, а проаналізувати її обчислювальну складність. Зверніть увагу на операцію обернення матриці та зробіть висновок, чому цей метод стає непридатним для великих наборів даних (Big Data), що логічно підведе вас до необхідності ітераційних методів.

Центральна частина вашої роботи – **регуляризація**. Ви повинні чітко розрізняти L1 (Lasso) та L2 (Ridge) регуляризатори. Не обмежуйтеся формулами. Спробуйте зрозуміти геометричну інтерпретацію: чому обмеження у вигляді ромба (L1) призводить до занулення ваг (відбору ознак), а обмеження у вигляді кола (L2) – лише до їх зменшення. Це фундаментальне знання для побудови стійких моделей.

Завершити тему слід оглядом сучасних **стохастичних алгоритмів оптимізації**. Якщо класичний градієнтний спуск (GD) та його стохастична версія (SGD) вам вже знайомі, то тут ви маєте розібратися з методами **SAG (Stochastic Average Gradient)** та **SAGA**. Зосередьтеся на математичних гарантіях їх збіжності. Зрозумійте, як саме ці алгоритми використовують пам'ять для збереження попередніх градієнтів, щоб досягти швидкості збіжності, як у повного градієнтного спуску, але з витратами, як у стохастичного.

Вказівки до виконання індивідуальних завдань

У практичній площині вам необхідно реалізувати (або детально розписати на папері) виведення градієнта для функції втрат MSE з L2-регуляризацією. Переконайтеся, що ви можете записати це у матричному вигляді.

Також проведіть чисельний експеримент (можна подумки або в коді): як зміниться розв'язок нормального рівняння, якщо матриця ознак X буде виродженою ($\text{determinant} = 0$), і як додавання регуляризатора вирішує цю проблему, роблячи матрицю оборотною.

Контрольні запитання

1. Чому перед навчанням лінійних моделей обов'язково потрібно проводити нормалізацію або стандартизацію даних?
2. У чому полягає основний недолік методу нормальних рівнянь при роботі з великою кількістю ознак?
3. Поясніть геометричну різницю між L1 та L2 регуляризацією. Чому L1 сприяє розрідженості моделі?
4. Що таке мультиколінеарність і як вона впливає на ваги лінійної моделі та їх інтерпретацію?
5. Як алгоритм SAGA покращує звичайний стохастичний градієнтний спуск?
6. Яка різниця між параметрами моделі та гіперпараметрами у контексті Ridge-регресії?
7. За яких умов гарантується збіжність алгоритму SAG до глобального мінімуму?

Рекомендована література: [1, 4, 6, 10, 13, 16]

Тема 2. Імовірнісний підхід у машинному навчанні

Перехід до цієї теми вимагає зміни парадигми мислення: від геометричної інтерпретації (проведення ліній) до імовірнісної (оцінка розподілів).

Розпочніть з **логістичної регресії**. Ваше завдання – зрозуміти, що це не просто «евристика з сигмоїдою», а суворо математично обґрунтований метод класифікації. Виведіть формулу логістичної регресії, виходячи з припущення, що розподіл класів підпорядковується розподілу Бернуллі. Розібрatися з поняттям «шансів» (odds) та «логарифму шансів» (log-odds/logits). Ви повинні вміти показати, як лінійна комбінація ознак перетворюється на ймовірність через максимізацію функції правдоподібності (Maximum Likelihood Estimation – MLE).

Далі заглибтеся у **повний байєсівський підхід**. Порівняйте його з класичним частотним підходом (frequentist). Вивчіть формулу Байєса в контексті навчання моделей. Чітко розмежуйте поняття *ап'юріорного розподілу* (ваші знання про параметри до бачення даних), *правдоподібності* (як дані описуються моделлю) та *апостеріорного розподілу* (оновлені знання). Зрозумійте різницю між оцінками MAP (Maximum A Posteriori) та MLE.

Наступний блок – **гауссовські моделі**. Пригадайте властивості багатовимірного нормального розподілу. Особливу увагу приділіть коваріаційній матриці: як вона визначає форму і орієнтацію "хмари" даних у просторі. Розгляньте квадратичний дискримінантний аналіз (QDA) та лінійний дискримінантний аналіз (LDA) як наслідки застосування формули Байєса до гауссовських даних.

Завершіть тему детальним розбором **Лінійного дискримінанта Фішера**. Тут важливо зрозуміти інший підхід до тієї ж задачі: ми не моделюємо розподіли, а шукаємо таку проекцію даних на пряму, щоб максимізувати відстань між середніми значеннями класів і одночасно мінімізувати дисперсію всередині кожного класу. Вивчіть поняття матриці розкиду (scatter matrix) – внутрішньокласової та міжкласової.

Вказівки до виконання індивідуальних завдань

Перше завдання – аналітичне. Спробуйте вручну вивести правило оновлення ваг для логістичної регресії, взявши похідну від функції Log-Loss. Ви побачите, що воно дуже схоже на правило для лінійної регресії, але з іншою гіпотезою.

Друге завдання – дослідницьке. Проаналізуйте, у чому полягає зв'язок між L2-регуляризацією в лінійній регресії та Байєсівським висновуванням з гауссовим апріорним розподілом ваг. (Підказка: знаходження моди апостеріорного розподілу еквівалентне мінімізації помилки з L2-штрафом).

Контрольні запитання

1. Чому для задачі класифікації метод найменших квадратів (MSE) є поганим вибором функції втрат?
2. Який імовірнісний зміст має вихід сигмоїдної функції в логістичній регресії?
3. Поясніть різницю між оцінками максимальної правдоподібності (MLE) та максимуму апостеріорної ймовірності (MAP).
4. Яку роль відіграє коваріаційна матриця у формулі багатовимірного нормального розподілу?
5. У чому полягає основна ідея критерію Фішера при побудові лінійного дискримінанта?
6. Чому LDA (Linear Discriminant Analysis) створює лінійні границі рішень, а QDA (Quadratic Discriminant Analysis) – квадратичні (криволінійні)?
7. Як пов'язані наївний байєсівський класифікатор та припущення про діагональність коваріаційної матриці?

Рекомендована література: [1, 4, 5, 6, 10]

Тема 3. Метод опорних векторів, РСА та показники якості

Ця тема поєднує в собі потужний метод класифікації та фундаментальні методи зниження розмірності. Спільною рисою тут є інтенсивне використання лінійної алгебри.

Розпочніть з **Методу опорних векторів (SVM)**. Ваше завдання – не просто зрозуміти ідею «найширшої смуги», що розділяє класи, а розібратися в математичній постановці задачі. Ви повинні вміти записати задачу квадратичної оптимізації з обмеженнями: мінімізація норми ваг (що максимізує відступ) за умови, що всі точки класифіковані правильно. Опрацюйте метод невизначених множників Лагранжа та перехід до **двоїстої задачі**. Саме тут ви зрозумієте, чому метод називається «опорних векторів» – лише невелика частина точок (ті, що лежать на межі) впливають на розв'язок, решта – ігноруються.

Наступний критичний крок – **Теорія ядер (Kernel Trick)**. Це математичний трюк, який дозволяє будувати нелінійні розділяючі поверхні, залишаючись в рамках лінійного алгоритму. Розберіться, як скалярний добуток у просторі високої розмірності замінюється ядерною функцією у вихідному просторі. Ознайомтеся з теоремою Мерсера, яка визначає умови, за яких функція може бути ядром (невід'ємна визначеність матриці Грама).

Далі перейдіть до навчання без учителя – **Методу головних компонент (PCA)**. Не сприймайте його як "чорну скриньку" з sklearn. Вивчіть два підходи до його виведення: як максимізацію дисперсії проекції даних та як мінімізацію середньоквадратичної помилки реконструкції. Основою PCA є **Сингулярний розклад матриці (SVD)**. Ви повинні чітко розуміти структуру розкладу $X = U\Sigma V^T$ і фізичний зміст кожної матриці. Зрозумійте, як власні числа коваріаційної матриці пов'язані з "важливістю" компонент.

Завершіть тему вивченням **метрик якості ранжування**. Відрізняйте задачі класифікації від задач Information Retrieval.

- *AUC-ROC*: зрозумійте її імовірнісну інтерпретацію (ймовірність того, що випадковий позитивний об'єкт отримає вищу оцінку, ніж випадковий негативний).
- *MAP (Mean Average Precision) та NDCG*: розберіть, як ці метрики враховують порядок видачі (помилка на першій позиції "коштує" дорожче, ніж на десятій).

Вказівки до виконання індивідуальних завдань

Перше завдання – теоретичне. Запишіть функцію Лагранжа для "м'якого" SVM (Soft Margin), де допускаються помилки класифікації через змінні (slack variables). Поясніть роль гіперпараметра у балансі між шириною зазору та помилками на навчальній вибірці. Друге завдання – обчислювальне. Для невеликої матриці даних (наприклад, 3x2) вручну проведіть процедуру SVD-розкладу або хоча б знайдіть власні вектори матриці, щоб отримати головні компоненти. Це допоможе демістифікувати PCA.

Контрольні запитання

1. У чому полягає основна ідея переходу до двоїстої задачі в SVM?
2. Які вектори називаються опорними і що станеться з розділяючою гіперплощиною, якщо видалити всі неопорні вектори?
3. Сформулюйте сутність Kernel Trick: як ми обчислюємо скалярний добуток у просторі спрямлення без явного перетворення даних?
4. Яка умова Мерсера гарантує існування простору, в якому ядро є скалярним добутком?
5. Чому перед застосуванням PCA дані необхідно центрувати?
6. Як інтерпретувати власні числа коваріаційної матриці в контексті втрати інформації при зниженні розмірності?
7. У чому перевага метрики NDCG перед звичайною точністю (Precision) для оцінки пошукових систем?

Рекомендована література: [1, 4, 6, 9, 13, 15]

Тема 4. Метричні методи та кластеризація

В основі цієї теми лежить просте, але фундаментальне поняття: *схожі об'єкти знаходяться близько один до одного*. Ваше завдання – зрозуміти, як математично формалізувати "близькість" та "групування".

Розпочніть з **метрик відстаней**. Евклідова відстань не є універсальною. Детально вивчіть **відстань Махаланобіса**, яка враховує кореляції між ознаками та різницю в дисперсіях (по суті, це евклідова відстань у просторі головних компонент, нормованих на дисперсію). Розгляньте **косинусну відстань** і зрозумійте, чому в задачах обробки тексту (Text Mining) кут між векторами важливіший за їх довжину. Згадайте відстань Мінковського як узагальнення.

Переходячи до **кластеризації**, порівняйте жорсткий та імовірнісний підходи.

Основну увагу приділіть **GMM (Gaussian Mixture Models)**. Це узагальнення методу k-means. Якщо k-means описує кластери сферами, то GMM – еліпсоїдами. Ключовим тут є **ЕМ-алгоритм (Expectation-Maximization)**. Розберіть його покроково:

1. *E-step*: оцінюємо ймовірність приналежності точки до кластера.
2. *M-step*: оновлюємо параметри кластерів (середнє та коваріацію) на основі цих ймовірностей. Зрозумійте, що GMM – це метод оцінки щільності розподілу, а не просто розбиття точок.

Далі опрацюйте **ієрархічну кластеризацію**. Зосередьтеся на методах об'єднання (Linkage). Ви маєте чітко розуміти різницю між *Single Linkage* (відстань між найближчими точками) та *Complete Linkage* (між найдальшими). Проаналізуйте так званий "ланцюговий ефект" (chaining effect) у Single Linkage, коли кластери витягуються в довгі лінії.

Завершіть тему методами **оцінювання якості**. Оскільки в кластеризації зазвичай немає правильних відповідей, вивчіть внутрішні метрики:

- *Силует (Silhouette Coefficient)*: показує, наскільки об'єкт схожий на свій кластер порівняно з сусідніми.

- *Індекс Девіса-Болдіна*: відношення розкиду всередині кластерів до відстані між ними.

Вказівки до виконання індивідуальних завдань

Перше завдання – візуально-аналітичне. Намалюйте ситуацію, де Евклідова відстань дасть неправильний результат визначення найближчого сусіда, а відстань Махаланобіса – правильний (наприклад, сильно витягнута хмара точок).

Друге завдання – алгоритмічне. Опишіть псевдокод агломеративної ієрархічної кластеризації. Поясніть, як за допомогою дендрограми вибрати оптимальну кількість кластерів.

Контрольні запитання

1. Чому при збільшенні розмірності простору (Curse of Dimensionality) Евклідова відстань втрачає свою інформативність?
2. У чому полягає основна відмінність між відстанню Махаланобіса та Евклідовою відстанню?
3. Поясніть сутність EM-алгоритму для навчання моделі суміші гауссіан (GMM).
4. Яку перевагу надає «м'яка» кластеризація (Soft Clustering) порівняно з «жорсткою» (Hard Clustering)?
5. Як вибір методу зв'язування (Linkage) в ієрархічній кластеризації впливає на форму утворених кластерів?
6. Що показує силуетний коефіцієнт: як інтерпретувати значення близькі до 1, 0 та -1?
7. Чому k-means можна вважати окремим (спрощеним) випадком GMM?

Рекомендована література: [1, 4, 6, 13, 14]

Тема 5. Регресійні дерева та ансамблі

У цій темі ви маєте розібратися з алгоритмами, що не базуються на лінійній алгебрі та диференціюванні, а використовують логіку рекурсивного розбиття простору. Дерева рішень – це фундамент для найсучасніших методів.

Розпочніть з математики **критеріїв розбиття**. Дерево жадібно шукає поріг, який найкраще розділяє дані.

Для класифікації: вивчіть поняття **Ентропії Шеннона** (міра хаосу) та **Неоднорідності Джині (Gini Impurity)**. Ви повинні розуміти, що таке **Information Gain** (приріст інформації) – різниця між ентропією батьківського вузла та зваженою ентропією дітей.

Для регресії: розберіть, як **MSE (Mean Squared Error)** використовується як критерій, де передбаченням у листі є середнє значення об'єктів, що туди потрапили.

Далі зосередьтеся на головній проблемі дерев – **перенавчанні**. Дерево може нескінченно ділити простір, поки в кожному листі не залишиться по одному об'єкту. Вивчіть методи регуляризації через обмеження росту (**pre-pruning**): **max_depth** (максимальна глибина), **min_samples_split** (мінімум об'єктів для розбиття), **min_samples_leaf**. Зрозумійте, як кожен з цих гіперпараметрів впливає на складність моделі.

Окремий блок – робота з даними. Зверніть увагу, що дерева, на відміну від лінійних моделей, теоретично не потребують **One-Hot Encoding** для категоріальних ознак (хоча реалізація в **sklearn** має обмеження) і не вимагають масштабування даних. Вивчіть концепцію **сурогатних розбиттів (surrogate splits)** для обробки пропусків у даних.

Перехід до ансамблів розпочніть з методу **Беггінг (Bagging – Bootstrap Aggregating)**. Ключовим тут є поняття **бутстрепу** – вибірки з поверненням. Ви маєте статистично обґрунтувати, чому усереднення відповідей багатьох незалежних моделей зменшує дисперсію (**Variance**), але не змінює зсув (**Bias**). Це пояснює, чому **Random Forest** будується з глибоких (перенавчених) дерев. Також

розберіть метод випадкових підпросторів (Random Subspace Method), який використовується в лісі для декореляції дерев.

Вказівки до виконання індивідуальних завдань

Перше завдання – розрахункове. Вручну розрахуйте ентропію та індекс Джині для кореня невеликого дерева (наприклад, 10 об'єктів: 6 класу "А" і 4 класу "Б"). Порахуйте Information Gain для одного можливого розбиття.

Друге завдання – аналітичне. Поясніть, чому Random Forest не перенавчається зі збільшенням кількості дерев, на відміну від бустингу. Опишіть механізм оцінки якості Out-of-Bag (OOB) error, який дозволяє обійтися без окремої валідаційної вибірки.

Контрольні запитання

1. У чому полягає геометричний зміст побудови дерева рішень (як воно ділить n -вимірний простір)?
2. Яка принципова різниця між ентропією та індексом Джині? Що обчислюється швидше?
3. Чому глибокі дерева рішень мають низький зсув (bias), але високу дисперсію (variance)?
4. Що таке бутстреп-вибірка і яка приблизно частка унікальних об'єктів потрапляє в неї з початкового датасету?
5. Чому в Random Forest ознаки для розбиття вибираються випадковим чином, а не перебираються всі можливі?
6. Як дерева рішень можуть обробляти пропущені значення без видалення об'єктів?
7. Чому дерева рішень не потребують нормалізації вхідних ознак?

Рекомендована література: [1, 6, 9, 13, 18, 19]

Тема 6. Бустинг

Це найскладніша тема курсу, яка вимагає розуміння оптимізації у функціональному просторі. На відміну від бегтінгу, тут моделі будуються послідовно, виправляючи помилки попередників.

Почніть з класичного алгоритму **AdaBoost**. Розберіть його аналітичне виведення через мінімізацію **експоненційної функції втрат**. Зрозумійте механізм перезважування об'єктів: на кожній ітерації вага об'єктів, на яких помилився попередній алгоритм, збільшується. Це змушує нову слабку модель (зазвичай "пеньок" – decision stump) фокусуватися на складних прикладах.

Далі переходьте до узагальнення – **Градiєнтного бустингу (GBM)**. Це ключовий момент: ви повинні зрозуміти, що бустинг – це, по суті, **градієнтний спуск**, де ми оновлюємо не ваги коефіцієнтів, а *значення функції передбачення*. Зрозумійте, чому ми навчаємо кожне наступне дерево прогнозувати **антиградієнт функції втрат** (для MSE це просто залишки/residuals).

Вивчіть поняття **стохастичного бустингу**. За аналогією з Random Forest, використання підмножин рядків та стовпців (subsample, colsample_bytree) при побудові кожного дерева не лише пришвидшує навчання, а й покращує здатність до узагальнення.

Завершіть тему порівняльним аналізом сучасних бібліотек, які є стандартом індустрії. Ви маєте знати їхні "фішки":

- **XGBoost**: регуляризація в цільовій функції (L1/L2 для листя), наближений алгоритм пошуку розбиттів (для швидкості).
- **LightGBM**: навчання на гiстограмах (значне прискорення), техніка GOSS (Gradient-based One-Side Sampling – ігнорування об'єктів з малим градієнтом/помилкою) та EFB. Росте "вглиб" (leaf-wise), а не "вшир" (level-wise).
- **CatBoost**: робота з категоріальними ознаками "з коробки" (Target Encoding з зсувом, щоб уникнути витоку даних), симетричні дерева (менше схильні до перенавчання), упорядкований бустинг (Ordered Boosting).

Вказівки до виконання індивідуальних завдань

Перше завдання – теоретичне доведення. Покажіть, що для функції втрат MSE антиградієнт точно дорівнює залишку (різниці між фактом і прогнозом). Це пояснює, чому в найпростішому бустингу ми вчимо дерева передбачати помилку.

Друге завдання – порівняльне. Складіть таблицю відмінностей між Random Forest та Gradient Boosting за критеріями: спосіб побудови (паралельний/послідовний), боротьба з компонентами помилки (Variance/Bias), глибина базових дерев (глибокі/мілкі), чутливість до шуму.

Контрольні запитання

1. У чому полягає фундаментальна відмінність стратегії побудови ансамблю в бустингу порівняно з бегінгом?
2. Як AdaBoost використовує експоненційну функцію втрат для визначення ваги кожного нового дерева в ансамблі?
3. Чому градієнтний бустинг називають "градієнтним спуском у функціональному просторі"?
4. Яку роль відіграє параметр learning rate (shrinkage) у бустингу і як він пов'язаний з кількістю дерев?
5. Чому LightGBM працює швидше за класичний XGBoost на великих даних?
6. У чому особливість обробки категоріальних ознак у CatBoost порівняно з One-Hot Encoding?
7. Чому бустинг більш схильний до перенавчання, ніж випадковий ліс, і як із цим боротися?

Рекомендована література: [1, 6, 7, 13, 19, 20]

ПІСЛЯМОВА ТА АВТОРСЬКА ПРИМІТКА

На завершення варто зазначити важливий нюанс щодо створення цього навчального посібника. Дана методична розробка є результатом співпраці викладача та сучасних технологій штучного інтелекту.

Текст методичних вказівок був згенерований великою мовною моделлю (LLM), після чого пройшов ретельну верифікацію, редагування та авторську вичитку. Це символічний момент: ви починаєте вивчати класичне машинне навчання, використовуючи матеріали, створені за допомогою технологій, що стали вершиною розвитку цієї галузі (Deep Learning та трансформери).

Той факт, що алгоритм здатен не лише вирішувати обчислювальні задачі, а й допомагати у структуруванні навчального контенту, найкраще демонструє потенціал вашої майбутньої професії. Однак пам'ятайте: ШІ — це потужний інструмент, але відповідальність за верифікацію знань та прийняття рішень завжди лежить на людині-фахівці. Сподіваємось, цей приклад надихне вас на глибше вивчення математичних основ, які роблять подібні технології можливими.

Список рекомендованої літератури

1. Машинне навчання : навчальний посібник / Т. М. Басюк, В. В. Литвин, Л. М. Захарія, Н. Е. Кунанець ; за наук. ред. д-ра техн. наук, проф. В. В. Пасічника. – 3-тє вид., стер. – Львів : Новий Світ–2000, 2025. – 330 с. – ISBN 978-617-7519-42-2.
2. Литвин В. В. Глибинне навчання : навчальний посібник / В. В. Литвин, Р. М. Пелешак, В. А. Висоцька ; М-во освіти і науки України, Нац. ун-т «Львівська політехніка». – Львів : Вид-во Львівської політехніки, 2021. – 264 с. : іл.
3. Басюк Т. М. Машинне навчання та аналіз даних (розділи з оптимізації, класифікації, регресії) // у складі навчального посібника [1].
4. Bishop C. M. Pattern Recognition and Machine Learning. – New York : Springer, 2006. – 738 p.
5. Murphy K. P. Machine Learning: A Probabilistic Perspective. – Cambridge, MA : MIT Press, 2012. – 1104 p.
6. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. – 2nd ed. – New York : Springer, 2009. – 745p.
7. Rao C. R., Rao A. S. R. S., Govindaraju V. Deep Learning. – Amsterdam : Elsevier, 2023. – 512 p.
8. Bishop C. M., Bishop H. Deep Learning. – Cham : Springer International Publishing, 2024. – 656 p. – DOI: <https://doi.org/10.1007/978-3-031-45468-4>
9. Lazzeri F. Machine Learning for Time Series Forecasting with Python. – Hoboken : Wiley, 2021. – 250 p.
10. Goodfellow I., Bengio Y., Courville A. Deep Learning : electronic book. – 2016. – Режим доступу: <https://www.deeplearningbook.org/>
11. Jentzen A., Kuckuck B., von Wurstemberger P. Mathematical Introduction to Deep Learning: Methods, Implementations, and Theory. – 2023. – Режим доступу: <https://arxiv.org/abs/2310.20360>
12. Petersen P., Zech J. Mathematical Theory of Deep Learning. – 2024. – Режим доступу: <https://arxiv.org/abs/2407.18384>
13. Dive into Deep Learning : online book. – Режим доступу: <https://d2l.ai/>

- 14.Крупський, О. В., Приходько, О. С. Розробка та валідація агент-орієнтованої симуляції з простим штучним інтелектом для моделювання еволюційної адаптації. Наукові нотатки, (81), 50–54 / URL: <https://doi.org/10.36910/775.24153966.2025.81.7>
- 15.Крупський О. В., Приходько О. С. Дослідження еволюційної динаміки генетичних стратегій в умовах змінної доступності ресурсів // Міжвузівський збірник «Наукові нотатки». – Луцьк, 2025. – № 82. – С. 88–93 / URL: <https://doi.org/10.36910/775.24153966.2025.82.15>
- 16.Ющук І. С., Приходько О. С. Аналіз обмежень машинного навчання у прогнозуванні спортивних результатів // Міжвузівський збірник «Наукові нотатки». – Луцьк, 2025. – № 82. – С. 107–111 / URL: <https://doi.org/10.36910/775.24153966.2025.82.17>
- 17.Ягелюк О. О., Приходько О. С. Методика прогнозування врожайності льону-волокна та насіння за допомогою БПЛА-зйомки та автоматизованого аналізу // Міжвузівський збірник «Наукові нотатки». – Луцьк, 2025. – № 83. – С. 166–171 / URL: <https://doi.org/10.36910/775.24153966.2025.83.26>
- 18.Приходько, О.С., Приходько, В.О. (2025). Ітераційна оптимізація складу бетону із заданою швидкістю тверднення за допомогою машинного навчання. Сучасні технології та методи розрахунків у будівництві, 24, 331–339 / URL: [https://doi.org/10.36910/6775-2410-6208-2025-14\(24\)-27](https://doi.org/10.36910/6775-2410-6208-2025-14(24)-27)
- 19.Печончик Н.Р., Приходько О. С. Порівняльний аналіз моделей машинного навчання для прогнозування міцності бетону// Міжвузівський збірник «Наукові нотатки». – Луцьк, 2025. – № 84. – С. 273–278 / URL: <https://doi.org/10.36910/775.24153966.2025.84.26>
- 20.Закревський В.Л., Приходько О. С. Розробка адаптивного методу керування БПЛА на основі глибокого навчання з підкріпленням та доменної рандомізації// Міжвузівський збірник «Наукові нотатки». – Луцьк, 2025. – № 84. – С. 259–264 / URL: <https://doi.org/10.36910/775.24153966.2025.84.26>

Машинне навчання: методичні вказівки до виконання самостійної роботи для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Штучний інтелект та аналіз масивів даних» спеціальності «Прикладна математика» денної форми навчання / уклад. О. Приходько Луцьк: ЛНТУ, 2025. – 20 с.

Комп'ютерний набір

О. Приходько

Редактор

О. Приходько

Підп. до друку « ____ » _____ 2025 р. Формат 60x84/16. Папір офс.

Гарн. Таймс. Ум. друк. арк. 2,25.

Тираж 50 прим.

Інформаційно-видавничий відділ
Луцького національного технічного університету
43018, м. Луцьк, вул. Львівська, 75
Друк – ІВВ ЛНТУ