НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

МЕТОДИЧНІ ВКАЗІВКИ
по виконанню лабораторних робіт з дисципліни
ПРОГРАМУВАННЯ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

для студентів спеціальності
121 «Інженерія програмного забезпечення»
другого (магістерського) рівня
освітня програма
«Програмне забезпечення інформаційних систем»

Київ - 2024

УДК 621.391(075.8)

Методичні вказівки по виконанню лабораторних робіт з дисципліни Програмування систем штучного інтелекту для студентів спеціальності 121 «Інженерія програмного забезпечення» освітньо-професійної програми «Програмне забезпечення інформаційних систем»

Рекомендовано вченою радою факультету інформаційних технологій НУБіП України, протокол № від жовтня 2024 р.

Укладачі: професор Р.А. Руденський,
доцент Г.О. Вайганг

Рецензенти: к.т.н., доц. Б.Л. Голуб,
к.т.н., доцент кафедри інтелектуальних технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка О.Є. Іларіонов

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ
по виконанню лабораторних робіт з дисципліни
ПРОГРАМУВАННЯ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

для студентів спеціальностей
121 «Інженерія програмного забезпечення»
другого (магістерського) рівня
освітня програма
«Програмне забезпечення інформаційних систем»

Укладачі: професор Р.А. Руденський, доцент Г.О. Вайганг

Зміст

ВСТУП

Методи штучного інтелекту і зокрема машинне навчання знаходить все більше застосувань у сільському господарстві, дозволяючи підвищити продуктивність та ефективність різноманітних процесів. Завдяки своїм особливостям, таким як здатність обробляти великі масиви даних і виявляти складні закономірності, машинне навчання надає фермерам і аграрним компаніям нові інструменти для оптимізації виробництва.

У рослинництві машинне навчання може бути використане для визначення оптимальних умов вирощування рослин, аналізу стану ґрунту та визначення необхідних добрив і пестицидів. Це допомагає зменшити витрати та підвищити врожайність за рахунок точкових обробок і прийняття обґрунтованих рішень. Також можливе застосування машинного навчання для виявлення захворювань рослин та вибору найбільш ефективних методів боротьби з ними, що дозволяє зменшити втрачену продукцію та покращити загальну якість врожаю.

У тваринництві та птахівництві машинне навчання стає корисним для аналізу поведінки тварин і визначення їхнього здоров’я. Наприклад, системи машинного навчання можуть використовуватися для моніторингу харчування та активності тварин, а також для виявлення захворювань і прогнозування можливих проблем. Це сприяє покращенню умов утримання та ефективності виробництва, а також знижує ризики.

Загалом, перехід до методів машинного навчання може суттєво допомогти сільському господарству підвищити продуктивність та ефективність різних процесів, що приведе до покращення якості та кількості продукції, роблячи аграрну галузь більш стійкою до викликів сучасності.

ТЕОРЕТИЧНА ЧАСТИНА

Основи штучного інтелекту та машинного навчання

Методи штучного інтелекту включають методи, які дозволяють комп'ютерам імітувати людську поведінку, даючи їм змогу навчатися, приймати рішення, розпізнавати закономірності та вирішувати складні проблеми так, як це робить людський інтелект.

Машинне навчання – це підгрупа методів штучного інтелекту, що використовує вдосконалені алгоритми для виявлення закономірностей у великих наборах даних, дозволяючи машинам навчатися та адаптуватися. Таким чином у вузькому сенсі методи машинного навчання можна визначити як алгоритми та моделі, які використовуються для обробки та аналізу великих обсягів даних. Вони дозволяють комп'ютерним системам витягувати знання з даних і використовувати їх для прийняття рішень і формування прогнозів.

Іншими словами машинне навчання – це підхід до (1) вивчення (2) складних закономірностей на основі (3) наявних даних і використання цих закономірностей для (4) передбачення невідомих даних.

Рис. 1 – Структура задач машинного навчання для агропромислової сфери

Наприклад, для задачі прогнозування врожайності пшениці відповідні блоки будуть спрямовані на вирішення наступних задач:

1. **Алгоритм** — встановлює зв'язки або вплив різних факторів, таких як кількість опадів, температура, використання добрив і технології обробки ґрунту, тощо на врожайність пшениці

2. **Дані** — історичні спостереження за врожайністю пшениці, погодні умови кожного сезону, використання добрив, процедури обробки ґрунту, попередні культури і інші агротехнічні заходи

3. **Модель** — сталий вплив на врожайність певних діапазонів температур та кількості опадів, впливи пов’язані із взаємодією різними фаторів, як-то видами добрив та типами ґрунту, попередні культури, тощо

4. **Прогнози** — очікувана врожайність в наступному сезоні на певному грунті, якщо погодні умови та агротехнічні заходи будуть подібними до одного з попередніх сезонів.

Типи машинного навчання

Методи машинного навчання можна класифікувати на три основні категорії (рис. 2): навчання із учителем, навчання без учителя та навчання з підкріпленням. Навчання із учителем передбачає використання пар “вхідні дані – вихідні дані”, де модель навчається на основі розмічених даних, що дозволяє їй прогнозувати результати для нових спостережень.

Навчання без учителя, на відміну від цього, працює з не розміченими даними, де алгоритми самостійно виявляють закономірності та структури в даних, наприклад, через кластеризацію або зменшення розмірності.

Нарешті, навчання з підкріпленням зосереджене на навчанні агента через взаємодію з середовищем; агент виконує дії в певних станах і отримує винагороди або штрафи, що стимулює його до максимізації загальної нагороди через оптимізацію своїх дій.

Навчання без учителя – це метод, при якому система навчається на основі не розмічених даних. Мета полягає у виявленні закономірностей та структури в даних без будь-яких попередньо заданих умов. Приклади таких методів включають кластеризацію та зменшення розмірності.

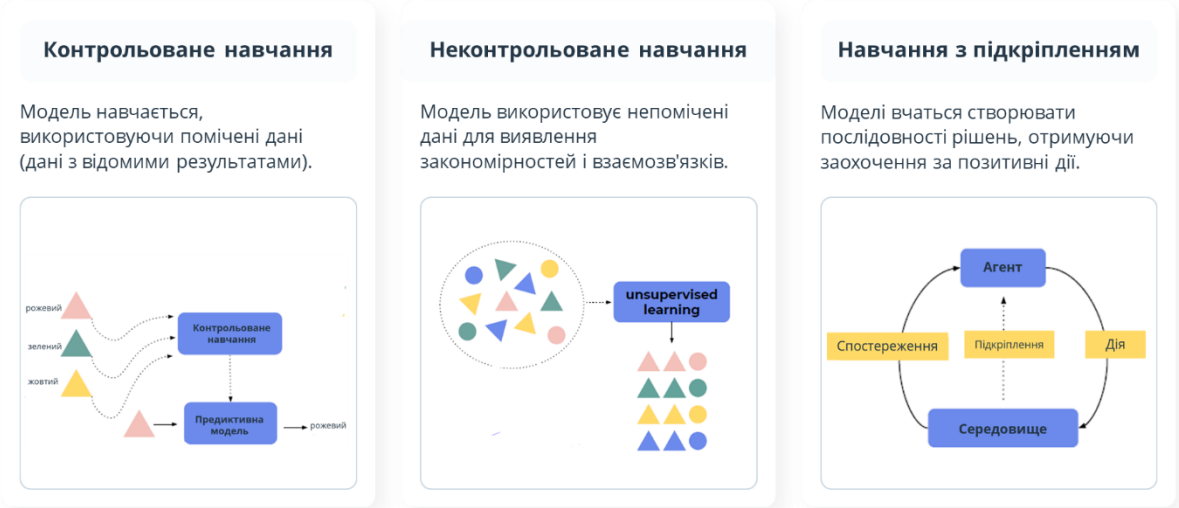


Рис. 2 – Категорії машинного навчання

Кластеризація – це метод, коли модель розділяє дані на групи (кластери) на основі їхньої схожості. Основними алгоритмами, що використовуються для кластеризації, є k-середніх і DBSCAN. Сниження розмірності, в свою чергу, дозволяє зменшити кількість ознак у даних, зберігаючи при цьому основну інформацію. Методи навчання без учителя можуть застосовуватися в різних сферах, таких як аналіз даних, пошук аномалій, стиснення даних та інші.

Рис. 3 – Методи навчання без учителя

Навчання з учителем – це метод, при якому система навчається на основі пар «вхідні дані – вихідні дані». Це означає, що модель отримує вхідні дані разом із правильними відповідями, що дозволяє їй навчитися передбачати правильні

результати для нових вхідних даних. Наприклад, у випадку прогнозування урожайності сільськогосподарських культур, система може використовувати дані про погоду, тип ґрунту та рівень добрив, щоб навчитися передбачати ймовірний урожай. Основні етапи процесу навчання з учителем включають підготовку даних, вибір моделі, навчання, оцінку моделі та використання для прогнозів. Якість моделі залежить від якості даних, на яких вона навчається, а також від правильного вибору моделі і параметрів.

Навчання з учителем в свою чергу ділиться на дві підгрупи: моделі (або алгоритми) регресії та моделі (або алгоритми) класифікації.

Регресія – це, по суті, спосіб передбачення майбутніх числових значень на основі минулих ознак. Вона допомагає прогнозувати такі показники, як продажі, ціни, врожайність, робити рекомендації. Основні методи регресії наведено на рисунку 4.

Рис. 4 – Базові методи регресії

Методи класифікації на відміну допомагають передбачити, до якої категорії чи групи належить кожне спостереження. Ці методи можуть допомогти визначити, чи є певні типи хвороб у рослин чи тварин, чи електронний лист спамом, або передбачити, чи має банк схвалити кредитну заявку.

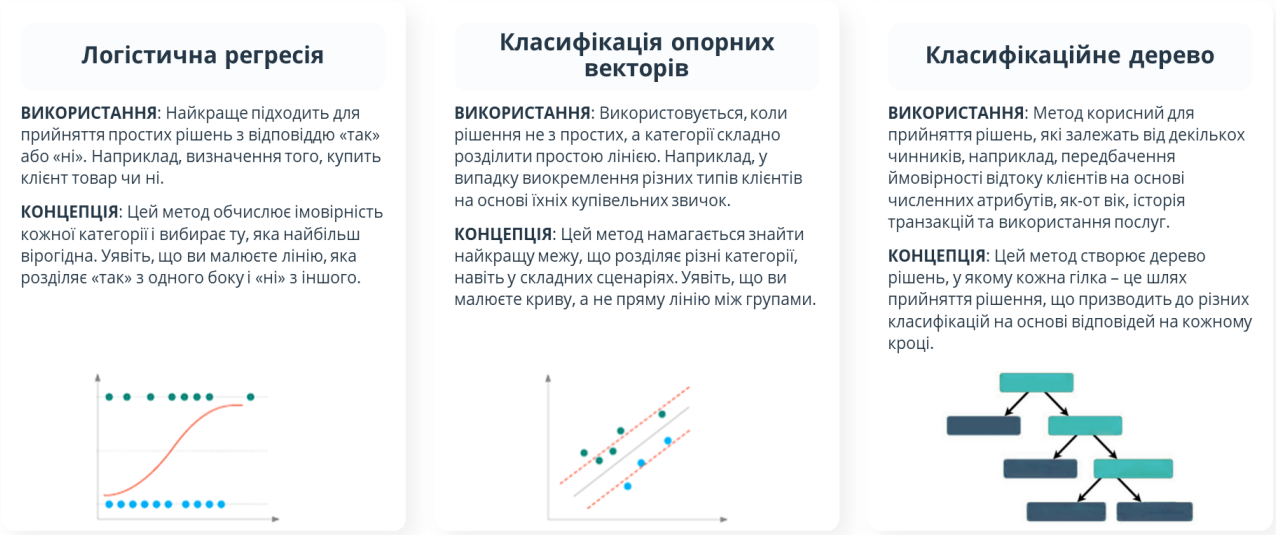


Рис. 5 – Базові методи класифікації

Для складних задач у машинному навчанні часто є недостатньо лише однієї моделі, оскільки одна модель може не повністю впоратися з усіма аспектами або нюансами досліджуваної проблеми. У таких випадках використовують ансамблеві методи, які об'єднують кілька моделей для покращення загальної точності та надійності прогнозів.

Основні типи ансамблевих методів включають методи узгодження (bagging), такі як Random Forest, які створюють численні моделі з випадкових підвибірок навчальних даних і комбінують їхні результати для зменшення варіативності та підвищення стабільності. Іншим підходом є методи підсилювання (boosting), як-от AdaBoost або Gradient Boosting, які навчають моделі послідовно, зосереджуючи увагу на прикладах, які були класифіковані неправильно попередніми моделями, щоб покращити загальні результати.

Ці ансамблеві техніки часто призводять до значного збільшення продуктивності ансамбля моделей, оскільки окремі моделі мають свої сильні сторони та дозволяють компенсувати слабкості інших.

Рис. 6 – Ансамблеві методи

Навчання з підкріпленням — цей тип навчання складається з декількох етапів: визначення задачі, визначення станів і дій, функції нагороди, навчання моделі, оцінки моделі та використання. Основними методами навчання з підкріпленням є Q-навчання та методи глибокого навчання, такі як Deep Q-Networks (DQN) і Policy Gradient. Q-навчання розраховує функцію Q(s,a), яка визначає очікувану винагороду за виконання певних дії за різних станах

Глибоке навчання

Окремою галуззю машинного навчання є *Глибоке навчання.* Воно базується на використанні багатошарових нейронних мереж для автоматичного навчання і виявлення складних патернів у великих обсягах даних. Перехід від класичного машинного навчення до глибокого навчання зазвичай відбувається, коли розмір і складність даних перевищують можливості традиційних алгоритмів, а також коли є необхідність працювати з неструктурованими даними, такими як зображення, аудіо або текст. Основна відмінність глибокого навчання від класичних підходів полягає в здатності нейронних мереж автоматично витягувати ознаки з сирих даних без ручного відбору.

Глибокі моделі здатні вчитися на різних рівнях абстракції; вони можуть виявляти прості ознаки на нижчих шарах, а складніші патерни на верхніх.

Глибоке навчання застосовується в широкому спектрі задач, таких як комп'ютерний зір, обробка природної мови та генерація контенту, вимагаючи значних обчислювальних потужностей і великих обсягів даних для навчання.

Рис. 7 – Базовий принцип роботи глибоких нейронних мереж

Основи управління даними в системах Штучного Інтелекту

Управління даними в системах штучного інтелекту є критично важливим етапом, оскільки якість та релевантність даних прямо впливають на ефективність і точність моделей штучного інтелекту. Основи управління даними включають кілька ключових етапів:

1. *Збір даних* – цей етап передбачає збирання даних з різноманітних джерел, які можуть включати бази даних, API, анкети, датчики та інше. Якість зібраних даних визначає успішність подальших етапів.

У контексті сільськогосподарських технологій збір даних може включати в себе такі методи та джерела, як Супутникові зображення, Дані від сенсорів на сільськогосподарській техніці, Метеодані, Текстові дані, Табличні дані

2. *Розмітка даних* – на цьому етапі дані отримують мітки, які описують їхній зміст. Це може бути виконане вручну експертами або автоматизовано за допомогою алгоритмів, залежно від задачі. Правильне маркування є критично

важливим для навчання моделей, оскільки воно дозволяє системам навчатися на основі зрозумілих прикладів. З точки зору моделі машинного навчання Розмітка даних - це процес ідентифікації у вхідних даних одного чи кількох цільових признаків, значення яких має прогнозувати модель. Це дозволяє моделі МН вчитися на даних і розпізнавати закономірності. Для керуючих моделей навчання мітки є важливими, оскільки вони діють як прямі індикатори бажаного виходу.

3. *Розвідувальний аналіз даних* (Exploratory Data Analysis, EDA) – цей етап включає використання статистичних і візуалізаційних методів для дослідження зібраних даних. EDA допомагає виявити закономірності, аномалії та інші особливості даних, що можуть вплинути на подальший аналіз і розробку моделей.

Отримані в результаті узагальнення визначають наступні кроки підготовки даних та моделювання, забезпечуючи високу якість та глибоке розуміння даних, які подаються на вхід моделям машинного навчання.

Рис. 8 – Ключові етапи управління даними у системах штучного інтелекту

4. Підготовка даних – на цьому етапі дані очищуються і трансформуються для подальшої обробки. Це може включати видалення пропущених значень, нормалізацію, кодування категоріальних змінних і масштабування числових даних. Якісна підготовка даних забезпечує, що моделі отримують коректну і чітко структуровану інформацію, що підвищує ймовірність успішного навчання.

Попередня обробка є важливою для перетворення сирої інформації в чистий, організований формат, який підходить для побудови надійних моделей

машинного навчання. Це включає не лише очищення і перетворення даних, але також зменшення варіативності і доповнення їх для покращення результатів навчання моделі. Типові техніки попередньої обробки даних наведені на рисунку 9.

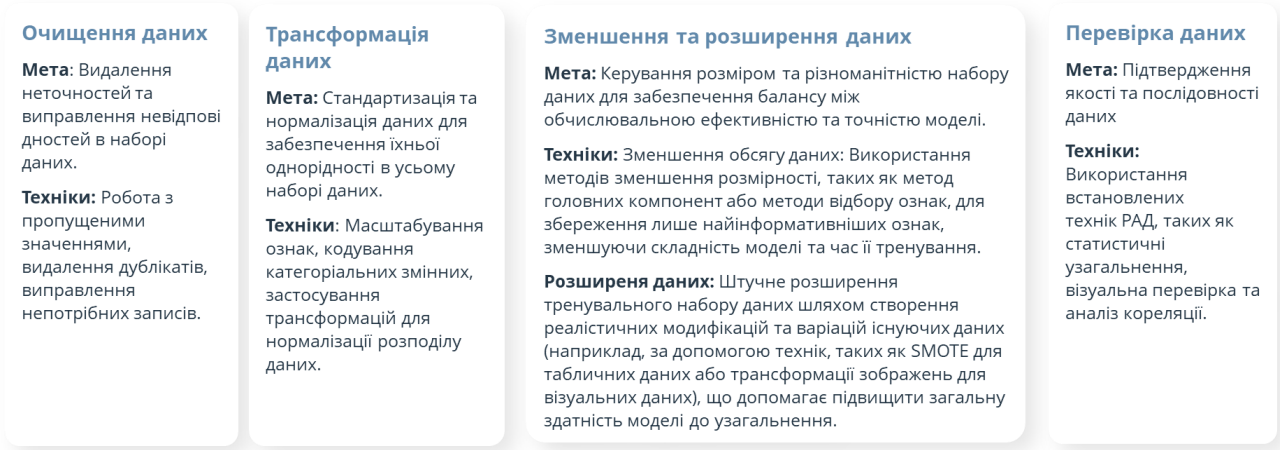


Рис. 9 – Техніки попередньої обробки даних

Таблиця 1 – Методи трансформації даних

Назва методу	Формула	Результат	Умови використання
StandardScaler	$X^{std} = \frac{(X - \bar{X})}{\sigma_x}$	Середнє значення 0, стандартне відхилення 1	Нормальний закон розподілу
MinMaxScaler	$X_{minmax} = \frac{X - X_{min}}{X_{max} - X_{min}}$	Масштабовані значення знаходяться в межах бажаного діапазону, зазвичай [0, 1]	Необхідність забезпечити певні межі діапазону змінних та зберегти форму розподілу (використовується майже завжди)
RobustScaler	$X_{roβust} = \frac{X - median(X)}{Q_3 - Q_1}$	Масштабування засновано на значеннях міжквартильного діапазону (IQR). Викиди мають менший вплив	Використовується, коли є певні викиди в даних
MaxAbsScaler	$X_{maxaβs} = \frac{X}{X_{aβsmax}}$	Масштабовані значення знаходяться в межах діапазону [-1, 1]	Використовується, коли дані мають середнє значення близько 0, розріджені дані

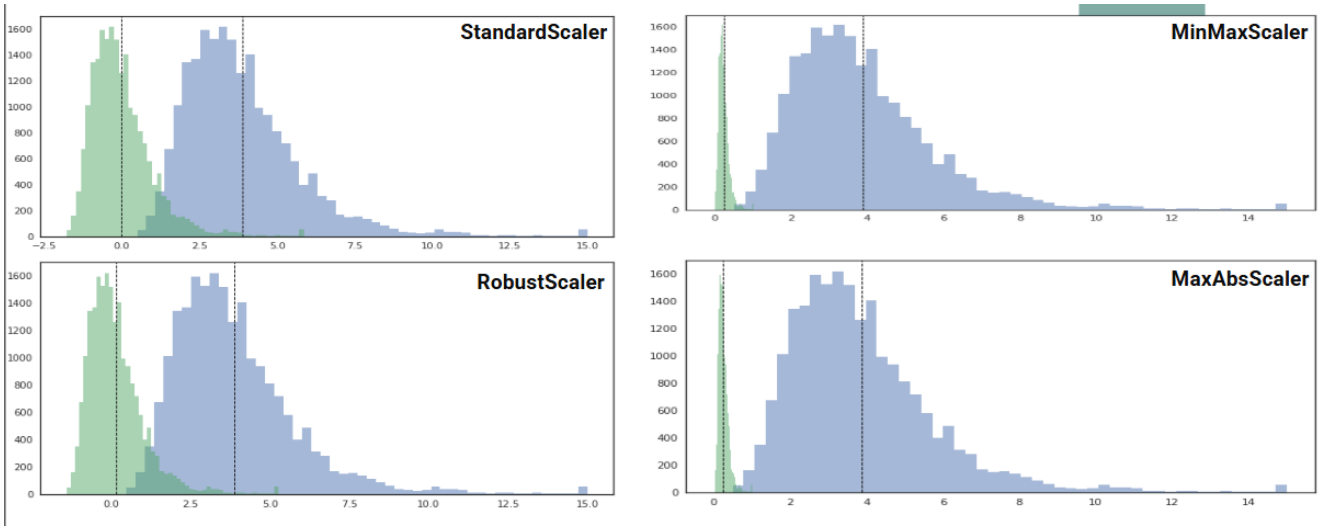


Рис. 10 – Візуалізація методів трансформації

Таблиця 2 – Методи кодування якісних ознак

Метод	Умови використання	Результат
One-Hot	1. Категоріальні дані, які не мають природної впорядкованості	Вектор бінарних змінних
	2. Кількість категорій відносно мала	
Label	1. Категоріальні дані, які мають природну впорядкованість	Цілі числа замість категорій
	2. У подальшому використовується методи на основі дерев	
Target	Велика кількість унікальних значень категорій	Вектор ймовірностей (частот) спостереження кожного класу для кожного унікального значення категорії, або середнє значення залежної змінної

One-Hot

	Area	Year	Item	Cassava	Maize	Plantains and others	Potatoes	Rice, paddy	Sorghum	Soybeans	Sweet potatoes	Wheat	Yams
0	Albania	1990	Maize	0	1	0	0	0	0	0	0	0	0
1	Albania	1990	Potatoes	0	0	0	1	0	0	0	0	0	0
2	Albania	1990	Rice, paddy	0	0	0	0	1	0	0	0	0	0
3	Albania	1990	Sorghum	0	0	0	0	0	1	0	0	0	0
4	Albania	1990	Soybeans	0	0	0	0	0	0	1	0	0	0

Target encoding

	Area	Year	Item	Items_target_enc
0	Albania	1990	Maize	36310.07
1	Albania	1990	Potatoes	199801.55
2	Albania	1990	Rice, paddy	40730.43
3	Albania	1990	Sorghum	18635.78
4	Albania	1990	Soybeans	16731.09

Label Encoding

	Area	Year	Item	Items_LE
0	Albania	1990	Maize	3
1	Albania	1990	Potatoes	9
2	Albania	1990	Rice, paddy	4
3	Albania	1990	Sorghum	1
4	Albania	1990	Soybeans	0

Рис. 11 – Порівняння методів кодування ознак

Таблиця 3 – Методи вибору ознак для побудови моделі

Метод	Способи втілення
Фільтрація	прибирання ознак із майже унікальними або майже постійними значеннями
Статистичний відбір	Для задач регресії: • Коефіцієнт кореляції Пірсона (f_regression) • Mutual information (mutual_info_regression) $MI(X,Y) = \iint p(x,y) log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy$
	Для задач класифікації: • Chi-2 тест (chi2) • ANOVA F-значення (f_classif) • Matual information (mutual_info_classif) $I(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y) log\left(\frac{p(x,y)}{p(x)p(y)}\right)$
Вибір на основі моделі	L1 регуляція, Feature Importance, Feature Permutation

Забезпечення надійності моделі за допомогою розділення даних

Забезпечення надійності моделі є критично важливим етапом у процесі машинного навчання, і один із ключових аспектів цього процесу — розподіл даних на три основні частини: навчальну (train), валідаційну (validation) та тестову (test) вибірки.

Навчальна вибірка використовується для тренування моделі, на основі якої вона навчається розпізнавати закономірності в даних. Валідаційна вибірка служить для налаштування гіперпараметрів моделі і визначення її продуктивності під час навчання, що дозволяє відстежувати можливість перенавчання.

Тестова вибірка використовується для остаточної оцінки моделі після її навчання і налаштування, надаючи дієву оцінку її здатності до генералізації на нових, невідомих даних. Додатково, використовування крос-валідації дозволяє ефективніше оцінити модель, розбиваючи дані на кілька підвибірок і проводячи навчання/тестування на різних їх комбінаціях. Це забезпечує більш стабільні оцінки моделей, оскільки дозволяє максимально використовувати доступні дані і зменшує ризик випадкових похибок, пов'язаних з наявними вибірками. Завдяки продуманому розподілу даних, можна досягти оптимізації продуктивності моделі та підвищити її надійність в умовах реального використання.

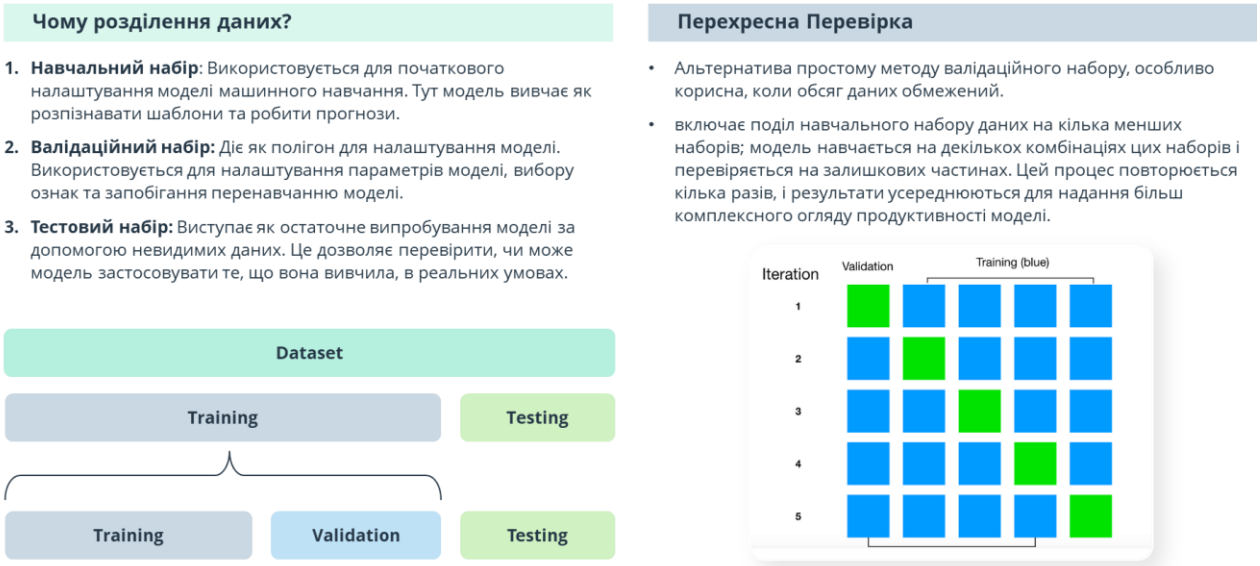


Рис. 12 – Забезпечення надійності моделі за допомогою розділення даних

Коли класи в навчальному наборі даних представлені нерівномірно у задачах класифікації виникає проблема незбалансованості спостережень. Це означає, що одна або декілька категорій мають значно більше прикладів, ніж інші. Наприклад, у задачі виявлення хвороб рослин, приклади здорових культур можуть суттєво перевершувати хворі, що призводить до важливих проблем. Моделі, навчальні на таких даних, можуть показувати високу загальну точність, однак не матимуть адекватної здатності розпізнавати менш представлені класи, що може призвести до серйозних наслідків у застосуванні.

Для вирішення цих проблем використовують спеціальні способи підготовки та навчання моделей на незбалансованих даних:

1. Перебалансування вхідних даних (Oversampling) - цей метод полягає в збільшенні кількості спостережень менш представленого класу шляхом випадкового копіювання прикладів або генерації нових (наприклад, з використанням SMOTE — Synthetic Minority Over-sampling Technique). Це дозволяє моделі отримати більше інформації про труднощі розпізнавання меншого класу.

2. Зменшення вхідних даних (Undersampling) — цей підхід полягає у зменшенні кількості спостережень більш представленого класу, щоб збалансувати навчальний набір. Хоча цей метод забезпечує збалансованість

даних, він не завжди є оптимальним, оскільки може призвести до втрати важливої інформації.

3. Зважування помилки в процесі навчання - під час навчання моделі можна використовувати зважені функції втрат, при цьому присвоюючи більшу вагу помилкам для менш представлених класів. Це дозволяє моделі зосереджуватися на покращенні точності в розпізнаванні рідкісних класів без зменшення загальної продуктивності.

4. Вибір спеціальних метрик оцінки якості моделі - для оцінки моделей, навчених на незбалансованих даних, важливо використовувати спеціальні метрики, які відображають справжню продуктивність моделі. Приклади таких метрик включають:

Precision (точність) - відсоток правильних позитивних прогнозів з усіх позитивних прогнозів (менш представлених класів).

Recall (повнота) - відсоток правильно виявлених позитивних випадків з усіх справжніх позитивних (менш представлених класів).

F1-міра - гармонічне середнє між точністю і повнотою, що забезпечує баланс між цими двома метриками.

ROC-AUC - Графічне представлення співвідношення між справжньо позитивними та хибно позитивними результатами, що дозволяє оцінити якість класифікації.

Метрики оцінки якості моделей

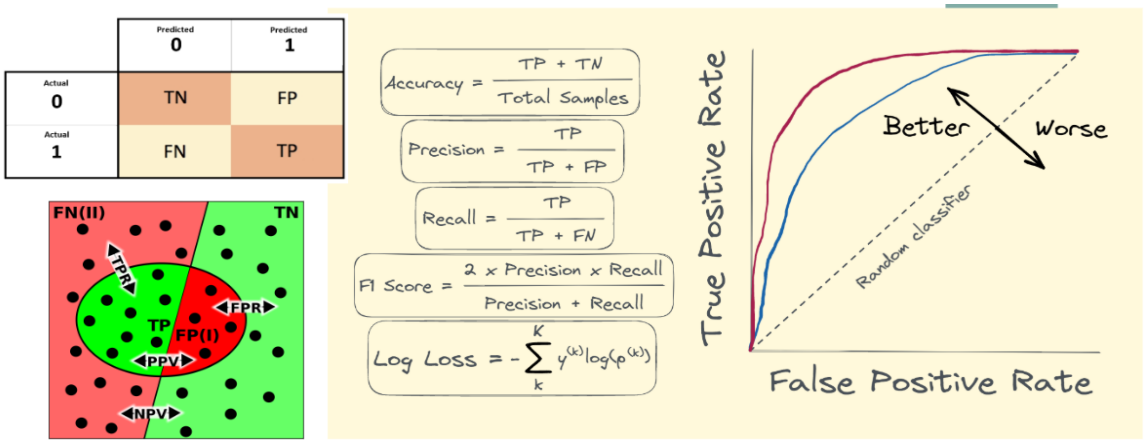


Рис. 13 – Метрики оцінки якості моделей (класифікація)

Метрики оцінки якості моделей регресії переважно засновані на оцінці відхилень між спостережуваними та прогнозованими значеннями. Основні метрики включають:

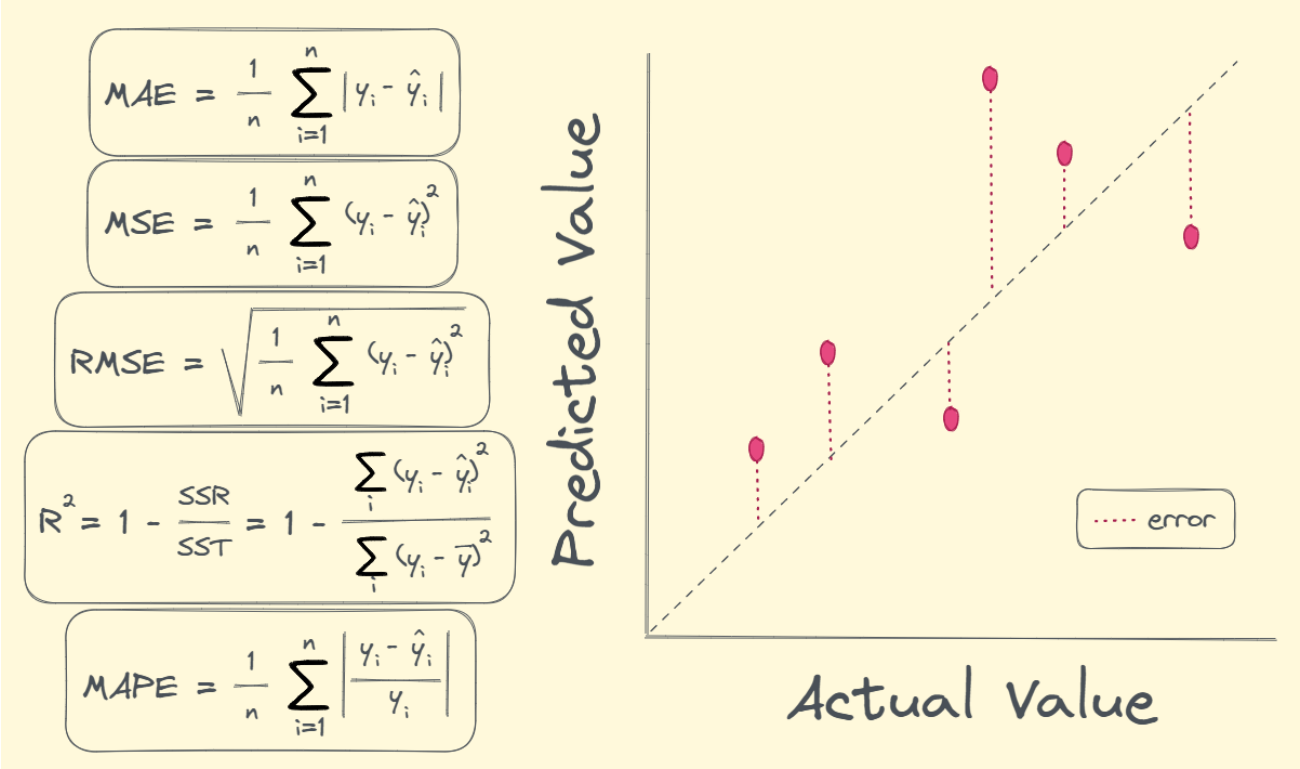


Рис. 14 – Метрики оцінки якості моделей регресії

Середня абсолютна помилка (MAE - Mean Absolute Error) - це середнє значення абсолютних різниць між фактичними (справжніми) та прогнозованими значеннями. MAE вимірює, наскільки далеко, у середньому, прогнози відправляються від справжніх значень. Використовуйте MAE, коли важливо мати однорідну, інтуїтивно зрозумілу метрику, оскільки вона вимірює помилки в тих же одиницях, що й вихідні дані.

Середня квадратична помилка (MSE - Mean Squared Error) - це середнє значення квадратів різниць між фактичними та прогнозованими значеннями. MSE є чутливим до великих помилок, оскільки використовуються квадрати помилок. Використовуйте MSE, коли важливо більше акцентувати увагу на більших помилках. Це може бути корисним у випадках, коли великі помилки мають серйозні наслідки.

Корінь середньої квадратичної помилки (RMSE - Root Mean Squared Error) - це квадратний корінь з MSE, що повертає метрику до тих же одиниць, що й вихідні дані, отже вона оцінює продуктивність різних моделей в одних і тих же одиницях, при цьому залишається чутливою до великих відхилень, як і MSE.

Коефіцієнт детермінації (R^2) - ця метрика показує частку варіації залежної змінної, яка може бути пояснена незалежними змінними у моделі. Значення R^2 варіюється від 0 до 1, де 1 означає, що модель повністю пояснює дані. Використовуйте R^2 для оцінки загальної якості моделі. Однак, він може бути оманливим, якщо не врахувати кількість параметрів у моделі.

ПРАКТИЧНА ЧАСТИНА

Практичне завдання 1. Налаштування середи розробника (Python)

1. Встановлення Python:

Завантажте та встановіть Python з офіційного веб-сайту (https://www.python.org/).

Під час встановлення переконайтеся, що вибрана опція додавання Python до системного шляху.

2. Встановлення Anaconda:

Завантажте та встановіть Anaconda з офіційного веб-сайту (https://www.anaconda.com/products/distribution).

Дотримуйтесь інструкцій щодо встановлення, наданих для вашої операційної системи.

Anaconda включає Python, разом із багатьма популярними бібліотеками та інструментами для науковців з даних.

3. Встановлення PyCharm чи Visual Studio:

Завантажте та встановіть PyCharm чи Visual Studio з офіційного веб-сайту (https://www.jetbrains.com/pycharm/download/).

Виберіть Community (безкоштовну) або Professional версію відповідно до ваших потреб.

Дотримуйтесь інструкцій щодо встановлення, наданих для вашої операційної системи.

4. Створення та налаштування середовища:

Із використанням Анаконду Навігатор:

- Відкрийте Анаконду Навігатор (встановлену на кроці 2).
- Перейдіть на вкладку "Середовища".
- Натисніть кнопку "Створити", щоб створити нове середовище.
- Вкажіть назву середовища та виберіть потрібну версію Python.
- Опційно виберіть додаткові пакети для встановлення в середовищі.
- Натисніть кнопку "Створити", щоб створити середовище.

Із використанням командної строки:

- Запустити Anaconda Powershell
- створення нового середовища: conda create --name myenv
- Активувати середовище conda activate myenv

5. Додавання середовища до Jupyter Notebook:

- Відкрийте Анаконду Навігатор.
- Перейдіть на вкладку "Домашня".
- Запустіть Jupyter Notebook, натиснувши кнопку "Запустити".

У інтерфейсі Jupyter Notebook перейдіть до меню "Новий" і виберіть "Python [conda env:<environment_name>]" для створення нового ноутбуку використовуючи вказане середовище.

Додавання середовища до Jupyter Notebook через командну строку:

- Активувати середовище conda activate myenv
- Встановити ipykernel: pip install ipykernel
- Додавання середовища до jupyter notebook python -m ipykernel install --user --name myenv --display-name "Python (myenv)"

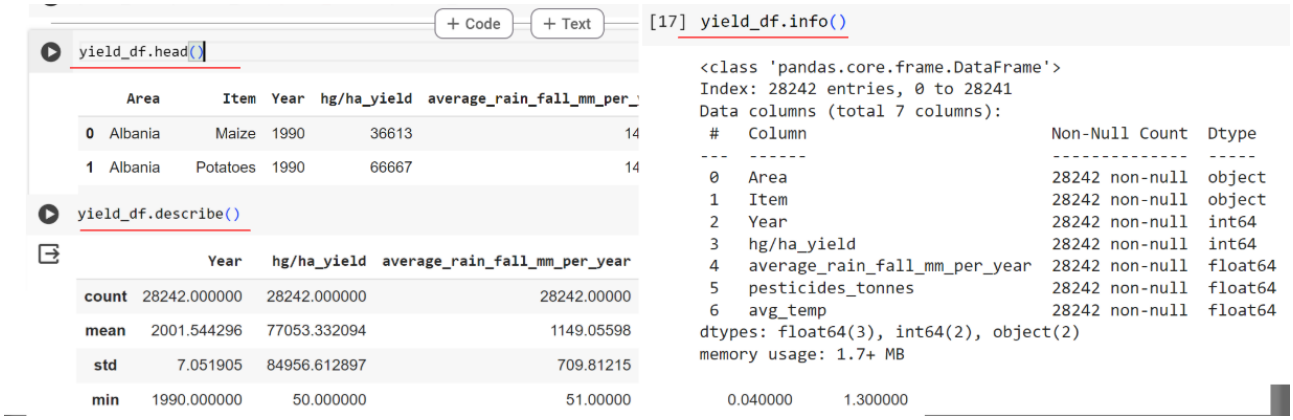
Практичне завдання 2. Розвідувальний аналіз даних.

Методи розвідувального аналізу (EDA) та підготовки даних:

1. Підготувати дані для аналізу (завантажте власний набір даних як датафрейм библиотеки pandas).

```
yield_df = pd.read_csv('yield_df.csv', index_col=0)
```

2. Провести загальний аналіз даних структури та якості даних використовуйте методи shape, head, describe, info бібліотеки pandas.



```
yield_df.head()
```

	Area	Item	Year	hg/ha_yield	average_rain_fall_mm_per_
0	Albania	Maize	1990	36613	14
1	Albania	Potatoes	1990	66667	14

```
yield_df.describe()
```

	Year	hg/ha_yield	average_rain_fall_mm_per_year
count	28242.000000	28242.000000	28242.00000
mean	2001.544296	77053.332094	1149.05598
std	7.051905	84956.612897	709.81215
min	1990.000000	50.000000	51.00000

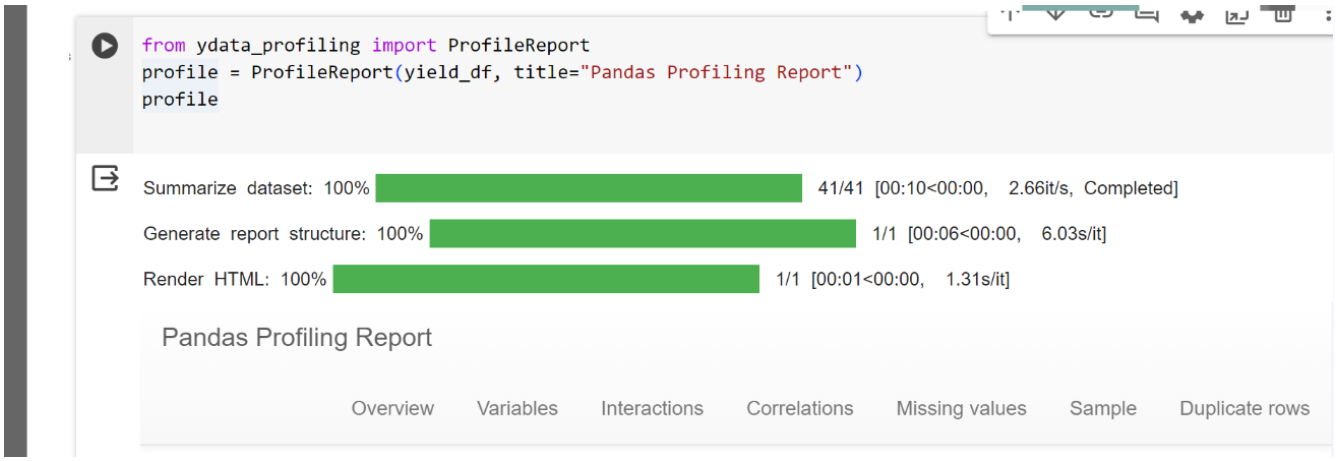
```
[17] yield_df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 28242 entries, 0 to 28241
Data columns (total 7 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Area                            28242 non-null  object
 1   Item                            28242 non-null  object
 2   Year                            28242 non-null  int64
 3   hg/ha_yield                     28242 non-null  int64
 4   average_rain_fall_mm_per_year   28242 non-null  float64
 5   pesticides_tonnes               28242 non-null  float64
 6   avg_temp                        28242 non-null  float64
dtypes: float64(3), int64(2), object(2)
memory usage: 1.7+ MB
```

0.040000 1.300000

Використання цих методів дасть уявлення про наступні характеристики даних:

- типи полів
- назви та сутність кожного признака
- дослідити загальні статистичні харакрeристики кожного признака

Альтернативним спопобом є дослідження загальних характеристик за допомогою бібліотеки ProfileReport:



```
from ydata_profiling import ProfileReport
profile = ProfileReport(yield_df, title="Pandas Profiling Report")
profile
```

Summarize dataset: 100% 41/41 [00:10<00:00, 2.66it/s, Completed]

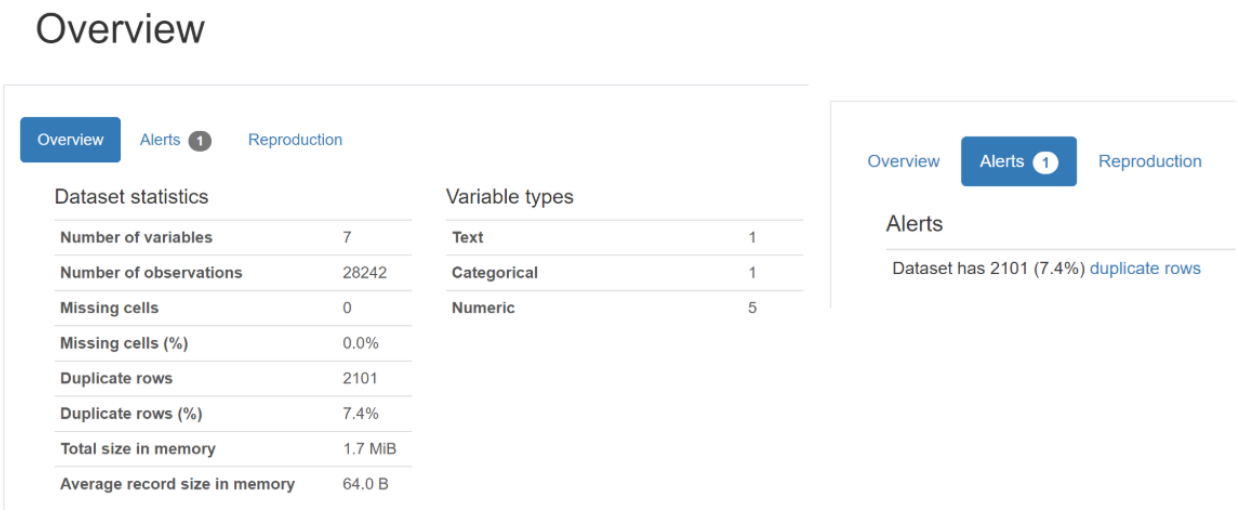
Generate report structure: 100% 1/1 [00:06<00:00, 6.03s/it]

Render HTML: 100% 1/1 [00:01<00:00, 1.31s/it]

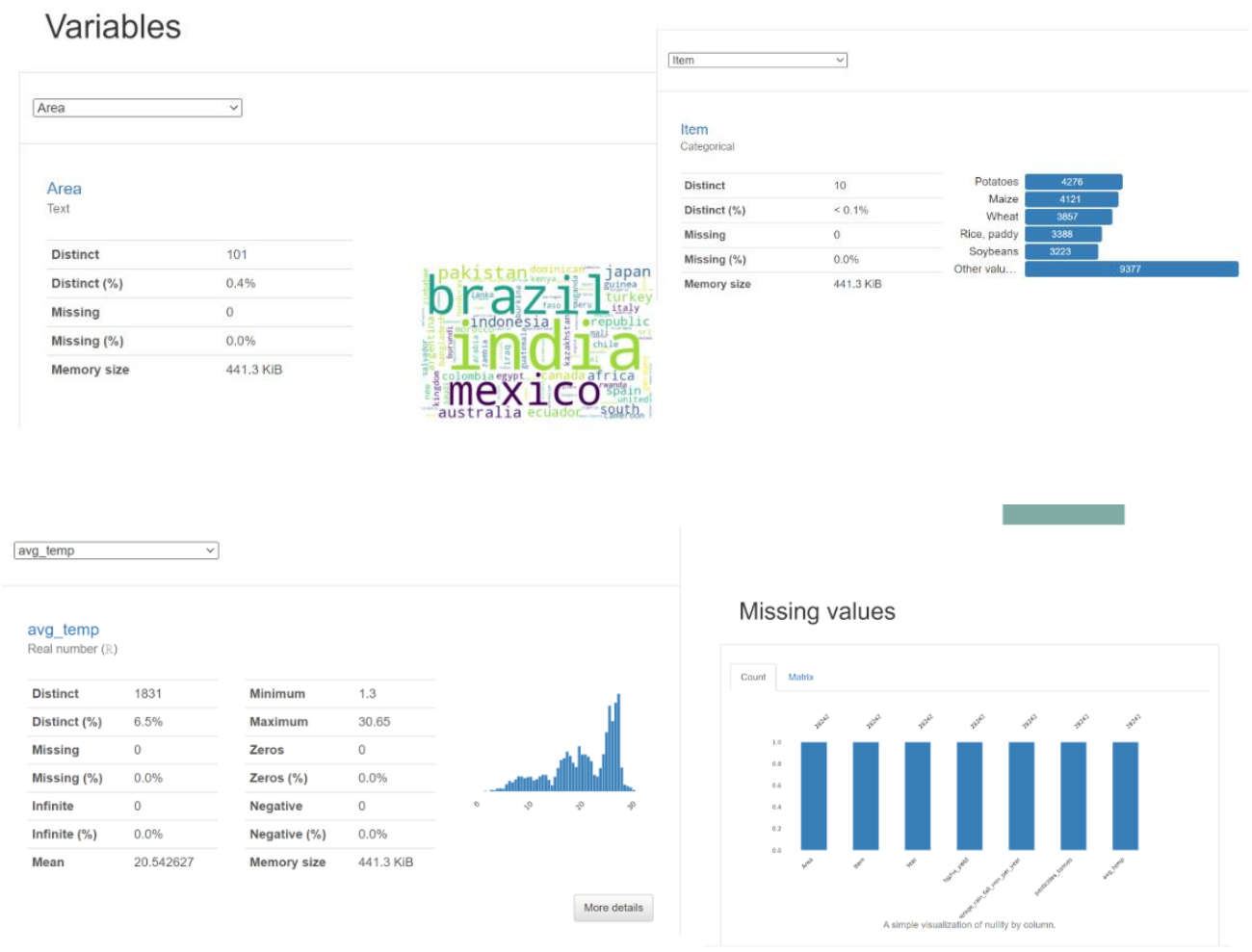
Pandas Profiling Report

Overview Variables Interactions Correlations Missing values Sample Duplicate rows

Загальний огляд даних: ProfileReport



Загальний огляд даних: ProfileReport - аналіз змінних



3. Провести аналіз та виправлення пропущених даних різними способами (якщо обрані дані не містять пропущених значень, самостійно видалити 5% значень за обраними змінними)

- Провести виправлення пропущених даних із використанням sklearn *SimpleImputer*

sklearn.impute.SimpleImputer

class sklearn.impute.**SimpleImputer**(*, *missing_values=nan*, *strategy='mean'*, *fill_value=None*, *copy=True*, *add_indicator=False*, *keep_empty_features=False*)

missing_values ***int, float, str, np.nan, None or pandas.NA, default=np.nan***

strategy, *default='mean'* - replace missing values using the mean along each column.
"median" - replace missing values using the median along each column.
"most_frequent" - replace missing using the most frequent value along each column.
"constant" - replace missing values with fill_value.

- Провести виправлення пропущених даних із використанням sklearn *KNNImputer*

sklearn.impute.KNNImputer

class sklearn.impute.**KNNImputer**(*, *missing_values=nan*, *n_neighbors=5*, *weights='uniform'*, *metric='nan_euclidean'*, *copy=True*, *add_indicator=False*, *keep_empty_features=False*)

missing_values*int, float, str, np.nan or None, default=np.nan*
n_neighbors*int, default=5* Number of neighboring samples to use for imputation.

weights*{'uniform', 'distance'} or callable, default='uniform'* Weight function used in prediction.
metric*{'nan_euclidean'} or callable, default='nan_euclidean* Distance metric for searching neighbors.

4. Виявлення та обробка викидів та аномалій

Викиди та аномалії представляють собою завеликі або замалі значення що є нетиповими або помилковими. Зазвичай ці значення зустрічаються досить рідко. Виходячи із цієї гіпотези викидами та аномаліями вважаються всі дуже рідкі значення. Для їх ідентифікації використовуйте методи 3 Сигма, Inter-Quartile Range (IQR), a-Percentile

(якщо обрані дані не містять викидів, самостійно додати 5 нетипових значень за обраними змінними)

Виправлення викидів полягає або в видаленні спостережень (метод Trimming), або у коригуванні таких значень (метод Capping)

Методи виявлення:

1. Правило **3 Сигма** - якщо дані розподілені нормально
2. Правило **Inter-Quartile Range (IQR)**:
   (Q1 – 1.5 IQR, Q3 + 1.5 IQR), де IQR = Q3 - Q1
3. a-Percentile:
   (1% percentile, 99% percentile)

Методи обробки:

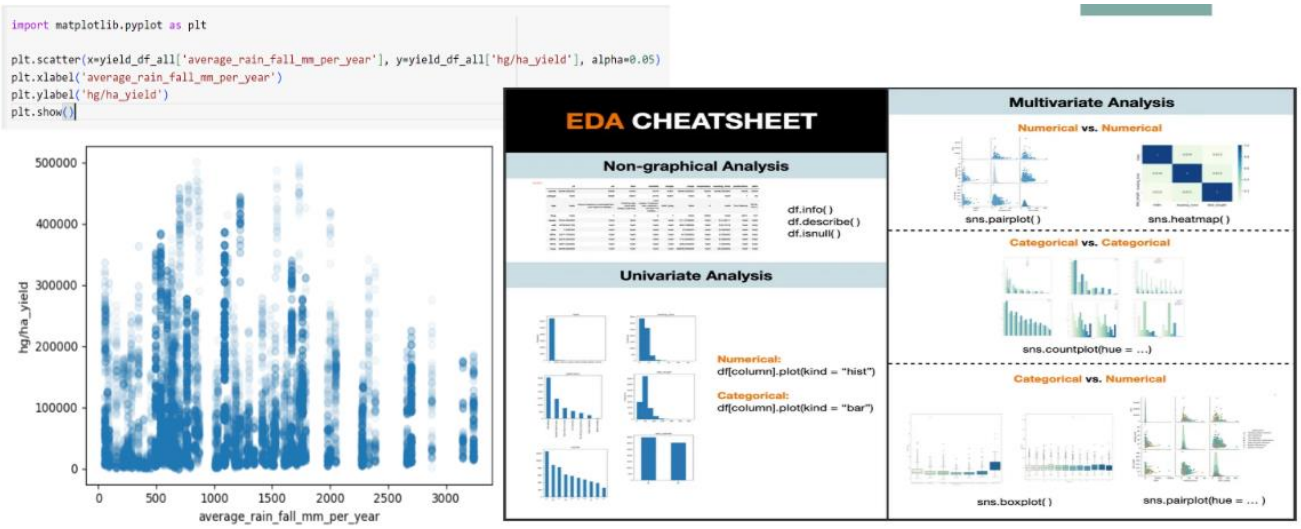
Trimming - ***видалення*** - спостереження із викидами видаляються із аналізу.
Capping - ***обмеження*** - значення, що виходять за межі певних граничних значень замінюються на ці граничні або якісь наперед задані значення.
Discretization - ***дискретизація*** - заміна фактичних даних на групи.

5. Провести аналіз розподілів та пошук зв'язків в даних

Для візуалізацій використовуйте бібліотеки matplotlib, seaborn, plotly.

Для аналіза окремих змінних використовуйте гістограми, діагрвми «Ящик з вусами» (Boxplot), «Скрипичні діаграми» (violin plot)

Для аналізу зв’язків використовуйте: парні діаграми (pairplots), діаграми розсіювання (scatter plots), теплові карти по матрице корреляции (heat maps).



Практичне завдання 3. Методи підготовки та розробки ознак для побудови моделі машинного навчання.

1. Методи масштабування:

- провести масштабування кількісних ознак за допомогою метода стандартизації (*StandatdScaler*)

```
#Standardization
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_train_std=sc.fit_transform(X_train)
X_validation_std=sc.transform(X_validation)
```

- провести масштабування кількісних ознак за допомогою метода масштабування (*MinMaxScaler*)

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()    # create an object of class

scaler.fit(x_train)       #  fit on training i/p data

x_train_scaled = scaler.transform(x_train)    # transform model on both traing and testing data
x_test_scaled = scaler.transform(x_test)
```

- провести масштабування кількісних ознак за допомогою метода "стійкого" масштабування (*RobustScaler*)

```
from sklearn.preprocessing import RobustScaler

r_scaler = RobustScaler()

x_train_scaled = r_scaler.fit_transform(x_train)

x_test_scaled = r_scaler.transform(x_test)
```

2. Методи кодування якісних ознак:

1. провести кодування якісних ознак із використанням методу *OneHotEncoding*

```
import pandas as pd
from sklearn.preprocessing import OneHotEncoder

# define example
data = pd.DataFrame({
    'value': range(10),
    'weather': ['cold', 'cold', 'warm', 'cold', 'hot', 'hot', 'warm', 'cold', 'warm', 'hot'],
})

onehot_encoder = OneHotEncoder(sparse=False)

onehot_encoded = onehot_encoder.fit_transform(data[['weather']])
```

– провести кодування якісних ознак із використанням методу *LabelEncoding*

```
from numpy import array
from numpy import argmax
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# define example
data = pd.DataFrame({
    'value': range(10),
    'weather': ['cold', 'cold', 'warm', 'cold', 'hot', 'hot', 'warm', 'cold', 'warm', 'hot'],
})

label_encoder = LabelEncoder()

label_encoded = label_encoder.fit_transform(data[['weather']])
```

– провести кодування якісних ознак із використанням методу *TargetEncoding*

```
import pandas as pd
from sklearn.preprocessing import TargetEncoder

# Define the example DataFrame with a target column
data = pd.DataFrame({
    'weather': ['cold', 'cold', 'warm', 'cold', 'hot', 'hot', 'warm', 'cold', 'warm', 'hot'],
    'yield': [10, 15, 25, 20, 30, 35, 28, 18, 22, 31]  # Hypothetical target values
})

# Initialize the TargetEncoder with fewer folds
target_encoder = TargetEncoder(target_type='continuous')

# Fit and transform the data
data['weather_encoded'] = target_encoder.fit(data[['weather']], data['yield']).transform(data[['weather']])
```

```
Data with Target Encoding:
   weather  yield  weather_encoded
0     cold     10        16.211559
1     cold     15        16.211559
2     warm     25        24.944095
3     cold     20        16.211559
4      hot     30        31.764457
5      hot     35        31.764457
6     warm     28        24.944095
7     cold     18        16.211559
8     warm     22        24.944095
9      hot     31        31.764457
```

3. **Сформувати шість наборів даних** для подальшого аналізу підготовлених різними способами поєднання методів підготовки даних:

- Набір 1 - Метод масштабування_1, метод_кодування_1
- Набір 2 - Метод масштабування_2, метод_кодування_1
- Набір 3 - Метод масштабування_1, метод_кодування_2
- Набір 4 - Метод масштабування_2, метод_кодування_2
- Набор 5 - Метод масштабування_1, метод_кодування_3
- Набор 6 - Метод масштабування_2, метод_кодування_3

Практичне завдання 4. Тренування та вибір найкращої моделі.

Для обраної задачі сформулювати та обрати ключовий показник якості моделі:

- Обрати метод валідації (проста або кросс-валідація) та провести розподіл датасетів на тренувальні та валідаційні
- Обрати декілька типів підходів для моделювання (наприклад лінійна модель, ансамблі дерев (bagging), ансамблі дерев (boosting) та провести тренування кожної з обраних моделей на кожному з підготовлених датасетів для тренування (має бути 18 моделей 6 датасетів х 3 типа моделей)
- Розрахувати якість кожної з натренованих моделей на відповідних валідаційних датасетах
- Обрати найкращий спосіб підготовки даних та тип моделі для прогнозування

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013
2. Aurélien Géron Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition : O'Reilly Media, Inc. 2022
3. Andreas Muller, Sarah Guido. Introduction to Machine Learning with Python: A Guide for Data Scientists 1st Edition: O'Reilly, 2016
4. Chris Albon, Kyle Gallatin. Machine Learning with Python Cookbook.: O'Reilly, 2023

Додаткова

1. API Reference scikit-learn https://scikit-learn.org/stable/modules/classes.html
2. Курси Udemy.com
3. Курси Coursea.org