# Практична робота №3 Підготовка даних для обробки

**Мета роботи**: отримати навички первинної обробки і підготовки даних: очищення, нормалізації та стандартизації даних (підготовки даних), покращити навички роботи з пакетом pandas.

## 1 Теоретичні відомості

### 1.1 Етапи первинної обробки і підготовки даних

**Підготовка і первинна обробка даних (data wrangling, data munding)** — це процес очищення, структурування та перетворення необроблених даних у придатний для аналізу формат.

Цей процес включає такі завдання, як обробка відсутніх або неузгоджених даних, форматування типів даних і об’єднання різних наборів даних для підготовки даних для подальшого дослідження та моделювання в аналізі даних або проектах машинного навчання.

Оскільки кількість даних і джерел даних швидко зростають і розширюються, стає все більш важливою організацією великих обсягів доступних даних для аналізу. Цей процес зазвичай включає в себе ручне перетворення та відображення даних з однієї форми в інший формат, щоб забезпечити більш зручне використання та організацію даних.

Процес обробки даних

Обробка даних передбачає систематичне та повторюване перетворення необроблених, неструктурованих або безладних даних у чистий, структурований і придатний для використання формат для науки про дані та аналітики (рис. 1).

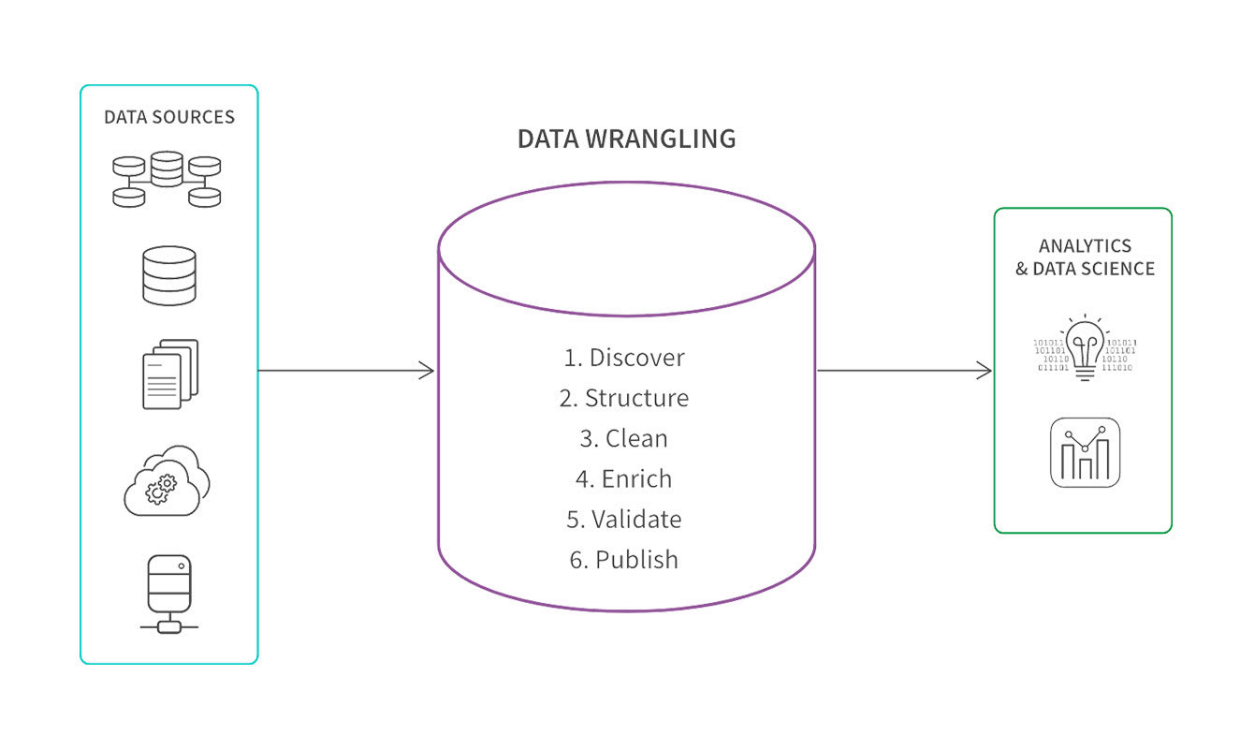


Рис. 1 Блок-схема підготовки і первинної обробки даних

Блок-схема процесу обробки даних містить наступні кроки: виявлення, структурування, очищення, збагачення, перевірка та публікація. Джерела даних розташовано ліворуч, а аналітика та наука про дані – праворуч.

Нижче наведено опис цих 6 ключових кроків:

Крок 1: Виявлення даних

Спочатку ваша увага зосереджена на розумінні та дослідженні даних, які ви зібрали. Це передбачає визначення джерел даних, оцінку якості даних і отримання уявлень про структуру та формат даних. Ваша мета — створити основу для наступних етапів підготовки даних, визнаючи потенційні проблеми та можливості в даних.

Крок 2: Структурування даних

На етапі структурування даних ви впорядковуєте та форматуєте необроблені дані таким чином, щоб полегшити ефективний аналіз. Конкретна форма ваших даних залежить від того, яку аналітичну модель ви використовуєте, але структурування зазвичай передбачає зміну форми даних, обробку відсутніх значень і перетворення типів даних. Це гарантує, що дані представлені в узгодженій і стандартизованій формі, закладаючи основу для подальших маніпуляцій і досліджень.

Крок 3: Очищення даних

Очищення даних — важливий крок для усунення невідповідностей, помилок і викидів у наборі даних. Це передбачає видалення або виправлення неточних даних, обробку дублікатів і усунення будь-яких аномалій, які можуть вплинути на надійність аналізів. Очищаючи дані, ви зосереджуєтеся на підвищенні точності та надійності даних для подальших процесів.

Крок 4: Збагачення даних

Збагачення ваших даних передбачає доповнення наявних даних додатковою інформацією для надання більш зрозумілих контексту чи глибини. Це може включати об’єднання наборів даних, виявлення релевантних ознак або включення зовнішніх джерел даних. Мета полягає в тому, щоб розширити вихідний набір даних, зробивши його більш повним і цінним для аналізу. Якщо ви додаєте дані, обов’язково структуруйте та очистіть ці нові дані.

Крок 5: Перевірка даних

Перевірка гарантує якість і надійність ваших оброблених даних. Треба перевірити наявність невідповідностей, перевірите цілісність даних і підтвердити, що дані відповідають попередньо визначеним стандартам. Перевірка допомагає зміцнити впевненість у точності набору даних і гарантує, що він відповідає вимогам для змістовного аналізу.

Крок 6: Публікація даних

Публікація даних або розповсюдження серед бізнес-користувачів передбачає документування походження даних і кроків, зроблених під час усього процесу суперечок, обмін метаданими та підготовку даних для зберігання або інтеграції в інструменти аналізу даних і аналітики. Публікація полегшує співпрацю та дозволяє іншим використовувати дані для аналізу чи прийняття рішень.

Основні операції data wrangling виконуються за допомогою пакетів pandas, matplotlib, seaborn.

### 1.2 Зразок виконання послідовності підготовки даних

Дані для аналізу отримаємо з набору даних

https://archive.ics.uci.edu/dataset/10/automobile

Цей набір даних складається з трьох типів об’єктів:

1. специфікація автомобіля з точки зору різних характеристик,
2. присвоєний йому рейтинг страхового ризику,
3. його нормалізовані втрати під час використання порівняно з іншими автомобілями.

Другий рейтинг відповідає ступеню ризику автомобіля, ніж вказує його ціна. Автомобілям спочатку присвоюється символ фактора ризику, пов'язаний з його ціною. Потім, якщо це більш ризиковано (або менше), цей символ коригується шляхом переміщення його вгору (або вниз) за шкалою. Актуарії називають цей процес «символізація». Значення +3 вказує на те, що авто є ризикованим, -3, що воно, ймовірно, досить безпечне.

Третім фактором є відносна середня виплата збитку за рік страхування автомобіля. Це значення нормалізовано для всіх автомобілів у певній класифікації розміру (дводверні маленькі, універсали, спортивні/спеціальні тощо) і являє собою середні втрати на автомобіль на рік.

Дані можна імпортувати безпосередньо з репозиторію UC Irvine Machine Learning Repository після встановлення пакету ucimlrepo

pip install ucimlrepo

Необхідний набір команд наведено нижче:

from ucimlrepo import fetch\_ucirepo

# fetch dataset

automobile = fetch\_ucirepo(id=10)

# data (as pandas dataframes)

X = automobile.data.features

y = automobile.data.targets

# metadata

print(automobile.metadata)

# variable information

print(automobile.variables)

Інший вариант завантаження даних – використання методу Pandas read\_csv(), щоб завантажити дані з завантаженого файлу. Приклад:

import pandas as pd

filename = "automobile/imports-85.data"

names = [

'symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiration', 'num-of-doors',

'body-style', 'drive-wheels', 'engine-location', 'wheel-base', 'length',

'width', 'height', 'curb-weight', 'engine-type', 'num-of-cylinders',

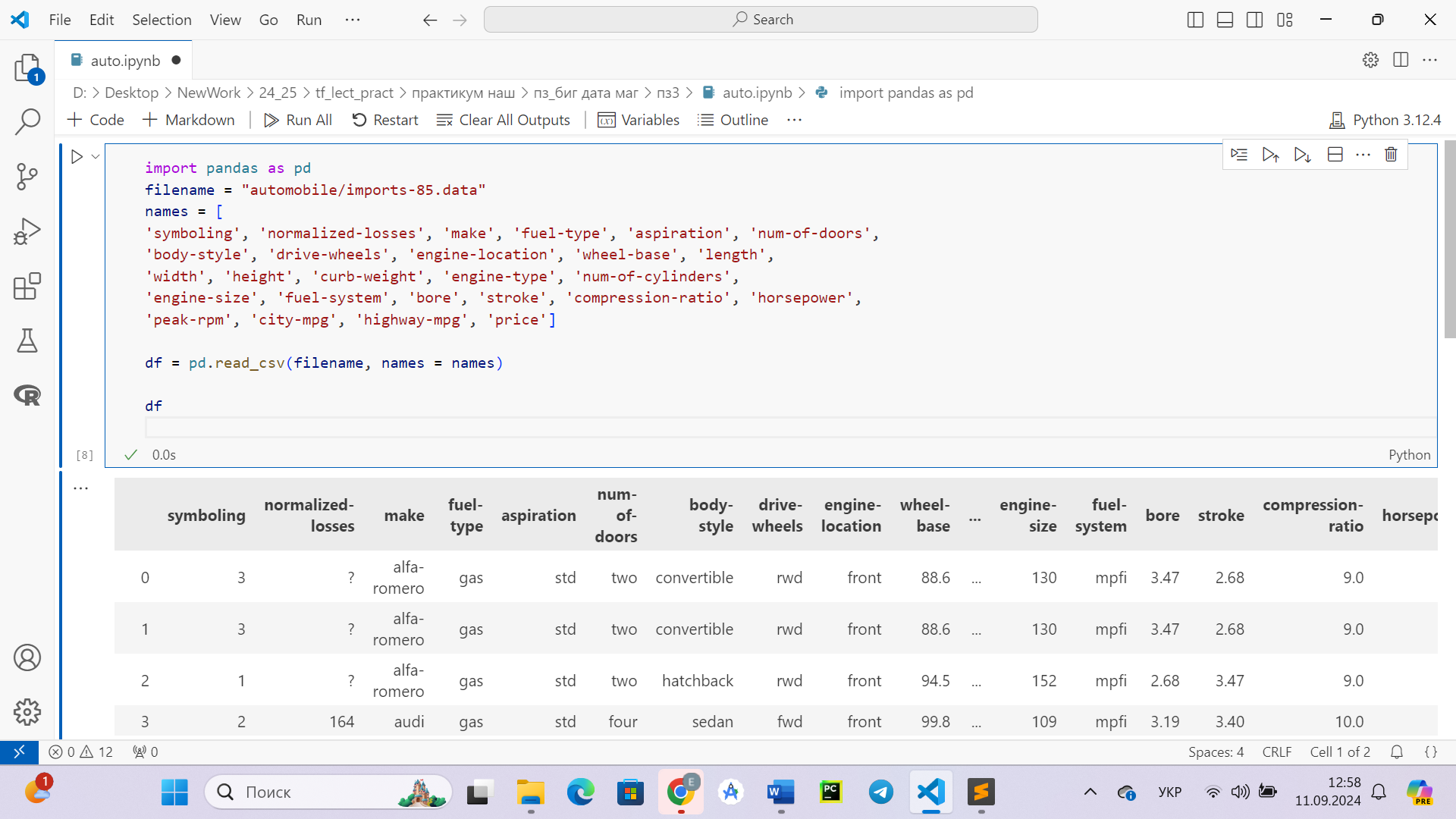
'engine-size', 'fuel-system', 'bore', 'stroke', 'compression-ratio', 'horsepower',

'peak-rpm', 'city-mpg', 'highway-mpg', 'price']

df = pd.read\_csv(filename, names = names)

df

Отримаємо наступний результат:



Використовуйте метод head(), щоб відобразити перші п’ять рядків фрейму даних.

#### 1.2.1 Визначення та обробка відсутніх значень

Задача перетворити "?" до NaN (not a number).

У наборі даних автомобілей є відсутні дані. позначені знаком питання "?". Замінюємо "?" на NaN (не число), маркером відсутнього значення Python за замовчуванням з міркувань швидкості та зручності обчислень. Тут ми використовуємо функцію:

.replace(A, B, inplace = True) замінити А на В.

df.replace("?", np.nan, inplace = **True**)

#### 1.2.2 Оцінка на предмет відсутніх даних

Відсутні значення перетворюються за замовчуванням. Існує два способи виявлення відсутніх даних:

.isnull()

.notnull()

missing\_data = df.isnull() missing\_data.head(5)

Вихідним є логічне значення, яке вказує, чи фактично в значенні, яке передається в аргумент, відсутні дані чи наявні. Порахуйте пропущені значення в кожному стовпці. Використовуючи цикл for у Python, ми можемо швидко визначити кількість відсутніх значень у кожному стовпці. Як зазначалося вище, "True" представляє відсутнє значення, а "False" означає, що значення присутнє в наборі даних. У тілі циклу for метод ".value\_counts()" підраховує кількість значень "True".

Можливе рішення задачі поставленої вище:

**for** column **in** missing\_data.columns.values.tolist():

print(column)

**print** (missing\_data[column].value\_counts()) print("")

#### 1.2.3 Як боротися з відсутніми даними?

Найпростіший варіант - видаліть дані, а саме

1. Видаліть весь рядок
2. Видаліть всю колонку

Трохи складніши рішення - замініть дані

1. Заміна відсутніх даних на середнє значення
2. Заміна відсутніх даних за частотою
3. Заміна відсутніх даних за допомогою інших функцій

Цілі стовпці слід вилучати, лише якщо більшість записів у стовпці порожні. В розглянутому наборі даних жоден із стовпців не є достатньо порожнім, щоб повністю його видалити. Є певна свобода у виборі методу заміни даних; однак деякі методи можуть здатися більш розумними, ніж інші.

Деякі приклади роботи з даними:

**Обчисліть середнє значення для стовпця «normalizedlosses»**

avg\_norm\_loss = df["normalized-losses"].astype("float").mean(axis=0) print("Average of normalized-losses:", avg\_norm\_loss)

**Замініть "NaN" середнім значенням у стовпці "normalizedlosses"**

df["normalized-losses"].replace(np.nan, avg\_norm\_loss, inplace=True)

**Обчисліть середнє значення для стовпця 'bore'**

avg\_bore=df['bore'].astype('float').mean(axis=0) print("Average of bore:", avg\_bore)

**Замініть "NaN" середнім значенням у стовпці 'bore'**

df["bore"].replace(np.nan, avg\_bore, inplace=True)

### 1.2.4 Коректування формату даних

Останнім кроком очищення даних є перевірка та переконання, що всі дані мають правильний формат (int, float, текстовий або інший).

У Pandas використовуються методи:

.dtype(), щоб перевірити тип даних;

.astype(), щоб змінити тип даних.

Приклад перетворення типу даних у необхідний формат:

df[["bore", "stroke"]] = df[["bore", "stroke"]].astype("float") df[["normalized-losses"]] = df[["normalizedlosses"]].astype("int") df[["price"]] = df[["price"]].astype("float") df[["peak-rpm"]] = df[["peak-rpm"]].astype("float")

### 1.2.5 Деякі інши методи робоьт з даними

**Стандартизація даних**

Дані зазвичай збираються від різних агентств у різних форматах. (Стандартизація даних також є терміном для певного типу нормалізації даних, де ми віднімаємо середнє і ділимо на стандартне відхилення.)

Стандартизація — це процес перетворення даних у загальний формат, що дозволяє досліднику проводити змістовне порівняння.

**Нормалізація даних**

Нормалізація — це процес перетворення значень кількох змінних у подібний діапазон. Типові нормалізації включають масштабування змінної таким чином, щоб середнє значення змінної становило 0, масштабування змінної, щоб дисперсія становила 1, або масштабування змінної, щоб значення змінної були в діапазоні від 0 до 1.

**Бінінг**

Бінінг — це процес перетворення неперервних числових змінних у дискретні категоріальні «біни» для згрупованого аналізу.

df["horsepower"]=df["horsepower"].astype(int, copy=True)

**Індикаторна змінна**

Індикаторна змінна (або фіктивна змінна) — це числова змінна, яка використовується для позначення категорій. Їх називають dummies, тому що самі числа не мають внутрішнього значення. Ми використовуємо індикаторні змінні, щоб ми могли використовувати категоріальні змінні для регресійного аналізу в наступних модулях.

# 2 Завдання

1. Проаналізувати умову задачі.
2. Написати мовою Python програму для вирішення наступних завдань:
   1. Замініть NaN у стовпці ‘stroke’ середнім значенням.
   2. Нормалізуйте стовпець ‘height’.

2.2.1 Створіть змінну індикатора для стовпця ‘aspiration’

2.2.2 Об’єднайте новий фрейм даних із вихідним фреймом даних, а потім видаліть стовпець ‘aspiration’.

* 1. Замініть усі NaN у всіх стовбцях середнім значенням.

1. Результати роботи оформити протоколом.

## 1.5. Контрольні запитання

1. Що таке стандартизація?
2. Що таке нормалізація?
3. Якими способами можна позбутись відсутності даних?
4. Що таке бінінг?
5. Як нормалізувати дані?