KARTA PRACY UCZESTNIKA

Moduł 1 Powtarzalny workflow analizy danych w Jupyter Notebook

Od pytania analitycznego do pierwszego odtwarzalnego mini workflow

Czas	Pracujesz z	Rezultat
60–90 min	pakietem startowym, Jupyter Notebook lub JupyterLab	projektem M1 i notebookiem uruchamianym od początku do końca

W tej karcie nie szukasz jeszcze pełnego rozwiązania analitycznego. Uczysz się prowadzić pracę tak, aby pytanie, dane, kod, wynik i obserwacja tworzyły jeden czytelny proces.

Cel i rezultat modułu

Po wykonaniu karty przygotujesz przenośny katalog projektu, zapiszesz konkretne pytanie analityczne, zbudujesz notebook M1 oraz przejdziesz przez pierwszy mini workflow na danych pomiarowych. Wynik będzie odtwarzalny po restarcie kernela i ponownym uruchomieniu wszystkich komórek.

Po zakończeniu masz

- notebooks/M1_szkielet_analizy.ipynb z opisem celu, kodem, wynikiem, obserwacją i ograniczeniem;
- projekt z katalogami data/raw, data/processed, notebooks, outputs/figures, outputs/tables i docs;
- pierwszy wykres temperatury urządzenia U-07 zapisany jako outputs/figures/M1_temperatura_U07.png;
- pierwszy wpis w rejestrze LLM albo widoczną informację w notebooku, że LLM nie był używany.

Ważne: nie wpisuj oczekiwanych wartości z internetu ani nie poprawiaj danych, aby wynik wyglądał lepiej. Otrzymujesz go samodzielnie z kodu i danych źródłowych.

Zanim zaczniesz

Pakiet startowy jest materiałem wejściowym. Właściwy projekt analityczny tworzysz jako osobny katalog o stabilnej nazwie, na przykład eksploatacja_urzadzen. Dzięki temu późniejsze moduły pracują na tych samych nazwach katalogów i ścieżkach.

Element pakietu startowego	Miejsce w projekcie M1	Zasada
Dane/pomiary_eksploatacyjne.csv	data/raw/pomiary_eksploatacyjne.csv	Kopiujesz bez zmiany nazwy i nie nadpisujesz.
Dane/urzadzenia.xlsx	data/raw/urzadzenia.xlsx	Kopiujesz bez zmiany nazwy.
Dane/slownik_danych.md	docs/slownik_danych.md	Kopiujesz jako opis znaczenia pól.

Element pakietu startowego	Miejsce w projekcie M1	Zasada
Notebooki/notebook_student.ipynb	notebooks/M1_szkielet_analizy.ipynb	Tworzysz własną kopię roboczą, a oryginał pozostaje bez zmian.
Dokumentacja/szablon_rejestru_LLM.csv	docs/rejestr_uzycia_LLM.csv	Kopiujesz szablon i w pierwszym wpisie ustawiasz modul na M1.

Nie używaj w tym module nazwy M1_notatnik_start.ipynb. Jedna nazwa roboczego notebooka w całym kursie to M1_szkielet_analizy.ipynb. Nie stosuj też nazw typu final, final2 lub ostateczny. Nazwa powinna opisywać zawartość albo etap pracy.

```
eksploatacja_urzadzen/
├── data/raw/
├── data/processed/
├── docs/
├── notebooks/M1_szkielet_analizy.ipynb
├── outputs/figures/
├── outputs/tables/
└── README.md
```

Katalog data/processed pozostaw na razie pusty. Szczegółowe czyszczenie danych i zapis pomiary_clean.csv są zadaniem modułu 2.

Workflow pracy krok po kroku

1. **Krok 1 Przygotuj projekt —** Utwórz katalog projektu i jego sześć podkatalogów.
   - Z pakietu startowego skopiuj wyłącznie potrzebne pliki do miejsc wskazanych w tabeli.
   - Nie pracuj na plikach bezpośrednio w archiwum ZIP ani nie zmieniaj nazwy katalogu projektu w trakcie modułu.
   - W README.md zapisz jedno zdanie o celu projektu i datę rozpoczęcia pracy.
2. **Krok 2 Zapisz pytanie i plan —** Zanim napiszesz kod, ustal, na co dane mają odpowiedzieć.
   - Na potrzeby mini workflow użyj pytania: Jak zmieniała się temperatura urządzenia U-07 w okresie objętym zbiorem?
   - Zapisz źródło data/raw/pomiary_eksploatacyjne.csv, potrzebne pola czas, urzadzenie i temperatura oraz oczekiwany rezultat: podsumowanie, wykres, obserwacja i ograniczenie.
   - Wskaż odbiorcę wyniku, na przykład osobę odpowiedzialną za monitorowanie pracy urządzeń.
3. **Krok 3 Zbuduj szkielet notebooka —** Otwórz własną kopię M1_szkielet_analizy.ipynb i ułóż kolejność, którą może prześledzić inna osoba.
   - Utwórz sekcje: Cel i pytanie, Dane i źródła, Importy i konfiguracja, Wczytanie danych, Kontrola jakości, Przygotowanie danych, Analiza, Wizualizacja, Obserwacje i wnioski, Ograniczenia, Rejestr użycia LLM.
   - Używaj Markdown do opisania celu, decyzji i znaczenia wyniku. Kod służy do wykonania operacji i kontroli jej skutku.
   - Każda komórka kodu powinna mieć jedno logiczne zadanie, które potrafisz opisać jednym zdaniem.
4. **Krok 4 Skonfiguruj ścieżki —** Nie wpisuj ścieżki z dyskiem ani nazwą konta użytkownika.
   - Załóż, że notebook uruchamiasz z katalogu notebooks.
   - Najpierw wyświetl bieżący katalog roboczy i sprawdź metodą exists(), czy plik wejściowy jest dostępny.

- Jeżeli korzystasz z notebook_student.ipynb jako punktu wyjścia, zastąp jego wyszukiwanie folderu Dane konfiguracją zgodną z poniższym projektem.

```
from pathlib import Path
import pandas as pd
import matplotlib.pyplot as plt

PROJECT_DIR = Path.cwd().parent
RAW_DIR = PROJECT_DIR / "data" / "raw"
FIGURES_DIR = PROJECT_DIR / "outputs" / "figures"
plik = RAW_DIR / "pomiary_eksploatacyjne.csv"

print("Katalog roboczy:", Path.cwd())
print("Plik istnieje:", plik.exists())
```

5. **Krok 5 Wczytaj i sprawdź dane —** Pierwsze spojrzenie na plik nie jest jeszcze analizą.

- Wczytaj CSV do df, obejrzyj df.head(), df.shape, df.columns i df.dtypes.
- Sprawdź braki tylko w polach potrzebnych do odpowiedzi: czas, urzadzenie i temperatura.
- Sprawdź liczbę pełnych duplikatów. Jeśli problem wpływa na odpowiedź, zatrzymaj analizę i opisz decyzję. Nie ukrywaj problemu po to, aby przejść do wykresu.

```
df = pd.read_csv(plik)
print(df.shape)
print(df.columns)
print(df.dtypes)

kolumny = ["czas", "urzadzenie", "temperatura"]
print(df[kolumny].isna().sum())
print("Pełne duplikaty:", df.duplicated().sum())
```

6. **Krok 6 Przygotuj podzbiór —** Wybierz tylko dane potrzebne do pytania, a następnie sprawdź efekt przekształcenia.

- Odfiltruj U-07, pozostaw trzy potrzebne kolumny i utwórz niezależną kopię przez copy().
- Zamień czas na typ daty i uporządkuj rekordy rosnąco po czasie.
- Potwierdź kodem, że podzbiór obejmuje wyłącznie U-07, czas jest uporządkowany, a temperatura nie ma braków.

```
u07 = df.loc[
    df["urzadzenie"] == "U-07",
    ["czas", "urzadzenie", "temperatura"]
].copy()
u07["czas"] = pd.to_datetime(u07["czas"])
u07 = u07.sort_values("czas")

print(u07["urzadzenie"].unique())
print(u07["czas"].is_monotonic_increasing)
print(u07["temperatura"].isna().sum())
```

7. **Krok 7 Utwórz wynik i zapisz jego znaczenie —** Pokaż zmianę temperatury w czasie, ale nie dopowiadaj przyczyny bez dowodów.

- Oblicz count, min, mean i max dla temperatury.
- Przygotuj wykres z tytułem, opisem osi czasu i osią temperatury wraz z jednostką.
- Zapisz wykres jako outputs/figures/M1_temperatura_U07.png.
- W Markdown rozdziel obserwację od ograniczenia. Obserwacja opisuje wynik. Ograniczenie wyjaśnia, czego z tych danych nie można stwierdzić.

```
podsumowanie = u07["temperatura"].agg(["count", "min", "mean", "max"])
print(podsumowanie)

FIGURES_DIR.mkdir(parents=True, exist_ok=True)
```

```
fig, ax = plt.subplots()
ax.plot(u07["czas"], u07["temperatura"])
ax.set_title("Temperatura U-07 w czasie")
ax.set_xlabel("Czas")
ax.set_ylabel("Temperatura [°C]")
fig.savefig(FIGURES_DIR / "M1_temperatura_U07.png", dpi=150, bbox_inches="tight")
```

Przykład bezpiecznej obserwacji: w analizowanym okresie wykres pokazuje zmiany temperatury urządzenia U-07, a podsumowanie podaje jej zakres i średnią. Przykład ograniczenia: wynik nie wskazuje przyczyny zmian, nie potwierdza awarii i nie opisuje pozostałych urządzeń.

8. **Krok 8 Sprawdź odtwarzalność —** Widoczny wynik nie wystarcza, jeśli powstał dzięki pamięci wcześniejszej sesji.

- Zapisz notebook, uruchom Restart Kernel and Run All Cells, a następnie zatrzymaj się przy pierwszym błędzie.
- Sprawdź, czy po czystym przebiegu numery In rosną od góry do dołu, a widoczne wyniki odpowiadają aktualnemu kodowi.
- Jeżeli wykonujesz operację losową, ustaw random_state. Zapisz użyte wersje Pythona i najważniejszych bibliotek w sekcji Importy i konfiguracja lub requirements.txt.

9. **Krok 9 Udokumentuj użycie LLM —** LLM może pomóc w planowaniu lub wyjaśnieniu kodu, ale jego odpowiedź jest propozycją, nie dowodem.

- Przekaż minimalny kontekst: cel, nazwy kolumn, krótki fragment kodu albo zweryfikowane podsumowanie. Nie przesyłaj całych plików ani danych wrażliwych bez potrzeby.
- W promptcie określ cel, kontekst, ograniczenia i oczekiwany format odpowiedzi.
- W docs/rejestr_uzycia_LLM.csv zapisz cel, prompt, odpowiedź modelu, test weryfikacyjny, werdykt i uzasadnienie. Jeśli LLM nie był używany, wpisz taką informację w sekcji Rejestr użycia LLM notebooka.

```
Pracuję w Jupyter Notebook. Pytanie: jak zmieniała się temperatura U-07?
Dostępne kolumny: czas, urzadzenie, temperatura.
Zaproponuj maksymalnie sześć kroków analizy. Nie twórz liczb ani wniosków.
Przy każdym kroku podaj sposób sprawdzenia wyniku.
```

Punkty kontrolne workflow

Po etapie	Masz sprawdzić	Działanie gdy wynik nie pasuje
Wczytanie	Czy plik istnieje, ma oczekiwany rozmiar i potrzebne kolumny	Sprawdź katalog roboczy i zbudowaną ścieżkę, nie wpisuj ścieżki bezwzględnej.
Kontrola danych	Czy braki, duplikaty lub typy wpływają na pytanie	Zatrzymaj się, opisz problem i nie usuwaj rekordów automatycznie.
Przygotowanie	Czy podzbiór zawiera tylko wybrane urządzenie i jest uporządkowany	Wróć do filtra, konwersji daty lub sortowania.
Wynik	Czy wykres ma tytuł, osie i jednostkę oraz odpowiada na pytanie	Popraw kod lub opis, nie zmieniaj ręcznie obrazu wyniku.
Odtwarzalność	Czy Restart and Run All tworzy wynik od początku	Napraw pierwszy błąd i powtórz pełny przebieg.
LLM	Czy sugestia zgadza się z kodem, danymi lub dokumentacją	Przyjmij, zmodyfikuj albo odrzuć każdy fragment osobno.

Nazewnictwo plików

Stosuj małe litery w nazwach katalogów oraz podkreślenia w nazwach plików. Nie używaj spacji, polskich znaków ani słów final, final2 i ostateczny. Stały prefiks M1 pozwala od razu rozpoznać etap kursu.

Stosuj	Nie stosuj
M1_szkielet_analizy.ipynb	M1_notatnik_start.ipynb, finalny_notatnik.ipynb
M1_temperatura_U07.png	wykres.png, wykres_final2.png
M1_podsumowanie_temperatury_U07.csv	dane_nowe.csv, tabela_ostateczna.csv
rejestr_uzycia_LLM.csv	llm.csv, nowy_rejestr.csv

W tej karcie `data/raw` oznacza niezmienione dane wejściowe, `data/processed` dane utworzone kodem, a `outputs` pliki wynikowe. Te same nazwy są używane w kolejnych instrukcjach modułowych.

Lista kontroli przed zakończeniem modułu

- □ Projekt ma nazwę i strukturę zgodną z kartą pracy.
- □ Pliki w data/raw zachowały oryginalne nazwy i nie zostały zmodyfikowane.
- □ Roboczy notebook nazywa się notebooks/M1_szkielet_analizy.ipynb.
- □ Notebook zawiera pełny szkielet sekcji, pytanie, źródło danych i odbiorcę wyniku.
- □ Kod sprawdza skutek filtrowania, konwersji czasu i sortowania.
- □ Wykres ma tytuł, opisane osie i jednostkę temperatury.
- □ Obserwacja opisuje wynik, a ograniczenie nie udaje wyjaśnienia przyczynowego.
- □ Restart Kernel and Run All Cells zakończył się bez ręcznego uruchamiania komórek poza kolejnością.
- □ Użycie LLM zostało udokumentowane i zweryfikowane albo w notebooku zapisano, że LLM nie był używany.

Gdy pojawi się problem

Objaw	Najpierw sprawdź
FileNotFoundError	Path.cwd(), pełną zbudowaną ścieżkę oraz plik.exists().
NameError po restarcie	Czy zmienna, import albo funkcja powstają przed pierwszym użyciem.
Wynik nie pasuje do kodu	Uruchom ponownie tę komórkę, a na końcu wykonaj Restart and Run All.
Sugestia LLM brzmi przekonująco	Czy wynika z danych i kodu, czy tylko z płynnego tekstu modelu.

Zasada końcowa: najpierw zapisz aktualny notebook, potem wykonaj Restart Kernel and Run All Cells, sprawdź wynik i dopiero wtedy oddaj rezultaty modułu.