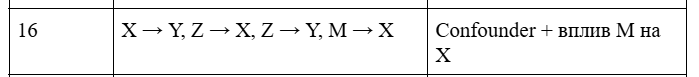
Ровний Григорій Олександрович

Варіант № 16

Лабораторна робота №1

Побудова та аналіз причинних структур на Python



|  |  |
| --- | --- |
| Завдання 1. Побудовати  причинний граф. Ознайомитись із бібліотекою networkx або dowhy для побудови графів; Створити простий причинний граф відповідно до причинної структури за варіантом; Візуалізувати граф. | |
| # X → Y, Z → X, Z → Y, M → X; Confounder + вплив M на X  import networkx as nx import matplotlib.pyplot as plt  G = nx.DiGraph() edges = [("X", "Y"), ("Z", "X"), ("Z", "Y"), ("M", "X")] G.add\_edges\_from(edges)  pos = nx.spring\_layout(G, seed=42) nx.draw(G, pos, with\_labels=True, node\_size=2000, node\_color="lightblue", arrowsize=20, font\_size=12) plt.title("Причинна структура (Варіант 16)") plt.show() | Скрін: |
| Завдання 2. Згенерувати дані відповідно до причинної структури та записати дані у таблицю (DataFrame) | |
| import numpy as np import pandas as pd  np.random.seed(42) N = 1000  # Confounder Z Z = np.random.normal(0, 1, N)  # Variable M (вплив на X) M = np.random.normal(0, 1, N)  # Treatment X залежить від Z і M X = 0.5 \* Z + 0.8 \* M + np.random.normal(0, 1, N)  # Outcome Y залежить від X і Z Y = 1.2 \* X + 0.7 \* Z + np.random.normal(0, 1, N)  data = pd.DataFrame({"Z": Z, "M": M, "X": X, "Y": Y}) print(data.head()) | |
| Завдання 3. Знайти оцінку ефекту методом ML. Використати лінійну регресію та Random Forest для передбачення Y за X; Обчислити середнє зміщення прогнозу при зміні X на одиницю; Зробити висновок по результатам ML-моделей. | |
| import numpy as np import pandas as pd from sklearn.linear\_model import LinearRegression from sklearn.ensemble import RandomForestRegressor  np.random.seed(42) N = 1000  Z = np.random.normal(0, 1, N) M = np.random.normal(0, 1, N) X = 0.5 \* Z + 0.8 \* M + np.random.normal(0, 1, N) Y = 1.2 \* X + 0.7 \* Z + np.random.normal(0, 1, N)  data = pd.DataFrame({"Z": Z, "M": M, "X": X, "Y": Y})  # ML-підхід: прогнозуємо Y за X X\_ml = data[["X"]] y\_ml = data["Y"]  # Лінійна регресія linreg = LinearRegression().fit(X\_ml, y\_ml) lin\_coef = linreg.coef\_[0] print(f"Оцінка ефекту (Linear Regression): {lin\_coef:.3f}")  # Random Forest rf = RandomForestRegressor(n\_estimators=100, random\_state=42).fit(X\_ml, y\_ml)  # обчислимо середню зміну Y при X+1 x\_plus1 = X\_ml + 1 rf\_pred\_diff = rf.predict(x\_plus1) - rf.predict(X\_ml) print(f"Оцінка ефекту (Random Forest, середнє): {rf\_pred\_diff.mean():.3f}") | |
| Завдання 4. Знайти оцінку ефекту методом Causal AI. Використати бібліотеку DoWhy; Задати причинну модель у вигляді графа; Оцінити середній причинний ефект (ATE) змінної X на Y. Порівняти результат з ML-оцінкою. | |
| import numpy as np import pandas as pd from dowhy import CausalModel  np.random.seed(42) N = 1000  # Confounder Z Z = np.random.normal(0, 1, N)  # Variable M (вплив на X) M = np.random.normal(0, 1, N)  # Treatment X залежить від Z і M X = 0.5 \* Z + 0.8 \* M + np.random.normal(0, 1, N)  # Outcome Y залежить від X і Z Y = 1.2 \* X + 0.7 \* Z + np.random.normal(0, 1, N)  data = pd.DataFrame({"Z": Z, "M": M, "X": X, "Y": Y}) print(data.head())   # описуємо граф у вигляді causal graph causal\_graph = """ digraph {  Z -> X;  Z -> Y;  M -> X;  X -> Y; } """  # створюємо модель model = CausalModel(  data=data,  treatment="X",  outcome="Y",  graph=causal\_graph )  model.view\_model()  identified\_estimand = model.identify\_effect() estimate = model.estimate\_effect(  identified\_estimand,  method\_name="backdoor.linear\_regression" ) print(f"ATE (DoWhy): {estimate.value:.3f}") | |
| **Висновок по результатам ML-моделей**  Лінійна регресія дала нам значення коефіцієнта для X, яке відображає кореляцію, а не справжній причинний ефект, а в свою чергу Random Forest також оцінив середню зміну Y при збільшенні X на 1, але його результат теж є лише прогнозним ефектом, що не враховує вплив змінних Z та M. Отож обидві ML-моделі не розрізняють кореляцію та каузальність, тому їх оцінки є зміщеними. В свою чергу dowhy побудував причинний граф і за допомогою backdoor-критерію врахував confounder Z. | |