HW3_complete_assignment

February 13, 2026

1 درس هوش مصنوعی قابل اعتماد | Trusted Artificial Intelligence

1.1 تمرین شماره ۳ | Homework 3

1.1.1 نوتبوک کامل و قابل اجرا (Q1 تا Q6)

دانشجو / Student: Taha Majlesi (810101504)
دانشگاه / University: University of Tehran, ECE Department
مدرس / Instructor: Dr. Mostafa Tavasolipour

این نوتبوک برای تصحیح نهایی طراحی شده است: - ترتیب دقیق سوالات مطابق صورت تمرین - برچسب‌های
اجرای - گزارش خوانایی برای فارسی/انگلیسی دوزبانه متن - (**زیربخش** و **سوال**) تمرین قالب با یکسان
ذخیره‌شده) خروجی‌های + ثابت seed) بازتولیدپذیر

1.2 نقشه نمره‌دهی | Grading Map

Question	بخش	Score
سوال اول	Observational vs Interventional Probability	10
سوال دوم	Causal Recourse for Two Individuals	12
سوال سوم	Airline SCM Graph + Modeling + Variance Analysis	20
سوال چهارم	Insulin Causal Effect Estimation	22
سوال پنجم	Complete Causal Recourse Pipeline	20
سوال ششم	Theory from Robust Causal Recourse Paper	16

Total: 100

1.3 تنظیمات اولیه و بازتولیدپذیری | Setup and Reproducibility

```
from __future__ import annotations

import os
import sys
import math
import json
import random
import subprocess
from pathlib import Path
```

```
import numpy as np
import pandas as pd
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import seaborn as sns
import torch
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.preprocessing import StandardScaler

SEED = 0
np.random.seed(SEED)
random.seed(SEED)
torch.manual_seed(SEED)

sns.set_theme(style='whitegrid')

# Resolve project root robustly.
ROOT = Path.cwd().resolve()
while ROOT != ROOT.parent and not (ROOT / 'description' / 'HW3_TAI.pdf').exists():
    ROOT = ROOT.parent
if not (ROOT / 'description' / 'HW3_TAI.pdf').exists():
    raise RuntimeError('Could not locate HW3 project root from current working directory.')

Q5_DIR_CANDIDATES = [ROOT / 'code' / 'q5_codes', ROOT / 'code' / 'Q5_codes']
Q5_DIR = next((p for p in Q5_DIR_CANDIDATES if p.exists()), None)
if Q5_DIR is None:
    raise RuntimeError('Could not locate q5_codes directory.')

if str(Q5_DIR) not in sys.path:
    sys.path.append(str(Q5_DIR))

import data_utils
import recourse
import trainers
import utils
import train_classifiers

DATASET_DIR = ROOT / 'dataset'
OUT_DIR = ROOT / 'output' / 'jupyter-notebook' / 'artifacts'
OUT_DIR.mkdir(parents=True, exist_ok=True)

print('ROOT:', ROOT)
print('Q5_DIR:', Q5_DIR)
print('DATASET_DIR:', DATASET_DIR)
print('Health source:', data_utils.get_health_source_path())
print('Health source tag:', data_utils.get_health_source_tag())
```

1.4 سوال اول (۱۰ نمره) | Question 1 (10 Points)

DAG: (S o A), (S o Y), (A o Y) with the exact probabilities from the assignment PDF.

1.4.1 زیربخش اول (۵ نمره)

محاسبه‌ی:- (P_X(Y=1□A=N)) - (P_X(Y=1□A=O))

Compute observational conditionals using Bayes + total probability.

```
# Q1 constants extracted from the assignment PDF.
pS_L = 0.49
pS_R = 1 - pS_L

pA_N_given_S = {'L': 0.77, 'R': 0.24}
pA_O_given_S = {'L': 1 - pA_N_given_S['L'], 'R': 1 - pA_N_given_S['R']}

pY1_given_SA = {
    ('L', 'N'): 0.73,
    ('L', 'O'): 0.69,
    ('R', 'N'): 0.93,
    ('R', 'O'): 0.87,
}

# Marginals for A
pA_N = pA_N_given_S['L'] * pS_L + pA_N_given_S['R'] * pS_R
pA_O = 1 - pA_N

# Bayes terms for observational conditionals
pS_L_given_A_N = (pA_N_given_S['L'] * pS_L) / pA_N
pS_R_given_A_N = 1 - pS_L_given_A_N

pS_L_given_A_O = (pA_O_given_S['L'] * pS_L) / pA_O
pS_R_given_A_O = 1 - pS_L_given_A_O

# Observational conditionals
pY1_given_A_N = (
    pY1_given_SA[('L', 'N')] * pS_L_given_A_N
    + pY1_given_SA[('R', 'N')] * pS_R_given_A_N
)
pY1_given_A_O = (
    pY1_given_SA[('L', 'O')] * pS_L_given_A_O
    + pY1_given_SA[('R', 'O')] * pS_R_given_A_O
)

# Interventional conditionals: cut incoming edges to A
pY1_given_do_A_N = (
    pY1_given_SA[('L', 'N')] * pS_L
    + pY1_given_SA[('R', 'N')] * pS_R
```