Module 3 Practice Sheet

Topics covered:

1. Standardization, Min-Max scaling, Robust scaling
2. Nominal vs ordinal variables, one-hot vs ordinal encoding
3. Vectors, dot product, norms, Euclidean and Manhattan distance

Part A. Quick basics

A1. Spot the right scaler

For each feature, pick one scaler and justify in one line.
a) Apartment_price_BDT with a few luxury penthouses
b) Skin_temperature_C measured from a wearable between 30 and 36
c) Daily_app_opens with many zeros and a few power users

A2. Manual Min-Max on a tiny set

Given scores = [20, 25, 30, 50], scale to [0, 1] by hand. Show each step.

A3. Z-scores on a subset

Given x = [8, 9, 11], compute mean, standard deviation, then standardize each. Use population standard deviation for this question.

A4. Robust scaling ingredients

Given y = [5, 6, 6, 7, 50], find median, Q1, Q3, IQR. Do not scale yet.

A5. Nominal or ordinal

Mark each as nominal or ordinal.
a) T-shirt_size {S, M, L, XL}
b) City {Dhaka, Chattogram, Rajshahi}
c) Satisfaction {Low, Medium, High}

Part B. Hands on practice

B1. Three scalers side by side

Heights = [150, 160, 170, 175, 180]
Weights = [58, 62, 65, 66, 190]
Tasks:
a) Min-Max scale both to [0, 1]
b) Standardize the first three values of each only
c) Robust scale Weights with median and IQR
d) One line on which scaler handles the outlier best

B2. One-hot by hand

Cities = [Dhaka, Chattogram, Dhaka, Rajshahi, Rajshahi]
Create three columns City_Dhaka, City_Chattogram, City_Rajshahi using 0 and 1.

B3. Ordinal mapping

Education = [High School, Bachelor, Master, Bachelor, Master]
Map with High School=0, Bachelor=1, Master=2.
Then change the map to High School=1, Bachelor=2, Master=3 and explain in one line how this shifts distances.

B4. Encoding mixup [Optional]

You mistakenly apply ordinal encoding to City and one-hot to Education. Write one sentence on the risk this creates in a linear model.

B5. Vectors and alignment [Optional]

a = [3, −1, 2], b = [4, 0, −2], c = [−6, 2, −4]
Tasks:
a) Compute a·b and a·c
b) Compare signs and magnitudes to comment on the alignment of a with b and with c
c) L2 normalize a and give the normalized vector to three decimals

B6. Two distances, different vibes

Points: P1(2, 3), P2(5, 7), P3(2, 10)
Tasks:
a) Compute Euclidean and Manhattan distances for all pairs
b) Which distance is more sensitive to a single large jump in one coordinate
c) Scale y by 10 and recompute d(P1, P2) for both distances, then explain the effect in one line

Part C. Mini datasets

Use these two tables for C-tasks.

C-Data-1

ID	Age	Hours_Study	GPA	Internet	City
1	20	1.0	3.10	Yes	Dhaka
2	21	0.5	2.60	No	Chattogram
3	22	2.2	3.40	Yes	Rajshahi
4	20	5.0	3.90	Yes	Dhaka
5	23	0.2	2.30	No	Rajshahi

C-Data-2

ID	Income_BDT	Transactions	Temp_C	Education	Satisfaction
1	30000	0	25.0	High School	Low
2	45000	1	26.0	Bachelor	Medium
3	52000	2	24.5	Master	High
4	300000	12	28.0	Bachelor	Medium
5	38000	0	25.5	Master	Medium

C1. Scaler choices with evidence

Pick a scaler for Income_BDT, Transactions, Temp_C. For each, give a one line justification and a two-line numeric illustration using C-Data-2 values.

C2. Mixed preprocessing plan

For C-Data-1 and C-Data-2 combined:
a) Identify nominal and ordinal columns
b) Propose one encoding plan listing exact columns to one-hot vs ordinal
c) Propose one scaling plan listing exact columns to Min-Max vs Standardization vs Robust

C3. Outlier stress test [Optional]

Using Income_BDT in C-Data-2, compute Min-Max scaled values. Then compute Robust scaled values. In one line, compare how each treats the 300000 outlier.

C4. Distance on feature space [Optional]

From C-Data-1, take feature pair (Hours_Study, GPA).
a) Compute the Euclidean distance between ID 1 and ID 4
b) Compute the Manhattan distance for the same pair
c) Normalize Hours_Study and GPA with Min-Max and recompute both distances, then comment in one line on scale effects

Part D. Mini project [Optional]

Goal:
Make one notebook that shows **encoding + scaling + distance change**. No train–test split, no models.

Step 1: Create a small DataFrame

- Manually create a pandas DataFrame with:
    - 3–4 numeric columns (like Income, Hours_Study, GPA)
    - 1–2 nominal columns (like City, Internet)
    - 1–2 ordinal columns (like Education_Level, Satisfaction)

Step 2: Decide preprocessing plan (short markdown cell)

- Write:
    - Which columns will be **one-hot encoded**
    - Which columns will be **ordinal encoded** (with mapping)

    - Which numeric columns will use **Standardization**, **Min–Max**, or **Robust**

Step 3: Apply ColumnTransformer

- Use `ColumnTransformer` to:
    - One-hot encode nominal columns
    - Ordinal encode ordinal columns
    - Scale numeric columns using your chosen scaler(s)
- Show the transformed array (shape + first few rows).

Step 4: Distance before vs after scaling

Pick **two numeric columns**, for example (Income, Transactions_7d):

1. Take 3 rows only, call them P1, P2, P3.
2. Compute Euclidean and Manhattan distances between them **before scaling**.
3. Apply **two different scalers** to these two columns (for example Standard vs Robust).
4. Recompute the distances after each scaler.
5. Put results in a tiny table in markdown.

Step 5: Short reflection

In 3–4 sentences:

- Which scaler handled outliers better for your chosen features?
- Did scaling change which points are “closer” to each other?
- Why does this matter for algorithms that use distance?