

Week 1 – Problem Statement & Dataset Exploration (Cardiovascular Disease Prediction)

In this week, we define the problem clearly and perform initial exploration of the dataset. We aim to understand the structure, quality, and basic statistical properties of the Cardiovascular Disease dataset.

Student Information

Name: Vishal Baraiya

Enrollment No.: 23010101014

Roll No.: C3-635

Course: Machine Learning & Deep Learning Project

Problem Definition

Cardiovascular disease (CVD) is one of the leading causes of mortality worldwide. Early detection can significantly reduce the risk of fatal outcomes.

In this project, our goal is to **predict whether a patient has cardiovascular disease** based on clinical and lifestyle-related features such as:

- Age
 - Gender
 - Height & Weight (BMI)
 - Blood Pressure (ap_hi, ap_lo)
 - Cholesterol level
 - Glucose
 - Smoking habits
 - Alcohol intake
 - Physical activity
-

Objective

Build a classification model that predicts:

- 0 → No cardiovascular disease
- 1 → Cardiovascular disease present

Dataset Description (Kaggle – Cardiovascular Disease Dataset)

This dataset contains 70,000 medical records of patients with the following features:

Feature	Description
id	Unique identifier for each patient record (not used for prediction)
age	Age of the patient in days — will be converted to years in Week 2
gender	Gender of the patient (1 = female, 2 = male)
height	Height in centimeters
weight	Weight in kilograms
ap_hi	Systolic blood pressure
ap_lo	Diastolic blood pressure
cholesterol	1 = normal, 2 = above normal, 3 = well above normal
gluc	1 = normal, 2 = above normal, 3 = well above normal
smoke	0 = non-smoker, 1 = smoker
alco	0 = does not consume alcohol, 1 = consumes alcohol
active	0 = not physically active, 1 = physically active
cardio	Target variable — 0 = no cardiovascular disease, 1 = has disease

1. Import Libraries

```
In [4]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid") # set the visual style of all Seaborn plots
```

2. Load Dataset

Dataset file: `cardio_train.csv` (semicolon-separated ;)

```
In [5]: df = pd.read_csv("../data/raw/cardio_train.csv", sep=';')
df.head()
```

Out[5]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active
0	0	18393	2	168	62.0	110	80	1	1	0	0	1
1	1	20228	1	156	85.0	140	90	3	1	0	0	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0
3	3	17623	2	169	82.0	150	100	1	1	0	0	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0

3. Basic Dataset Info

- Column names
- Data types
- Missing values status
- Basic descriptive statistics

Info()

```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   id          70000 non-null   int64  
 1   age         70000 non-null   int64  
 2   gender      70000 non-null   int64  
 3   height      70000 non-null   int64  
 4   weight      70000 non-null   float64 
 5   ap_hi       70000 non-null   int64  
 6   ap_lo       70000 non-null   int64  
 7   cholesterol 70000 non-null   int64  
 8   gluc        70000 non-null   int64  
 9   smoke       70000 non-null   int64  
 10  alco        70000 non-null   int64  
 11  active      70000 non-null   int64  
 12  cardio     70000 non-null   int64  
dtypes: float64(1), int64(12)
memory usage: 6.9 MB
```

Describe()

In [7]: `df.describe()`

	id	age	gender	height	weight	ap_hi
count	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000
mean	49972.419900	19468.865814	1.349571	164.359229	74.205690	128.817286
std	28851.302323	2467.251667	0.476838	8.210126	14.395757	154.011419
min	0.000000	10798.000000	1.000000	55.000000	10.000000	-150.000000
25%	25006.750000	17664.000000	1.000000	159.000000	65.000000	120.000000
50%	50001.500000	19703.000000	1.000000	165.000000	72.000000	120.000000
75%	74889.250000	21327.000000	2.000000	170.000000	82.000000	140.000000
max	99999.000000	23713.000000	2.000000	250.000000	200.000000	16020.000000



4. Check Missing Values

In [8]: `df.isnull().sum()`

```
Out[8]: id      0
        age     0
        gender  0
        height  0
        weight  0
        ap_hi   0
        ap_lo   0
        cholesterol  0
        gluc    0
        smoke   0
        alco    0
        active  0
        cardio  0
        dtype: int64
```

5. Target Variable Distribution

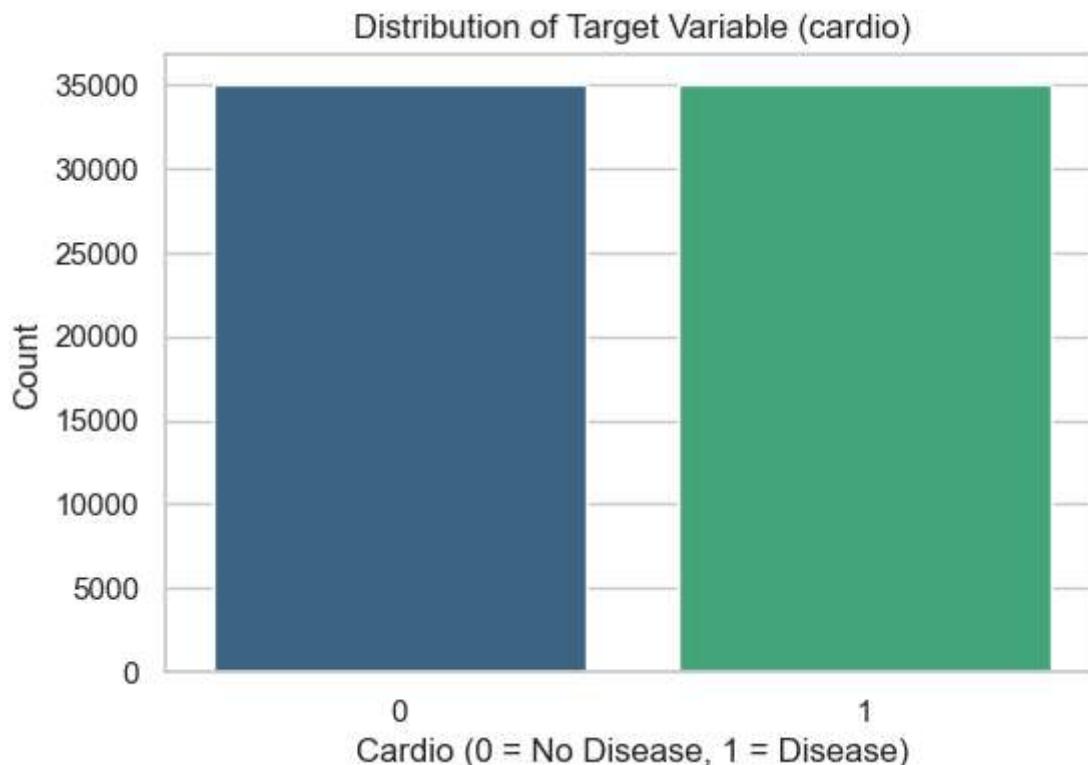
Target Variable Distribution (0 vs 1)

Understanding class balance is crucial for classification.

In [9]: `df['cardio'].value_counts()`

```
Out[9]: cardio
0    35021
1    34979
Name: count, dtype: int64
```

```
In [11]: plt.figure(figsize=(6,4))
sns.countplot(data=df, x='cardio', hue='cardio', palette="viridis", legend=False)
plt.title("Distribution of Target Variable (cardio)")
plt.xlabel("Cardio (0 = No Disease, 1 = Disease)")
plt.ylabel("Count")
plt.show()
```

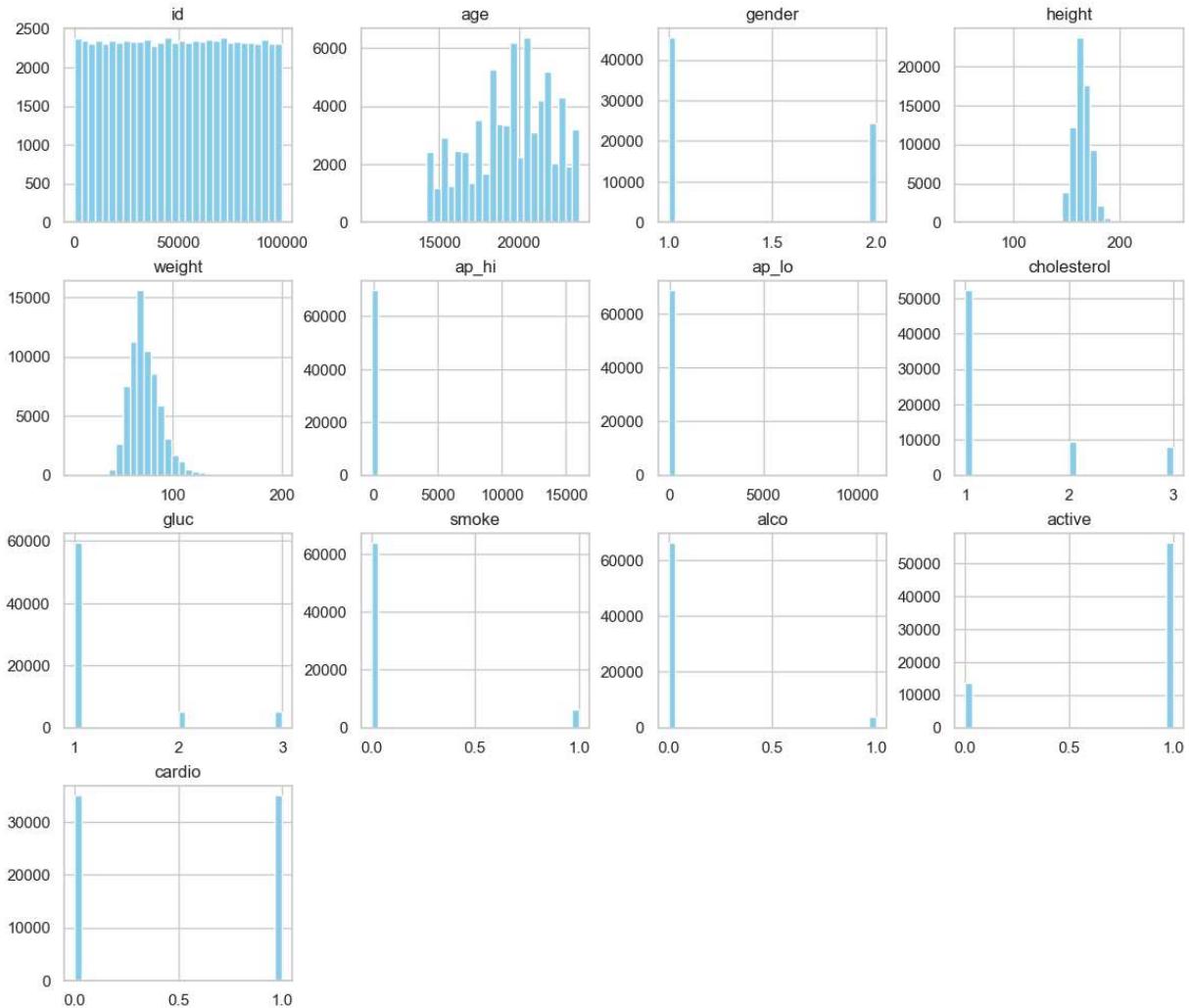


6. Basic Feature Exploration

- Numerical distributions
- Possible outliers
- Basic trends

```
In [12]: df.hist(figsize=(14, 12), bins=30, color='skyblue')
plt.suptitle("Feature Distributions - Before Cleaning", fontsize=16)
plt.show()
```

Feature Distributions – Before Cleaning



7. Initial Observations

Based on the initial exploration of the dataset, the following observations can be made:

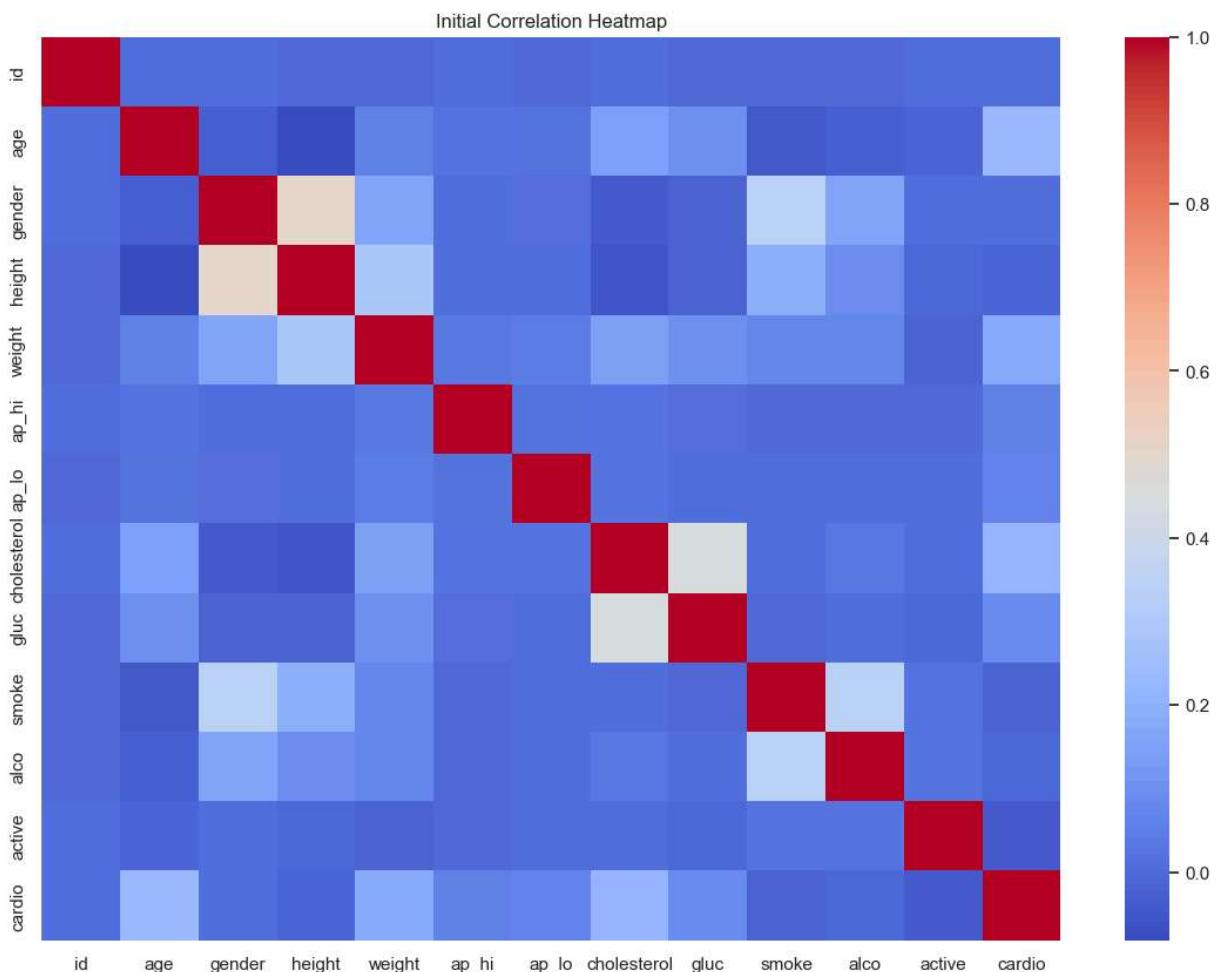
- The dataset **does not contain any missing values**, as all columns show a count of 70,000 entries.
- The **target variable (cardio) appears moderately balanced**, with both classes (0 = No CVD, 1 = CVD) present in comparable proportions.
- Some features show **potential outliers**, particularly:
 - `ap_hi` (systolic BP)
 - `ap_lo` (diastolic BP)
 - `height` and `weight`
- The `age` column is stored in **days**, not years, and will need to be converted into years during Week 2 preprocessing.

- Histograms indicate that several features have **non-normal distributions** and may require scaling or transformation.
- Medical indicators such as `cholesterol` and `gluc` use **three-level categorical encoding** (1, 2, 3), which will be handled during preprocessing.

8. Correlation Heatmap (Initial)

To understand relationships between features.

```
In [14]: plt.figure(figsize=(14,10))
sns.heatmap(df.corr(), cmap="coolwarm", annot=False)
plt.title("Initial Correlation Heatmap")
plt.show()
```



Week 1 Completed Successfully

- ✓ Defined the ML problem
- ✓ Understood dataset structure

- ✓ Explored distributions of features
- ✓ Checked missing values
- ✓ Visualized initial correlations
- ✓ Identified cleaning needs for Week 2

In []: