Lab - 1

05/03/25

import pandas as pd.

i) df = pd.read_csv("housing.csv")

ii) df.info()

iii) df.describe()

iv) df["Ocean Proximity"].value_counts()

v) mis_val = df.isnull().sum()
val = mis_val[mis_val > 0]
print(val)

Diabetes

import pandas as pd.

import numpy as np
from Sklearn preprocessing import MinMaxScaler, StandardScaler

from Sklearn.impute import SimpleImputer
from Sklearn.preprocessing import LabelEncoder

df = pd.read_csv("Dataset of Diabetes.csv")

print(df.head())

# missing values
print(df.isnull().sum())

# impute.
nc = df.select_dtypes(include=["float64", "int64"])
imputer = SimpleImputer(strategy = 'mean'), edunt

```
df[nc] = imputer_num.fit_transform(df[nc])
cat_c = df.select_dtypes(include=['object']).columns
imputer_cat = SimpleImputer(strategy='most_frequent')
df[cat_c] = imputer_cat.fit_transform(df[cat_c])
```

Handling categorical data.

```
label_encoder = LabelEncoder()
df['Gender'] = label_encoder.fit_transform(df['Gender'])
df['CLASS'] = label_encoder.fit_transform(df['CLASS'])

Q1 = df[num_columns].quantile(0.25)
Q3 = df[num_columns].quantile(0.75)

IQR = Q3 - Q1
df_clean = df[~(df[nc] < (Q1 - 1.5 * IQR) |
                (df[nc] > (Q3 + 1.5 * IQR))).
                any(axis=1)]

Scaler_choice = 'minmax'

Scaler = MinMaxScaler()

df_scaled = pd.DataFrame(Scaler.fit_transform(df_clean
            [nc]), (dtype=nc)

df_final = pd.concat([df_clean[cat_c], df_scaled],
                     axis=1)
```

1. No columns.
2. Gender and CLASS
3. Min-Max-Scaling
   i) fixed range for features, bounded range.
   ii) No. outliers.

Standard Scaler,
* outliers exist
* Normal distributed data