Lab-1

```
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('/content/Dataset of Diabetes.csv')

print(df.head())

print(df.isnull().sum())

nc = df.select_dtypes(include=['float64','int64']
                      ).columns
imputer = SimpleImputer(strategy='mean')
df[nc] = imputer.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]), columns=nc)

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

1. No columns
2. Gender and class
3. Min-max scaling
   if fixed range for feature - bounded range
   no outliers

Standard scaler
   Outliers exist
   normal distributed data