Lab: AI & ML Lab (23A31403)

College: Andhra Engineering College

8. Apply the following Pre-processing techniques for a given dataset.
a. Attribute selection
b. Handling Missing Values
c. Discretization
d. Elimination of Outliers

Program :

```
import pandas as pd
data = {
   'roll_number':['501','503','509','518','520','524','538','544','510','511','517','532'],

'Name':['HEMANTHKUMAR','MANEEL','PRAKASH','VENKATASUBBAIAH','JAYA','NANI','GOPI','
MANOJ','ANKESWARAMMA','DHARANI','PRAVEEN','PRAKASH'],
      # 'Passed':[1,3,1,1,3,1,2,3,6,6,6,5],
      # 'Failed':[5,3,5,5,3,5,4,3,0,0,0,1],
      'Passed':[1,3,1,1,None,1,2,3,6,6,6,5],
      'Failed':[None,3,5,5,3,None,4,3,0,0,0,1],
      'Total':[6,6,6,6,6,6,6,6,6,6,6,6],

'Branch':['AIML','AIML','AIML','AIML','AIML','AIML','AIML','AIML','AIML','AIML','AIML','AIML']
,
      'Age':[22,20,21,20,23,19,20,23,22,21,20,None]
}

df = pd.DataFrame(data)
print(df)
```

Output:

roll_number	Name	Passed	Failed	Total	Branch	Age
0	501	HEMANTHKUMAR	1.0	NaN	6	AIML	22.0
1	503	MANEEL	3.0	3.0	6	AIML	20.0
2	509	PRAKASH	1.0	5.0	6	AIML	21.0
3	518	VENKATASUBBAIAH	1.0	5.0	6	AIML	20.0
4	520	JAYA	NaN	3.0	6	AIML	23.0
5	524	NANI	1.0	NaN	6	AIML	19.0
6	538	GOPI	2.0	4.0	6	AIML	20.0
7	544	MANOJ	3.0	3.0	6	AIML	23.0
8	510	ANKESWARAMMA	6.0	0.0	6	AIML	22.0
9	511	DHARANI	6.0	0.0	6	AIML	21.0
10	517	PRAVEEN	6.0	0.0	6	AIML	20.0

11	532	PRAKASH	5.0	1.0	6	AIML	NaN

a. Attribute selection

Program:

```
df1=df[['Age','Passed','Failed','Total','Age']]
df1
```

Output

	Age	Passed	Failed	Total	Age	
0	22.0	1.0	NaN	6	22.0	
1	20.0	3.0	3.0	6	20.0	
2	21.0	1.0	5.0	6	21.0	
3	20.0	1.0	5.0	6	20.0	
4	23.0	NaN	3.0	6	23.0	
5	19.0	1.0	NaN	6	19.0	
6	20.0	2.0	4.0	6	20.0	
7	23.0	3.0	3.0	6	23.0	
8	22.0	6.0	0.0	6	22.0	
9	21.0	6.0	0.0	6	21.0	
10	20.0	6.0	0.0	6	20.0	
11	NaN	5.0	1.0	6	NaN	

b. Handling Missing Values

```
# Method 1: Remove missing rows
# df=df.dropna()
# df
```

```
df['Age'].fillna(df['Age'].mean(),inplace=True)
df['Passed'].fillna(df['Passed'].mean(),inplace=True)
df['Failed'].fillna(df['Failed'].mean(),inplace=True)
df['Age'].fillna(df['Age'].mode(), inplace=True)
# df['Age'].fillna(df['Age'].mode()[0], inplace=True)
# df['Age'].fillna(df['Age'].mean(), inplace=True)
df
```

Output:

roll_number	Name	Passed	Failed	Total	Branch	Age	
0	501	HEMANTHKUMAR	1.000000	2.4	6	AIML	22.0
1	503	MANEEL	3.000000	3.0	6	AIML	20.0
2	509	PRAKASH	1.000000	5.0	6	AIML	21.0
3	518	VENKATASUBBAIAH	1.000000	5.0	6	AIML	20.0
4	520	JAYA	3.181818	3.0	6	AIML	23.0
5	524	NANI	1.000000	2.4	6	AIML	19.0
6	538	GOPI	2.000000	4.0	6	AIML	20.0
7	544	MANOJ	3.000000	3.0	6	AIML	23.0
8	510	ANKESWARAMMA	6.000000	0.0	6	AIML	22.0
9	511	DHARANI	6.000000	0.0	6	AIML	21.0
10	517	PRAVEEN	6.000000	0.0	6	AIML	20.0
11	532	PRAKASH	5.000000	1.0	6	AIML	21.0

c. Discretization

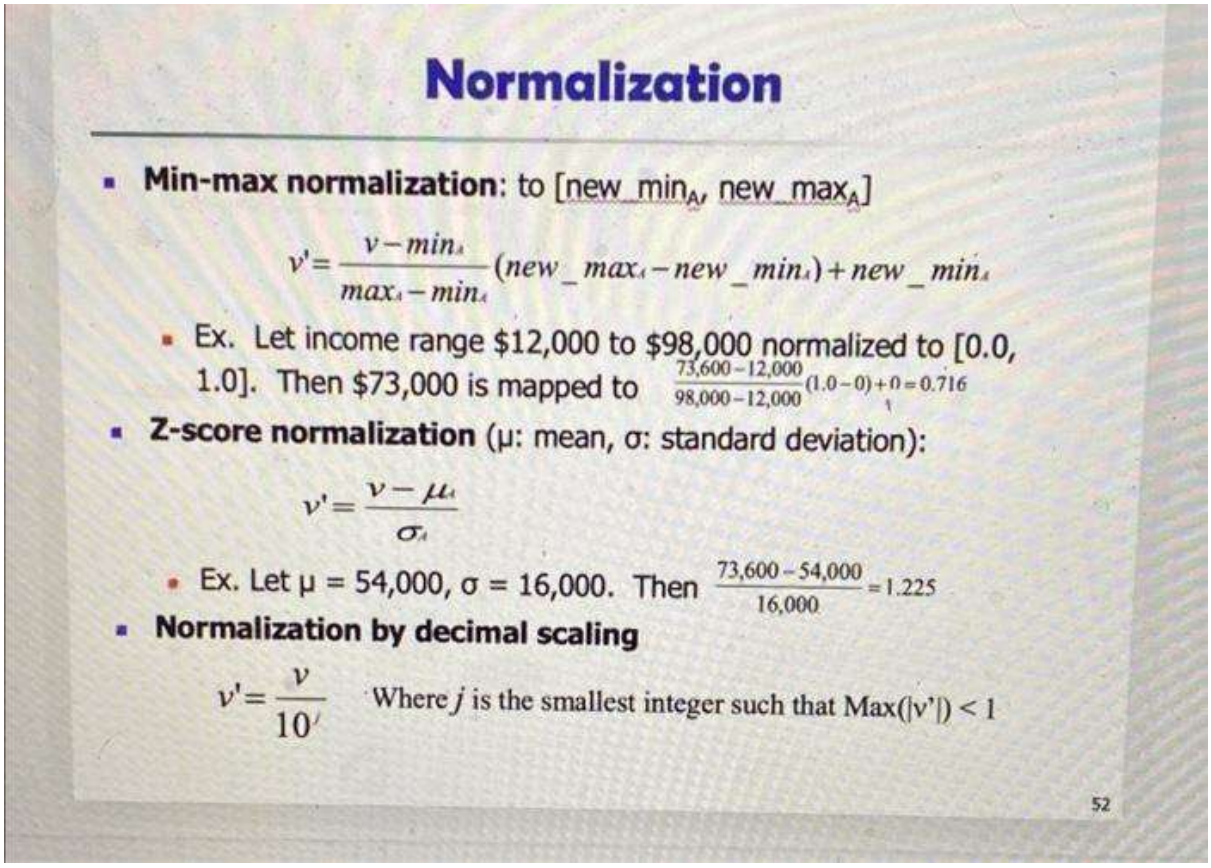
```
# bins = [-0.1, 0, 2, 4, 6]
# labels = ['Excellent', 'Good', 'Average', 'Poor Performance']
# df['Failed_Group'] = pd.cut(df['Failed'], bins=bins, labels=labels,include_lowest=True)
# print(df[['Failed', 'Failed_Group']])

bins_age = [0, 18, 25, 35, 50, 100]
labels_age = ['Minor', 'Young Adult', 'Adult', 'Senior', 'Very Senior']
df['Age_Group']=pd.cut(df['Age'], bins=bins_age, labels=labels_age)
df[['Age', 'Age_Group']]
```

Output:

	Age	Age_Group
0	22.0	Young Adult
1	20.0	Young Adult
2	21.0	Young Adult
3	20.0	Young Adult
4	23.0	Young Adult
5	19.0	Young Adult
6	20.0	Young Adult
7	23.0	Young Adult
8	22.0	Young Adult
9	21.0	Young Adult
10	20.0	Young Adult
11	21.0	Young Adult

d. Elimination of Outliers

Min–Max Normalization (0 to 1) – Correct Method



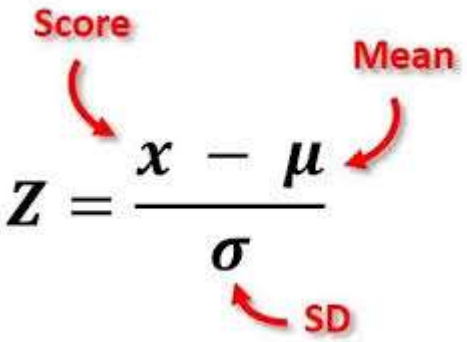
Program

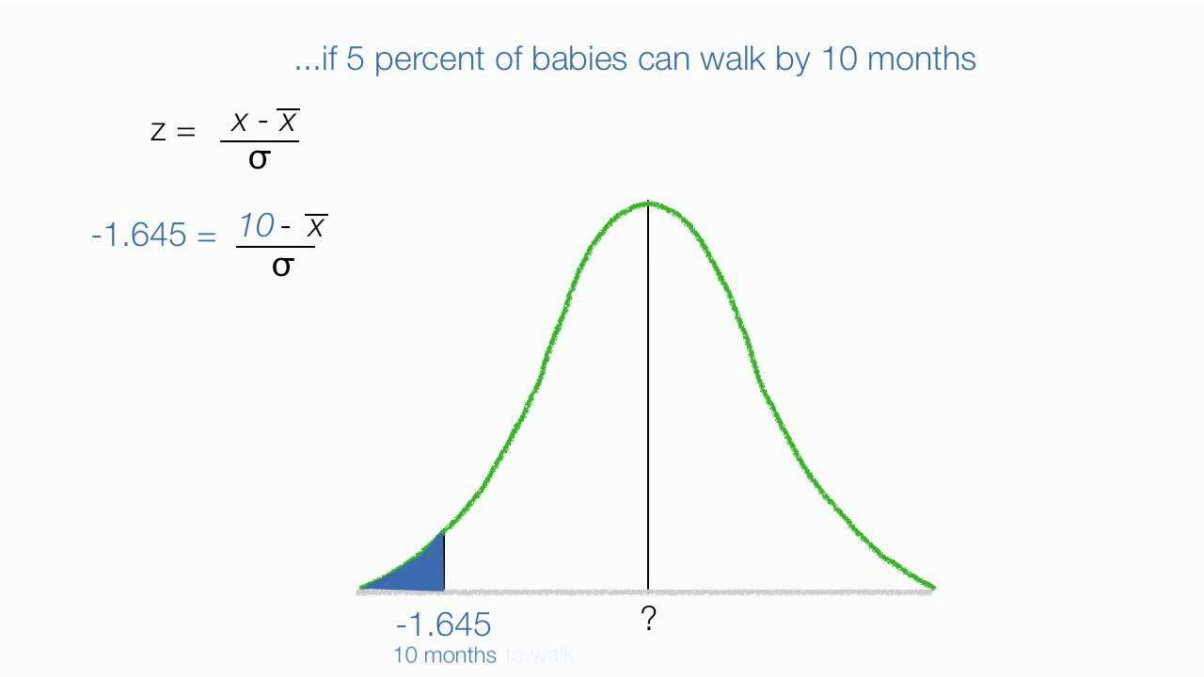
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df[['Age_mm', 'Passed_mm', 'Failed_mm']] = scaler.fit_transform(
   df[['Age', 'Passed', 'Failed']]
)

print(df[['Age', 'Age_mm', 'Passed', 'Passed_mm','Failed','Failed_mm']])
```

Output:

	Age	Age_mm	Passed	Passed_mm	Failed	Failed_mm
0	22.0	0.75	1.000000	0.000000	2.4	0.48
1	20.0	0.25	3.000000	0.400000	3.0	0.60
2	21.0	0.50	1.000000	0.000000	5.0	1.00
3	20.0	0.25	1.000000	0.000000	5.0	1.00
4	23.0	1.00	3.181818	0.436364	3.0	0.60
5	19.0	0.00	1.000000	0.000000	2.4	0.48
6	20.0	0.25	2.000000	0.200000	4.0	0.80
7	23.0	1.00	3.000000	0.400000	3.0	0.60
8	22.0	0.75	6.000000	1.000000	0.0	0.00
9	21.0	0.50	6.000000	1.000000	0.0	0.00
10	20.0	0.25	6.000000	1.000000	0.0	0.00
11	21.0	0.50	5.000000	0.800000	1.0	0.20

Z-Score Normalization (Standardization)





Program

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df[['Age_z', 'Passed_z', 'Failed_z']] = scaler.fit_transform(
   df[['Age', 'Passed', 'Failed']]
)
print(df[['Age', 'Age_z','Passed','Passed_z','Failed', 'Failed_z']])
```

Output

```
Age    Age_z   Passed  Passed_z  Failed  Failed_z
0   22.0  0.816497  1.000000 -1.095065     2.4  0.000000
1   20.0 -0.816497  3.000000 -0.091255     3.0  0.344502
2   21.0  0.000000  1.000000 -1.095065     5.0  1.492840
```

3	20.0	-0.816497	1.000000	-1.095065	5.0	1.492840
4	23.0	1.632993	3.181818	0.000000	3.0	0.344502
5	19.0	-1.632993	1.000000	-1.095065	2.4	0.000000
6	20.0	-0.816497	2.000000	-0.593160	4.0	0.918671
7	23.0	1.632993	3.000000	-0.091255	3.0	0.344502
8	22.0	0.816497	6.000000	1.414459	0.0	-1.378006
9	21.0	0.000000	6.000000	1.414459	0.0	-1.378006
10	20.0	-0.816497	6.000000	1.414459	0.0	-1.378006
11	21.0	0.000000	5.000000	0.912554	1.0	-0.803837