# CA 5314: Practice Exercise 2

## Data Pre-processing

**Aim:**

1. Explore Label Encoder
2. Explore Scikit Pre-processing routines like Scaling
3. Explore Scikit Pre-processing routines like Binarizer

**The variable in the dataset Female and Male can be changed to 0 or 1 using Label Encoder. It is done as given below:**

```
df_gender_encode=LabelEncoder()
```

```
df.gender=df_gender_encode.fit_transform(df.gender)
```

**Scaling can be done as follows:**

```
df.Marks = preprocessing.scale(df.Marks)
```

```
scaled_df= preprocessing.scale(df.Marks)
```

**Scaling removes the mean**

**Binarization uses threshold and converts values to binary as shown below:**

```
scaled_df_bin = preprocessing.Binarizer(threshold=0.5).transform(newarr)
```

**Duplicates can be removed as follows:**

```
df_duplicates_removed = pd.DataFrame.drop_duplicates(df_duplicated)
```

**The NaN of a column can be removed as shown below:**

```
df['m5']=df['m5'].fillna(0)
```

**This removes all the NaN to zero.**

**The command,**

```
df=df.dropna(axis=1)
```

**removes all the columns that has NaN.**

## Catalog 1

```
import pandas as pd
col_list=["id","first","last","gender","Marks","selected"]
df = pd.read_csv("SampleDB.csv",usecols=col_list)
print(df)
print("End of Listing\n\n\n")

# Let us convert the in Gender column, make Female as 0 and
# male as 1 using LabelEncoder in scikitlearn method

from sklearn.preprocessing import LabelEncoder
df_gender_encode=LabelEncoder()
df.gender=df_gender_encode.fit_transform(df.gender)
# One can observe that female is coded as 0 and Male as 1
print(df)
print("End of Listing\n\n\n")

# Now one can scale the marks to remove mean

from sklearn import preprocessing
df.Marks = preprocessing.scale(df.Marks)
scaled_df= preprocessing.scale(df.Marks)
print(df)
print("Scaling of marks is completed\n\n\n\n")


newarr = scaled_df.reshape(-1,1)
scaled_df_bin = preprocessing.Binarizer(threshold=0.5).transform(newarr)
df['Marks']=scaled_df_bin
print(df)
print("Binarizarion of marks is completed\n\n\n\n")
```

## Catalog 2

```
import pandas as pd
col_list=["id","first","last","gender","Marks","selected"]
df = pd.read_csv("sample.csv",usecols=col_list)
print(df)
print("End of Listing\n\n\n")

# Let us create duplicate elements in the given dataset
# This is done using the command concate 2 times as given below

df_duplicated = pd.concat([df]*2, ignore_index=True)
print(df_duplicated)

print("Display before duplication\n\n\n\n")

df_duplicates_removed = pd.DataFrame.drop_duplicates(df_duplicated)
print(df_duplicates_removed)

print("Display after duplication\n\n\n\n")
```

**Catalog 3**

```
import pandas as pd
df = pd.DataFrame({
                'm1':[50,'A',60,'A',80],
                'm2':[60,'A','60','A',80],
                'm3':[50,70,'A','A',60],
                'm4':[60,'A','A','A',60],
                'm5':['A','A','A',10,20]
                })

df = df.apply(pd.to_numeric,errors='coerce')

print(df)

print('Dataframe with NaN\n\n\n')
# Make all the NaN in Mark5 as zero
df['m5']=df['m5'].fillna(0)
print(df)
print('Making m5 NaN as 0 using fillna() function\n\n\n\n')

df1 = df.copy()
df1['m2'].fillna(df1['m2'].mean(),inplace=True)
print(df1)
print('Making m5 NaN as mean using fillna() function\n\n\n\n')

df2 = df.copy()
df1['m3'].fillna(df1['m2'].median(),inplace=True)
print(df2)
print('Making m5 NaN as median using fillna() function\n\n\n\n')

# Dropping all columns having NaN
df=df.dropna(axis=1)
print(df)
print('Dropping all columns having NaN\n\n\n\n')
```

## Catalog 4

This Catalog illustrates the use of MinMax scaling and Standard scaling for finding Z-scores.

```
from numpy import asarray
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

data = asarray([[1,3],[8,5],[6,7],[8,9]])
print("\n Original Data")
print(data)

scaler1 = MinMaxScaler()
scaler2 = StandardScaler()

scaled1 = scaler1.fit_transform(data)
scaled2 = scaler2.fit_transform(data)

print("\n\nThe output of MinMax Scaling")
print(scaled1)


print("\n\nThe output of Standard scaling as z-score")
print(scaled2)
```