Data Pre-processing

Aim of the Experiment.

The main aim of this experiment is to preprocess the given dataset. The database is created and is available in the file sample.csv.

Sample Dataset

id	first	last	gender	Marks	selected
1	Leone	Debrick	Female	50	TRUE
2	Romola	Phinnessy	Female	60	FALSE
3	Geri	Prium	Male	65	FALSE
4	Sandy	Doveston	Female	95	FALSE
5	Jacenta	Jansik	Female	31	TRUE
6	Diane-marie	Medhurst	Female	45	TRUE
7	Austen	Pool	Male	45	TRUE
8	Vanya	Teffrey	Male	70	FALSE
9	Giordano	Elloy	Male	36	FALSE
10	Rozele	Fawcett	Female	50	FALSE

The objectives of this experiment are

1. Explore Label Encoder
2. Explore Scikit Preprocessing routines like Scaling
3. Explore Scikit Preprocessing 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.

Listing 1

```
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 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")
```

Output

```
In [63]: runfile('D:/Test/listing 2.py', wdir='D:/Test')
   id        first       last  gender  Marks  selected
0   1        Leone    Debrick  Female     50      True
1   2       Romola  Phinnessy  Female     60     False
2   3         Geri      Prium    Male     65     False
3   4        Sandy   Doveston  Female     95     False
4   5      Jacenta     Jansik  Female     31      True
5   6  Diane-marie   Medhurst  Female     45      True
6   7       Austen       Pool    Male     45      True
7   8        Vanya    Teffrey    Male     70     False
8   9     Giordano      Elloy    Male     36     False
9  10       Rozele    Fawcett  Female     50     False
End of Listing
```

```
   id        first       last  gender  Marks  selected
0   1        Leone    Debrick       0     50      True
1   2       Romola  Phinnessy       0     60     False
2   3         Geri      Prium       1     65     False
3   4        Sandy   Doveston       0     95     False
4   5      Jacenta     Jansik       0     31      True
5   6  Diane-marie   Medhurst       0     45      True
6   7       Austen       Pool       1     45      True
7   8        Vanya    Teffrey       1     70     False
8   9     Giordano      Elloy       1     36     False
9  10       Rozele    Fawcett       0     50     False
End of Listing
```

```
   id        first       last  gender     Marks  selected
0   1        Leone    Debrick       0 -0.265401      True
1   2       Romola  Phinnessy       0  0.299282     False
2   3         Geri      Prium       1  0.581624     False
3   4        Sandy   Doveston       0  2.275674     False
4   5      Jacenta     Jansik       0 -1.338300      True
5   6  Diane-marie   Medhurst       0 -0.547743      True
6   7       Austen       Pool       1 -0.547743      True
7   8        Vanya    Teffrey       1  0.863966     False
8   9     Giordano      Elloy       1 -1.055958     False
9  10       Rozele    Fawcett       0 -0.265401     False
Scaling of marks is completed
```

```
   id        first       last  gender  Marks  selected
0   1        Leone    Debrick       0    0.0      True
1   2       Romola  Phinnessy       0    0.0     False
2   3         Geri      Prium       1    1.0     False
3   4        Sandy   Doveston       0    1.0     False
4   5      Jacenta     Jansik       0    0.0      True
5   6  Diane-marie   Medhurst       0    0.0      True
6   7       Austen       Pool       1    0.0      True
7   8        Vanya    Teffrey       1    1.0     False
8   9     Giordano      Elloy       1    0.0     False
9  10       Rozele    Fawcett       0    0.0     False
Binarizarion of marks is completed
```

Listing 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")
```

Output

```
   id        first       last  gender  Marks  selected
0   1        Leone    Debrick  Female     50      True
1   2       Romola  Phinnessy  Female     60     False
2   3         Geri      Prium    Male     65     False
3   4        Sandy   Doveston  Female     95     False
4   5      Jacenta     Jansik  Female     31      True
5   6  Diane-marie   Medhurst  Female     45      True
6   7       Austen       Pool    Male     45      True
7   8        Vanya    Teffrey    Male     70     False
8   9     Giordano      Elloy    Male     36     False
9  10       Rozele    Fawcett  Female     50     False
End of Listing
```

```
    id        first       last  gender  Marks  selected
0    1        Leone    Debrick  Female     50      True
1    2       Romola  Phinnessy  Female     60     False
2    3         Geri      Prium    Male     65     False
3    4        Sandy   Doveston  Female     95     False
4    5      Jacenta     Jansik  Female     31      True
5    6  Diane-marie   Medhurst  Female     45      True
6    7       Austen       Pool    Male     45      True
7    8        Vanya    Teffrey    Male     70     False
8    9     Giordano      Elloy    Male     36     False
9   10       Rozele    Fawcett  Female     50     False
10   1        Leone    Debrick  Female     50      True
11   2       Romola  Phinnessy  Female     60     False
12   3         Geri      Prium    Male     65     False
13   4        Sandy   Doveston  Female     95     False
14   5      Jacenta     Jansik  Female     31      True
15   6  Diane-marie   Medhurst  Female     45      True
16   7       Austen       Pool    Male     45      True
17   8        Vanya    Teffrey    Male     70     False
18   9     Giordano      Elloy    Male     36     False
19  10       Rozele    Fawcett  Female     50     False
Display before duplication
```

```
   id        first       last  gender  Marks  selected
0   1        Leone    Debrick  Female     50      True
1   2       Romola  Phinnessy  Female     60     False
2   3         Geri      Prium    Male     65     False
3   4        Sandy   Doveston  Female     95     False
4   5      Jacenta     Jansik  Female     31      True
5   6  Diane-marie   Medhurst  Female     45      True
6   7       Austen       Pool    Male     45      True
7   8        Vanya    Teffrey    Male     70     False
8   9     Giordano      Elloy    Male     36     False
9  10       Rozele    Fawcett  Female     50     False
Display after duplication
```

Listing 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')
```

Output

```
In [104]: runfile('D:/Test/listing 2B.py', wdir='D:/Test')
     m1    m2    m3    m4    m5
0  50.0  60.0  50.0  60.0   NaN
1   NaN   NaN  70.0   NaN   NaN
2  60.0  60.0   NaN   NaN   NaN
3   NaN   NaN   NaN   NaN  10.0
4  80.0  80.0  60.0  60.0  20.0
Dataframe with NaN
```

```
     m1    m2    m3    m4    m5
0  50.0  60.0  50.0  60.0   0.0
1   NaN   NaN  70.0   NaN   0.0
2  60.0  60.0   NaN   NaN   0.0
3   NaN   NaN   NaN   NaN  10.0
4  80.0  80.0  60.0  60.0  20.0
Making m5 NaN as 0 using fillna() function



     m1         m2    m3    m4    m5
0  50.0  60.000000  50.0  60.0   0.0
1   NaN  66.666667  70.0   NaN   0.0
2  60.0  60.000000   NaN   NaN   0.0
3   NaN  66.666667   NaN   NaN  10.0
4  80.0  80.000000  60.0  60.0  20.0
Making m5 NaN as mean using fillna() function
```

```
     m1         m2    m3    m4    m5
0  50.0  60.000000  50.0  60.0   0.0
1   NaN  66.666667  70.0   NaN   0.0
2  60.0  60.000000   NaN   NaN   0.0
3   NaN  66.666667   NaN   NaN  10.0
4  80.0  80.000000  60.0  60.0  20.0
Making m5 NaN as mean using fillna() function



     m1    m2    m3    m4    m5
0  50.0  60.0  50.0  60.0   0.0
1   NaN   NaN  70.0   NaN   0.0
2  60.0  60.0   NaN   NaN   0.0
3   NaN   NaN   NaN   NaN  10.0
4  80.0  80.0  60.0  60.0  20.0
Making m5 NaN as median using fillna() function
```

```
     m5
0   0.0
1   0.0
2   0.0
3  10.0
4  20.0
Dropping all columns having NaN
```

Listing 4

This listing 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)
```

Output

```
In [26]: runfile('D:/Test/Lab2.minmax.py', wdir='D:/Test')

 Original Data
[[1 3]
 [8 5]
 [6 7]
 [8 9]]


The output of MinMax Scaling
[[0.         0.        ]
 [1.         0.33333333]
 [0.71428571 0.66666667]
 [1.         1.        ]]


The output of Standard scaling as z-score
[[-1.66003771 -1.34164079]
 [ 0.78633365 -0.4472136 ]
 [ 0.08737041  0.4472136 ]
 [ 0.78633365  1.34164079]]
```