LAB ASSIGNMENT – 6

DATA TRANSFORMATION & NORMALIZATION

NAME : PRATHAPANI SATWIKA

REG.NO. : 20BCD7160

1.Download the bank.csv file and do the above transformations. After data transformation, save the data into preprocessed_bank.csv file.

Attribute	Transformation		Help
marital	single	0	Use user-defined function and apply to the entire attribute column
	Other	1	
housing	no	0	Use map function
	yes	1	
loan	no	0	Use replace function
	yes	1	
job	'unknown'	np.nan	Use replace function and inplace attribute
	'management'	0	
	'technician'	1	
	'entrepreneur'	2	
	'blue-collar'	3	
	'retired'	4	
	'admin.'	5	
	'services'	6	
	'self-employed'	7	
	'unemployed'	8	
	'housemaid'	9	
	'student'	10	
education	'unknown'	np.nan	Use replace or map function
	'tertiary'	0	
	'secondary'	1	
	'primary'	2	
default	no	0	Use replace or map function
	yes	1	
contact	unknown	np.nan	Use replace or map function
	telephone	0	
	cellular	1	
month	jan-dec	1-12	Use replace or map function
poutcome	'unknown'	np.nan	Use replace or map function
	'failure'	0	
	'other'	1	
	'success'	2	
y	no	0	Use replace or map function
	yes	1	
Develop user defined functions for following Normalizations:			

CODE :

```
[ ] import matplotlib.pyplot as plt
    import numpy as np
    import pandas as pd
```

```
data=pd.read_csv("bank.csv",sep=';')
df=pd.DataFrame(data)
data.head()
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	1	-1	0	unknown	no
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	1	339	4	failure	no
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	1	330	1	failure	no
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	4	-1	0	unknown	no
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	1	-1	0	unknown	no

```
[ ] def transformMarital(column,value):
      df[column] = np.where(df[column].str.contains(value), 0, 1)
      transformMarital("marital","single")
```

```
[ ] df['housing']=df['housing'].map({'no': 0,'yes':1})
```

```
[ ] df['loan']=df['loan'].replace(['yes'],1)
```

```
df['loan']=df['loan'].replace(['no'],0)
df['job'].replace({'unknown':np.nan,'management': 0,'technician': 1, 'entrepreneur': 2,'blue-collar': 3,'retired': 4,'admin.': 5,'services': 6,
                   'self-employed' :7, 'unemployed': 8,'housemaid': 9,'student': 10},inplace=True)
```

```
[ ] df['education'].replace({'unknown':np.nan,'tertiary': 0,'secondary': 1, 'primary': 2},inplace=True)
```

```
[ ] df['default']=df['default'].replace(['yes'],1)
```

```
[ ] df['default']=df['default'].replace(['no'],0)
```

```
[ ] df['contact'].replace({'unknown':np.nan,'telephone': 0,'cellular': 1},inplace=True)
```

```
[ ] df['month'].unique()
```

```
array(['oct', 'may', 'apr', 'jun', 'feb', 'aug', 'jan', 'jul', 'nov',
       'sep', 'mar', 'dec'], dtype=object)
```

```
[ ] df['month'].replace({ "jan" : 1 , "feb" : 2 , "mar" : 3 , "apr" : 4 , "may" : 5 , "june" : 6, "july" : 7,"august" :8 , "september": 9,"october" : 10, "november" : 11,"december" : 12}, inplace=True)
```

```
[ ] df=df.replace({'poutcome': {'failure': 0,'unknown': 1,'success': 2 }})
```

```
df=df.replace({'y': {'no': 0,'yes': 1 }})
```

OUTPUT :

```
[ ] df
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	30	8.0	married	2.0	0	1787	0	0	1.0	19	oct	79	1	-1	0	1	0
1	33	6.0	married	1.0	0	4789	1	1	1.0	11	5	220	1	339	4	0	0
2	35	0.0	single	0.0	0	1350	1	0	1.0	16	4	185	1	330	1	0	0
3	30	0.0	married	0.0	0	1476	1	1	NaN	3	jun	199	4	-1	0	1	0
4	59	3.0	married	1.0	0	0	1	0	NaN	5	5	226	1	-1	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4516	33	6.0	married	1.0	0	-333	1	0	1.0	30	jul	329	5	-1	0	1	0
4517	57	7.0	married	0.0	1	-3313	1	1	NaN	9	5	153	1	-1	0	1	0
4518	57	1.0	married	1.0	0	295	0	0	1.0	19	aug	151	11	-1	0	1	0
4519	28	3.0	married	1.0	0	1137	0	0	1.0	6	2	129	4	211	3	other	0
4520	44	2.0	single	0.0	0	1136	1	1	1.0	3	4	345	2	249	7	other	0

4521 rows × 17 columns

2. Develop user defined functions for following Normalizations

duration	min-max normalization to [0,1]	Observe the data distribution before and after data normalization with a plot.
	z-score normalization	Find the data points which are far from three standard deviations.
pdays	min-max normalization to [0,1]	Observe the data distribution before and after data normalization with a plot.
	z-score normalization	Find the data points which are far from three standard deviations.
balance	min-max normalization to [0,1]	Observe the data distribution before and after data normalization with a plot.
	z-score normalization	Find the data points which are far from three standard deviations.

CODE :

```
[20] def minmax(df,column):
       df[column]= (df[column] - df[column].min()) / (df[column].max() -df[column].min())
       return df
```

```
[32] def zscore(df,column):
       mean = np.mean(df[column])
       std = np.std(df[column])
       threshold=3
       outlier = []
       for i in df[column]: z = (i-mean)/std
       if z > threshold:
         fig, axs = plt.subplots(2,figsize=(10,8))
         fig.suptitle('Min-Max of duration')
         axs[0].scatter(df.index,df["duration"])
         axs[0].set_title("Before min-max norm")
         df=minmax(df,"duration")
         zscore(df,"duration")
         fig.suptitle('Min-Max of pdays')
         axs[0].scatter(df.index,df["pdays"])
         axs[0].set_title("Before min-max norm")
         df=minmax(df,"pdays")
         axs[1].scatter(df.index,df["pdays"])
         zscore(df,"pdays")
         df.to_csv('processed_bank.csv', index=False)
```

OUTPUT :

```
df
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	30	8.0	married	primary	no	1787	0	0	cellular	19	oct	79	1	-1	0	unknown	no
1	33	6.0	married	secondary	no	4789	1	1	cellular	11	may	220	1	339	4	failure	no
2	35	0.0	single	tertiary	no	1350	1	0	cellular	16	apr	185	1	330	1	failure	no
3	30	0.0	married	tertiary	no	1476	1	1	unknown	3	jun	199	4	-1	0	unknown	no
4	59	3.0	married	secondary	no	0	1	0	unknown	5	may	226	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4516	33	6.0	married	secondary	no	-333	1	0	cellular	30	jul	329	5	-1	0	unknown	no
4517	57	7.0	married	tertiary	yes	-3313	1	1	unknown	9	may	153	1	-1	0	unknown	no
4518	57	1.0	married	secondary	no	295	0	0	cellular	19	aug	151	11	-1	0	unknown	no
4519	28	3.0	married	secondary	no	1137	0	0	cellular	6	feb	129	4	211	3	other	no
4520	44	2.0	single	tertiary	no	1136	1	1	cellular	3	apr	345	2	249	7	other	no

4521 rows × 17 columns