

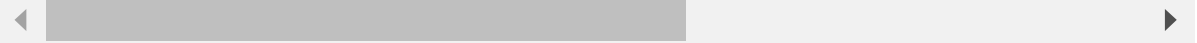
```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [31]: #Read the file
bank_df=pd.read_csv("bank.csv",sep=";")
bank_df
```

```
Out[31]:
```

	age	job	marital	education	default	balance	housing	loan	contact	di
0	30	unemployed	married	primary	no	1787	no	no	cellular	
1	33	services	married	secondary	no	4789	yes	yes	cellular	
2	35	management	single	tertiary	no	1350	yes	no	cellular	
3	30	management	married	tertiary	no	1476	yes	yes	unknown	
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	
...	...	...	...	...	...	...	...	...	...	
4516	33	services	married	secondary	no	-333	yes	no	cellular	
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	
4518	57	technician	married	secondary	no	295	no	no	cellular	
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	

4521 rows × 17 columns



===== Change categorical columns into numerical columns =====

**map**

```
In [35]: #change job column in categorical to numerical using map method
bank_df=pd.read_csv("bank.csv",sep=";")
unique_label=sorted(bank_df["job"].unique())
list1=[i for i in range(len(unique_label))]
dict1={i:j for i,j in zip(unique_label,list1)}
bank_df["job_update"]=bank_df["job"].map(dict1)
bank_df
```

Out[35]:

	age	job	marital	education	default	balance	housing	loan	contact	date
0	30	unemployed	married	primary	no	1787	no	no	cellular	...
1	33	services	married	secondary	no	4789	yes	yes	cellular	...
2	35	management	single	tertiary	no	1350	yes	no	cellular	...
3	30	management	married	tertiary	no	1476	yes	yes	unknown	...
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	...
...	...	...	...	...	...	...	...	...	...	...
4516	33	services	married	secondary	no	-333	yes	no	cellular	...
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	...
4518	57	technician	married	secondary	no	295	no	no	cellular	...
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	...
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	...

4521 rows × 18 columns

In [43]: *#apply map method on all categorical columns in the dataframe.*

```

bank_df=pd.read_csv("bank.csv",sep=";")
cat_col=bank_df.select_dtypes(include="object").columns
for i in cat_col:
    unique_label=sorted(bank_df[i].unique())
    list1=[i for i in range(len(unique_label))]
    dict1={i:j for i,j in zip(unique_label,list1)}
    bank_df[i]=bank_df[i].map(dict1)
bank_df

```

Out[43]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
<b>0</b>	30	10	1	0	0	1787	0	0	0	19	10
<b>1</b>	33	7	1	1	0	4789	1	1	0	11	8
<b>2</b>	35	4	2	2	0	1350	1	0	0	16	0
<b>3</b>	30	4	1	2	0	1476	1	1	2	3	6
<b>4</b>	59	1	1	1	0	0	1	0	2	5	8
<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>
<b>4516</b>	33	7	1	1	0	-333	1	0	0	30	5
<b>4517</b>	57	6	1	2	1	-3313	1	1	2	9	8
<b>4518</b>	57	9	1	1	0	295	0	0	0	19	1
<b>4519</b>	28	1	1	1	0	1137	0	0	0	6	3
<b>4520</b>	44	2	2	2	0	1136	1	1	0	3	0

4521 rows × 17 columns

**Labelencoder**

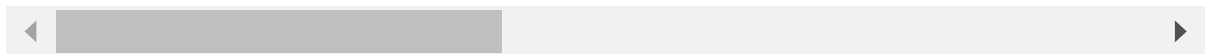
```
In [47]: # Apply Labelencoder method on dataframe to change columns value categorical to num
# add new columns.
bank_df=pd.read_csv("bank.csv",sep=";")
cat_col=bank_df.select_dtypes(include="object").columns

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for i in cat_col:
    bank_df[f"{i}_new"]=le.fit_transform(bank_df[i])
bank_df
```

Out[47]:

	age	job	marital	education	default	balance	housing	loan	contact	date
0	30	unemployed	married	primary	no	1787	no	no	cellular	...
1	33	services	married	secondary	no	4789	yes	yes	cellular	...
2	35	management	single	tertiary	no	1350	yes	no	cellular	...
3	30	management	married	tertiary	no	1476	yes	yes	unknown	...
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	...
...	...	...	...	...	...	...	...	...	...	...
4516	33	services	married	secondary	no	-333	yes	no	cellular	...
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	...
4518	57	technician	married	secondary	no	295	no	no	cellular	...
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	...
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	...

4521 rows × 27 columns

In [49]: *# Apply LabelEncoder method on dataframe to change columns value categorical to num*

```

bank_df=pd.read_csv("bank.csv",sep=";")
cat_col=bank_df.select_dtypes(include="object").columns

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for i in cat_col:
    bank_df[i]=le.fit_transform(bank_df[i])
bank_df

```

Out[49]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
<b>0</b>	30	10	1	0	0	1787	0	0	0	19	10
<b>1</b>	33	7	1	1	0	4789	1	1	0	11	8
<b>2</b>	35	4	2	2	0	1350	1	0	0	16	0
<b>3</b>	30	4	1	2	0	1476	1	1	2	3	6
<b>4</b>	59	1	1	1	0	0	1	0	2	5	8
<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>
<b>4516</b>	33	7	1	1	0	-333	1	0	0	30	5
<b>4517</b>	57	6	1	2	1	-3313	1	1	2	9	8
<b>4518</b>	57	9	1	1	0	295	0	0	0	19	1
<b>4519</b>	28	1	1	1	0	1137	0	0	0	6	3
<b>4520</b>	44	2	2	2	0	1136	1	1	0	3	0

4521 rows × 17 columns

**np.where**

```
In [51]: bank_df=pd.read_csv("bank.csv",sep=";")
bank_df["housing"].unique()
```

```
Out[51]: array(['no', 'yes'], dtype=object)
```

```
In [53]: #np.where
con=bank_df["housing"]=="no"
np.where(con,0,1)                                #limitation it apply only with two labels.
```

```
Out[53]: array([0, 1, 1, ..., 0, 0, 1])
```

**One-hot-Encoder**

```
In [67]: bank_df
```

Out[67]:

	age	job	marital	education	default	balance	housing	loan	contact	date
0	30	unemployed	married	primary	no	1787	no	no	cellular	...
1	33	services	married	secondary	no	4789	yes	yes	cellular	...
2	35	management	single	tertiary	no	1350	yes	no	cellular	...
3	30	management	married	tertiary	no	1476	yes	yes	unknown	...
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	...
...	...	...	...	...	...	...	...	...	...	...
4516	33	services	married	secondary	no	-333	yes	no	cellular	...
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	...
4518	57	technician	married	secondary	no	295	no	no	cellular	...
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	...
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	...

4521 rows × 17 columns



```
In [61]: #unique values in jobs columns
bank_df["job"].unique()
```

```
Out[61]: array(['unemployed', 'services', 'management', 'blue-collar',
               'self-employed', 'technician', 'entrepreneur', 'admin.', 'student',
               'housemaid', 'retired', 'unknown'], dtype=object)
```

```
In [65]: #Apply get_dummies method on job columns
job_dummes=pd.get_dummies(bank_df["job"],dtype="int")
job_dummes                                     # get seperated columns
```

Out[65]:

	admin.	blue-collar	entrepreneur	housemaid	management	retired	self-employed	services
<b>0</b>	0	0	0	0	0	0	0	0
<b>1</b>	0	0	0	0	0	0	0	1
<b>2</b>	0	0	0	0	1	0	0	0
<b>3</b>	0	0	0	0	1	0	0	0
<b>4</b>	0	1	0	0	0	0	0	0
<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>
<b>4516</b>	0	0	0	0	0	0	0	1
<b>4517</b>	0	0	0	0	0	0	1	0
<b>4518</b>	0	0	0	0	0	0	0	0
<b>4519</b>	0	1	0	0	0	0	0	0
<b>4520</b>	0	0	1	0	0	0	0	0

4521 rows × 12 columns

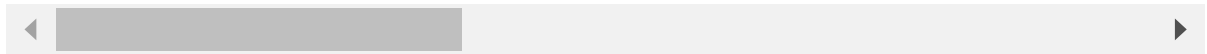


```
In [94]: # Apply the get_dummies methods on dataframe.
bank_dummes=pd.get_dummies(bank_df, dtype="int")
bank_dummes
```

Out[94]:

	age	balance	day	duration	campaign	pdays	previous	job_admin.	job_blue-collar	job
<b>0</b>	30	1787	19	79	1	-1	0	0	0	
<b>1</b>	33	4789	11	220	1	339	4	0	0	
<b>2</b>	35	1350	16	185	1	330	1	0	0	
<b>3</b>	30	1476	3	199	4	-1	0	0	0	
<b>4</b>	59	0	5	226	1	-1	0	0	1	
...	...	...	...	...	...	...	...	...	...	...
<b>4516</b>	33	-333	30	329	5	-1	0	0	0	
<b>4517</b>	57	-3313	9	153	1	-1	0	0	0	
<b>4518</b>	57	295	19	151	11	-1	0	0	0	
<b>4519</b>	28	1137	6	129	4	211	3	0	1	
<b>4520</b>	44	1136	3	345	2	249	7	0	0	

4521 rows × 53 columns



In [96]: *# After one hot encoding data frame columns and their counting.*  
 bank\_dummes.columns, len(bank\_dummes.columns)

Out[96]: (Index(['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous',  
 'job\_admin.', 'job\_blue-collar', 'job\_entrepreneur', 'job\_housemaid',  
 'job\_management', 'job\_retired', 'job\_self-employed', 'job\_services',  
 'job\_student', 'job\_technician', 'job\_unemployed', 'job\_unknown',  
 'marital\_divorced', 'marital\_married', 'marital\_single',  
 'education\_primary', 'education\_secondary', 'education\_tertiary',  
 'education\_unknown', 'default\_no', 'default\_yes', 'housing\_no',  
 'housing\_yes', 'loan\_no', 'loan\_yes', 'contact\_cellular',  
 'contact\_telephone', 'contact\_unknown', 'month\_apr', 'month\_aug',  
 'month\_dec', 'month\_feb', 'month\_jan', 'month\_jul', 'month\_jun',  
 'month\_mar', 'month\_may', 'month\_nov', 'month\_oct', 'month\_sep',  
 'poutcome\_failure', 'poutcome\_other', 'poutcome\_success',  
 'poutcome\_unknown', 'y\_no', 'y\_yes'],  
 dtype='object'),  
 53)

In [98]: *# categorical columns*  
 cat\_col

Out[98]: Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',  
 'month', 'poutcome', 'y'],  
 dtype='object')

In [100... *#categorical columns unique labels and their counting*  
 for i in cat\_col:



```
print(bank_df[i].unique(),bank_df[i].nunique())
```

```
['unemployed' 'services' 'management' 'blue-collar' 'self-employed'
 'technician' 'entrepreneur' 'admin.' 'student' 'housemaid' 'retired'
 'unknown'] 12
['married' 'single' 'divorced'] 3
['primary' 'secondary' 'tertiary' 'unknown'] 4
['no' 'yes'] 2
['no' 'yes'] 2
['no' 'yes'] 2
['cellular' 'unknown' 'telephone'] 3
['oct' 'may' 'apr' 'jun' 'feb' 'aug' 'jan' 'jul' 'nov' 'sep' 'mar' 'dec'] 12
['unknown' 'failure' 'other' 'success'] 4
['no' 'yes'] 2
```

```
In [122... df1=bank_df[["age","job"]]
```

```
In [128... df2=bank_df[["marital","education"]]
```

```
In [144... #concatination
df3=pd.concat([df1,df2],axis=1)
df3
```

```
Out[144...
   age  job  marital  education
0    30  unemployed  married    primary
1    33    services  married    secondary
2    35  management  single    tertiary
3    30  management  married    tertiary
4    59  blue-collar  married    secondary
...   ...      ...      ...      ...
4516  33    services  married    secondary
4517  57  self-employed  married    tertiary
4518  57    technician  married    secondary
4519  28  blue-collar  married    secondary
4520  44  entrepreneur  single    tertiary
```

4521 rows × 4 columns

```
In [152... df4=bank_df[["age","housing"]]
```

```
In [156... #merge
df5=pd.merge(df3,df4)
df5
```

Out[156...

	age	job	marital	education	housing
0	30	unemployed	married	primary	no
1	30	unemployed	married	primary	yes
2	30	unemployed	married	primary	no
3	30	unemployed	married	primary	yes
4	30	unemployed	married	primary	no
...	...	...	...	...	...
597344	44	entrepreneur	single	tertiary	yes
597345	44	entrepreneur	single	tertiary	no
597346	44	entrepreneur	single	tertiary	yes
597347	44	entrepreneur	single	tertiary	yes
597348	44	entrepreneur	single	tertiary	yes

597349 rows × 5 columns

In [168...

```
df6=bank_df[["loan"]]
```

In [170...

```
#join
df3.join(df6)
```

Out[170...

	age	job	marital	education	loan
0	30	unemployed	married	primary	no
1	33	services	married	secondary	yes
2	35	management	single	tertiary	no
3	30	management	married	tertiary	yes
4	59	blue-collar	married	secondary	no
...	...	...	...	...	...
4516	33	services	married	secondary	no
4517	57	self-employed	married	tertiary	yes
4518	57	technician	married	secondary	no
4519	28	blue-collar	married	secondary	no
4520	44	entrepreneur	single	tertiary	yes

4521 rows × 5 columns

In [ ]: