Encoding- converting categorical data into numerical data

- OneHotEncoder
- label endoder (it create same column instead of seperating it)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder

df=pd.read_csv(r"C:\Users\DELL\Downloads\my_python\Salary_EDA.csv")
df.head()
```

```
    Age  Gender Education Level          Job Title  Years of
Experience  \
0  32.0    Male      Bachelor's  Software Engineer
5.0
1  28.0  Female        Master's       Data Analyst
3.0
2  45.0    Male             PhD     Senior Manager
15.0
3  36.0  Female      Bachelor's    Sales Associate
7.0
4  36.0  Female      Bachelor's    Sales Associate
7.0

     Salary
0   90000.0
1   65000.0
2  150000.0
3   60000.0
4   60000.0
```

filter categorical feature

```
cat=['Education Level']
encoder=OneHotEncoder(drop=None,sparse_output=False)#drop =none used
to not delete any row in data and spares=false means not to modify
the matrix
en=encoder.fit_transform(df[cat])#original modify is done by
fit_transform like in 0's & 1's
en#in form of array

array([[1., 0., 0., 0.],
       [0., 1., 0., 0.],
       [0., 0., 1., 0.],
       ...,
       [1., 0., 0., 0.],
```

```
       [1., 0., 0., 0.],
       [0., 0., 1., 0.]])
```

the encoder data is in the form of array to convert it into the data frame , we need to convert the encodede data into the dataframe with categories as column name

```
endf=pd.DataFrame(en,columns=encoder.get_feature_names_out(cat))#colum
n name are take by the unique in that education level done by
get_feature_name
endf.head()
```

```
   Education Level_Bachelor's  Education Level_Master's  Education
Level_PhD  \
0                         1.0                       0.0
0.0
1                         0.0                       1.0
0.0
2                         0.0                       0.0
1.0
3                         1.0                       0.0
0.0
4                         1.0                       0.0
0.0

   Education Level_nan
0                  0.0
1                  0.0
2                  0.0
3                  0.0
4                  0.0
```

```
endf.drop(columns=["Education Level_nan"],inplace =True)#to delete the
column use drop /(["Education Level_nan"],axis=1,inplcae=True)

fdf=pd.concat([df,endf],axis=1)
fdf.head()
```

```
    Age  Gender Education Level          Job Title  Years of
Experience  \
0  32.0    Male      Bachelor's  Software Engineer
5.0
1  28.0  Female        Master's       Data Analyst
3.0
2  45.0    Male             PhD     Senior Manager
15.0
3  36.0  Female      Bachelor's    Sales Associate
7.0
4  36.0  Female      Bachelor's    Sales Associate
7.0
```

```
     Salary  Education Level_Bachelor's  Education Level_Master's  \
0   90000.0                         1.0                       0.0
1   65000.0                         0.0                       1.0
2  150000.0                         0.0                       0.0
3   60000.0                         1.0                       0.0
4   60000.0                         1.0                       0.0

   Education Level_PhD
0                  0.0
1                  0.0
2                  1.0
3                  0.0
4                  0.0
```

label encoder

```
df1=pd.read_csv(r"C:\Users\DELL\Downloads\my_python\Salary_EDA.csv")
df1.head()
```

```
    Age  Gender Education Level          Job Title  Years of
Experience  \
0  32.0    Male      Bachelor's  Software Engineer
5.0
1  28.0  Female        Master's       Data Analyst
3.0
2  45.0    Male             PhD     Senior Manager
15.0
3  36.0  Female      Bachelor's    Sales Associate
7.0
4  36.0  Female      Bachelor's    Sales Associate
7.0

     Salary
0   90000.0
1   65000.0
2  150000.0
3   60000.0
4   60000.0
```

```
le=LabelEncoder()#in this column name sare not created we have to give
that
df1["Gender_encode"]=le.fit_transform(df1["Gender"])
df1.head()
```

```
    Age  Gender Education Level          Job Title  Years of
Experience  \
0  32.0    Male      Bachelor's  Software Engineer
5.0
1  28.0  Female        Master's       Data Analyst
3.0
```

```
2  45.0    Male             PhD     Senior Manager
15.0
3  36.0  Female      Bachelor's    Sales Associate
7.0
4  36.0  Female      Bachelor's    Sales Associate
7.0

     Salary  Gender_encode  Education encode
0   90000.0              1                 0
1   65000.0              0                 1
2  150000.0              1                 2
3   60000.0              0                 0
4   60000.0              0                 0
```

```
le1=LabelEncoder()
df1["Education encode"]=le1.fit_transform(df1["Education Level"])
df1.head()
```

```
    Age  Gender Education Level          Job Title  Years of
Experience  \
0  32.0    Male      Bachelor's  Software Engineer
5.0
1  28.0  Female        Master's       Data Analyst
3.0
2  45.0    Male             PhD     Senior Manager
15.0
3  36.0  Female      Bachelor's    Sales Associate
7.0
4  36.0  Female      Bachelor's    Sales Associate
7.0

     Salary  Gender_encode  Education encode
0   90000.0              1                 0
1   65000.0              0                 1
2  150000.0              1                 2
3   60000.0              0                 0
4   60000.0              0                 0
```

standardization

min-max scaling: which can scale t he ma didgit like 100000000000,1444455565676 to simple range between(0-1) 0.24,0.989

```
from sklearn.preprocessing import MinMaxScaler

df2=pd.read_csv(r"C:\Users\DELL\Downloads\my_python\Salary_EDA.csv")
df2.head()
```

```
    Age  Gender Education Level          Job Title  Years of
Experience  \
```

```
0  32.0    Male      Bachelor's  Software Engineer
5.0
1  28.0  Female        Master's       Data Analyst
3.0
2  45.0    Male             PhD     Senior Manager
15.0
3  36.0  Female      Bachelor's    Sales Associate
7.0
4  36.0  Female      Bachelor's    Sales Associate
7.0

     Salary
0   90000.0
1   65000.0
2  150000.0
3   60000.0
4   60000.0
```

```
scale=MinMaxScaler()
df2[['salary_scale']]=scale.fit_transform(df2[['Salary']])
df2.head()
```

```
    Age  Gender Education Level          Job Title  Years of
Experience  \
0  32.0    Male      Bachelor's  Software Engineer
5.0
1  28.0  Female        Master's       Data Analyst
3.0
2  45.0    Male             PhD     Senior Manager
15.0
3  36.0  Female      Bachelor's    Sales Associate
7.0
4  36.0  Female      Bachelor's    Sales Associate
7.0

     Salary  salary_scale
0   90000.0      0.359103
1   65000.0      0.258963
2  150000.0      0.599439
3   60000.0      0.238935
4   60000.0      0.238935
```

Z-score Normalization

```
from sklearn.preprocessing import StandardScaler# x-mean/std (big
values are convert into smaller vales )

mor=StandardScaler()
df2[['Sal_std']]=mor.fit_transform(df2[['Salary']])
df2[['Sal_std','Salary']].head()
```

```
    Sal_std    Salary
0 -0.211488   90000.0
1 -0.733148   65000.0
2  1.040496  150000.0
3 -0.837480   60000.0
4 -0.837480   60000.0
```

```
df2.head()
```

```
    Age  Gender Education Level          Job Title  Years of
Experience  \
0  32.0    Male      Bachelor's  Software Engineer
5.0
1  28.0  Female        Master's       Data Analyst
3.0
2  45.0    Male             PhD     Senior Manager
15.0
3  36.0  Female      Bachelor's    Sales Associate
7.0
4  36.0  Female      Bachelor's    Sales Associate
7.0

     Salary  salary_scale   Sal_std
0   90000.0      0.359103 -0.211488
1   65000.0      0.258963 -0.733148
2  150000.0      0.599439  1.040496
3   60000.0      0.238935 -0.837480
4   60000.0      0.238935 -0.837480
```