encoding

October 16, 2023

[1]:

```
import pandas as pd

# load data from CSV
df = pd.read_csv('insurance.csv')
```

[2]:

```
df.head()
```

[2]:

```
   age     sex     bmi  children smoker     region      charges
0   19  female  27.900         0    yes  southwest  16884.92400
1   18    male  33.770         1     no  southeast   1725.55230
2   28    male  33.000         3     no  southeast   4449.46200
3   33    male  22.705         0     no  northwest  21984.47061
4   32    male  28.880         0     no  northwest   3866.85520
```

0.0.1 Measures of Central Tendency

[3]:

```
# mean value of a column
df.bmi.mean()
```

[3]:

```
30.663396860986538
```

[4]:

```
# median value of a column
df.bmi.median()
```

[4]:

```
30.4
```

[5]:

```
# mode value of a column
df.bmi.mode()
```

[5]:

```
0    32.3
dtype: float64
```

[6]:

```
# data correlation
df.corr()
```

[6]:

```
               age       bmi  children   charges
age       1.000000  0.109272  0.042469  0.299008
```

```
bmi       0.109272  1.000000  0.012759  0.198341
children  0.042469  0.012759  1.000000  0.067998
charges   0.299008  0.198341  0.067998  1.000000
```

[7]:
```
# counts of unique values
df['sex'].value_counts()
```

[7]:
```
male      676
female    662
Name: sex, dtype: int64
```

[8]:
```
# descriptive statistics
df.describe()
```

[8]:
```
               age          bmi     children       charges
count  1338.000000  1338.000000  1338.000000   1338.000000
mean     39.207025    30.663397     1.094918  13270.422265
std      14.049960     6.098187     1.205493  12110.011237
min      18.000000    15.960000     0.000000   1121.873900
25%      27.000000    26.296250     0.000000   4740.287150
50%      39.000000    30.400000     1.000000   9382.033000
75%      51.000000    34.693750     2.000000  16639.912515
max      64.000000    53.130000     5.000000  63770.428010
```

0.0.2 Handling NaN values (If have)

[9]:
```
# checking null values
df.isnull().sum()
```

[9]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

[10]:
```
# filling NaN data with mean value
df.bmi = df.bmi.fillna(df.bmi.mean())
```

1 Encoding

1.1 Label Encoder

```
[11]: df2 = df.copy()
      df2.head()
```

```
[11]:    age     sex     bmi  children smoker     region      charges
      0   19  female  27.900         0    yes  southwest  16884.92400
      1   18    male  33.770         1     no  southeast   1725.55230
      2   28    male  33.000         3     no  southeast   4449.46200
      3   33    male  22.705         0     no  northwest  21984.47061
      4   32    male  28.880         0     no  northwest   3866.85520
```

```
[12]: # unique values
      df2.region.unique()
```

```
[12]: array(['southwest', 'southeast', 'northwest', 'northeast'], dtype=object)
```

```
[13]: from sklearn.preprocessing import LabelEncoder
      le = LabelEncoder()

      df2.sex = le.fit_transform(df2['sex'])
      df2.smoker = le.fit_transform(df2['smoker'])
      df2.region = le.fit_transform(df2['region'])
```

```
[14]: df2.head()
```

```
[14]:    age  sex     bmi  children  smoker  region      charges
      0   19    0  27.900         0       1       3  16884.92400
      1   18    1  33.770         1       0       2   1725.55230
      2   28    1  33.000         3       0       2   4449.46200
      3   33    1  22.705         0       0       1  21984.47061
      4   32    1  28.880         0       0       1   3866.85520
```

1.1.1 using loop

```
[15]: df3 = df.copy()
      df3.head()
```

```
[15]:    age     sex     bmi  children smoker     region      charges
      0   19  female  27.900         0    yes  southwest  16884.92400
      1   18    male  33.770         1     no  southeast   1725.55230
      2   28    male  33.000         3     no  southeast   4449.46200
      3   33    male  22.705         0     no  northwest  21984.47061
      4   32    male  28.880         0     no  northwest   3866.85520
```

[16]:

```
#dataframe columns
columns = df3.columns
columns
```

[16]:

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'],
dtype='object')
```

[17]:

```
#all column in a single shot using numpy
#cautious: all unnecessary numerical data like 'age' will be transformed also
import numpy as np

for column in columns:
    if df3[column].dtype == np.number:
        continue
    df3[column] = le.fit_transform(df3[column])
```

[18]:

```
df3.head()
```

[18]:

```
   age  sex     bmi  children  smoker  region      charges
0    1    0  27.900         0       1       3  16884.92400
1    0    1  33.770         1       0       2   1725.55230
2   10    1  33.000         3       0       2   4449.46200
3   15    1  22.705         0       0       1  21984.47061
4   14    1  28.880         0       0       1   3866.85520
```

[19]:

```
#Bypass warning messages

#import warnings
#warnings.filterwarnings('ignore')
```

[20]:

```
df4 = df.copy()
df4.head()
```

[20]:

```
   age     sex     bmi  children smoker     region      charges
0   19  female  27.900         0    yes  southwest  16884.92400
1   18    male  33.770         1     no  southeast   1725.55230
2   28    male  33.000         3     no  southeast   4449.46200
3   33    male  22.705         0     no  northwest  21984.47061
4   32    male  28.880         0     no  northwest   3866.85520
```

[21]:

```
#all column in a single shot using pandas
from pandas.core.dtypes.common import is_numeric_dtype
for column in columns:
    if is_numeric_dtype(df4[column]):
        continue
    df4[column] = le.fit_transform(df4[column])
```

```
[22]: df4.head()
```

```
[22]:    age  sex     bmi  children  smoker  region      charges
      0   19    0  27.900         0       1       3  16884.92400
      1   18    1  33.770         1       0       2   1725.55230
      2   28    1  33.000         3       0       2   4449.46200
      3   33    1  22.705         0       0       1  21984.47061
      4   32    1  28.880         0       0       1   3866.85520
```

1.1.2 One hot

```
[23]: df5 = df.copy()
      df5.head()
```

```
[23]:    age     sex     bmi  children smoker     region      charges
      0   19  female  27.900         0    yes  southwest  16884.92400
      1   18    male  33.770         1     no  southeast   1725.55230
      2   28    male  33.000         3     no  southeast   4449.46200
      3   33    male  22.705         0     no  northwest  21984.47061
      4   32    male  28.880         0     no  northwest   3866.85520
```

```
[24]: from sklearn.preprocessing import OneHotEncoder
      ohe = OneHotEncoder(drop = 'first')

      sex = pd.DataFrame(ohe.fit_transform(df5[['sex']]).toarray(),columns=['male'])
      smoker = pd.DataFrame(ohe.fit_transform(df5[['smoker']]).
       ↪toarray(),columns=['smoker'])
      region = pd.DataFrame(ohe.fit_transform(df5[['region']]).
       ↪toarray(),columns=['northwest','southeast','southwest'])

      df5.drop(['sex','smoker','region'],axis=1, inplace=True)

      df5.join([sex,smoker,region]).head()
```

```
[24]:    age     bmi  children      charges  male  smoker  northwest  southeast  \
      0   19  27.900         0  16884.92400   0.0     1.0        0.0        0.0
      1   18  33.770         1   1725.55230   1.0     0.0        0.0        1.0
      2   28  33.000         3   4449.46200   1.0     0.0        0.0        1.0
      3   33  22.705         0  21984.47061   1.0     0.0        1.0        0.0
      4   32  28.880         0   3866.85520   1.0     0.0        1.0        0.0

         southwest
      0        1.0
      1        0.0
      2        0.0
      3        0.0
      4        0.0
```

1.1.3 using dummy table

[45]:
```
df6 = df.copy()
df6.head()
```

[45]:
```
   age     sex     bmi  children smoker     region      charges
0   19  female  27.900         0    yes  southwest  16884.92400
1   18    male  33.770         1     no  southeast   1725.55230
2   28    male  33.000         3     no  southeast   4449.46200
3   33    male  22.705         0     no  northwest  21984.47061
4   32    male  28.880         0     no  northwest   3866.85520
```

[48]:
```
pd.get_dummies(df6['sex'])
```

[48]:
```
      female  male
0          1     0
1          0     1
2          0     1
3          0     1
4          0     1
…      …     …
1333       0     1
1334       1     0
1335       1     0
1336       1     0
1337       1     0

[1338 rows x 2 columns]
```

[49]:
```
#dummy tables for columns
dummy = pd.get_dummies(df6,columns=['sex','smoker','region'],drop_first=True)

#dropping columns
df6 = df6.drop(columns=['sex','smoker','region'],axis=1)

#concating the dummy tables with rest dataframe
df6 = pd.concat([df6,dummy],axis=1)
```

[50]:
```
df6.head()
```

[50]:
```
   age     bmi  children      charges  age     bmi  children      charges  \
0   19  27.900         0  16884.92400   19  27.900         0  16884.92400
1   18  33.770         1   1725.55230   18  33.770         1   1725.55230
2   28  33.000         3   4449.46200   28  33.000         3   4449.46200
3   33  22.705         0  21984.47061   33  22.705         0  21984.47061
4   32  28.880         0   3866.85520   32  28.880         0   3866.85520
```

```
   sex_male  smoker_yes  region_northwest  region_southeast  region_southwest
0         0           1                 0                 0                 1
1         1           0                 0                 1                 0
2         1           0                 0                 1                 0
3         1           0                 1                 0                 0
4         1           0                 1                 0                 0
```

1.1.4 using loop

[53]:
```
df7 = df.copy()
df7.head()
```

[53]:
```
   age     sex     bmi  children smoker     region      charges
0   19  female  27.900         0    yes  southwest  16884.92400
1   18    male  33.770         1     no  southeast   1725.55230
2   28    male  33.000         3     no  southeast   4449.46200
3   33    male  22.705         0     no  northwest  21984.47061
4   32    male  28.880         0     no  northwest   3866.85520
```

[54]:
```
df7.columns
```

[54]:
```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'],
dtype='object')
```

[55]:
```
from pandas.core.dtypes.common import is_string_dtype

for column in columns:
#     df7[column].dtype.kind in 'biufc'
    if is_string_dtype(df7[column]):
        dummy = pd.get_dummies(df7[column],drop_first=True)
        df7 = df7.drop(column,axis=1)
        df7 = pd.concat([df7,dummy],axis=1)
```

[56]:
```
df7.head()
```

[56]:
```
   age     bmi  children      charges  male  yes  northwest  southeast  \
0   19  27.900         0  16884.92400     0    1          0          0
1   18  33.770         1   1725.55230     1    0          0          1
2   28  33.000         3   4449.46200     1    0          0          1
3   33  22.705         0  21984.47061     1    0          1          0
4   32  28.880         0   3866.85520     1    0          1          0

   southwest
0          1
1          0
2          0
3          0
```

```
4          0
```

```
[57]: x = df7.drop(columns=['charges'],axis=1)
      x.head()
```

```
[57]:    age     bmi  children  male  yes  northwest  southeast  southwest
      0   19  27.900         0     0    1          0          0          1
      1   18  33.770         1     1    0          0          1          0
      2   28  33.000         3     1    0          0          1          0
      3   33  22.705         0     1    0          1          0          0
      4   32  28.880         0     1    0          1          0          0
```

```
[35]: y = df7.charges
      y.head()
```

```
[35]: 0    16884.92400
      1     1725.55230
      2     4449.46200
      3    21984.47061
      4     3866.85520
      Name: charges, dtype: float64
```

1.1.5 Replace

```
[36]: df8 = df.copy()
      df8.head()
```

```
[36]:    age     sex     bmi  children smoker     region      charges
      0   19  female  27.900         0    yes  southwest  16884.92400
      1   18    male  33.770         1     no  southeast   1725.55230
      2   28    male  33.000         3     no  southeast   4449.46200
      3   33    male  22.705         0     no  northwest  21984.47061
      4   32    male  28.880         0     no  northwest   3866.85520
```

```
[37]: #replace data with corresponding identical value
      df8['sex'] = df8.sex.replace(['female','male'],[0,1])
      df8['smoker'] = df8.smoker.replace(['no','yes'],[0,1])
      df8['region'] = df8.region.
       ↪replace(['southwest','southeast','northwest','northeast'],[3,2,1,4])
```

```
[38]: df8.head()
```

```
[38]:    age  sex     bmi  children  smoker  region      charges
      0   19    0  27.900         0       1       3  16884.92400
      1   18    1  33.770         1       0       2   1725.55230
      2   28    1  33.000         3       0       2   4449.46200
      3   33    1  22.705         0       0       1  21984.47061
```

```
4   32     1  28.880          0        0        1   3866.85520
```

1.1.6 using loop

[39]:

```
df9 = df.copy()
df9.head()
```

[39]:

```
   age     sex     bmi  children smoker     region      charges
0   19  female  27.900         0    yes  southwest  16884.92400
1   18    male  33.770         1     no  southeast   1725.55230
2   28    male  33.000         3     no  southeast   4449.46200
3   33    male  22.705         0     no  northwest  21984.47061
4   32    male  28.880         0     no  northwest   3866.85520
```

[40]:

```
for column in columns:
    if is_string_dtype(df9[column]):
        unique = df9[column].unique()
        df9[column] = df9[column].replace(unique,list(range(len(unique))))
```

[41]:

```
df9.head()
```

[41]:

```
   age  sex     bmi  children  smoker  region      charges
0   19    0  27.900         0       0       0  16884.92400
1   18    1  33.770         1       1       1   1725.55230
2   28    1  33.000         3       1       1   4449.46200
3   33    1  22.705         0       1       2  21984.47061
4   32    1  28.880         0       1       2   3866.85520
```

1.1.7 Ordinal

[42]:

```
df10 = df.copy()
df10.head()
```

[42]:

```
   age     sex     bmi  children smoker     region      charges
0   19  female  27.900         0    yes  southwest  16884.92400
1   18    male  33.770         1     no  southeast   1725.55230
2   28    male  33.000         3     no  southeast   4449.46200
3   33    male  22.705         0     no  northwest  21984.47061
4   32    male  28.880         0     no  northwest   3866.85520
```

[43]:

```
from sklearn.preprocessing import OrdinalEncoder

#custom ordered
#OrdinalEncoder(categories=[['male','female'],['yes','no'],['northeast', 'northwest', 'southeast', 'southwest']])
oe = OrdinalEncoder()
```

```
df10[['sex','smoker','region']] = oe.
 ↪fit_transform(df10[['sex','smoker','region']])
df10.head()
```

```
[43]:    age  sex     bmi  children  smoker  region      charges
      0   19  0.0  27.900         0     1.0     3.0  16884.92400
      1   18  1.0  33.770         1     0.0     2.0   1725.55230
      2   28  1.0  33.000         3     0.0     2.0   4449.46200
      3   33  1.0  22.705         0     0.0     1.0  21984.47061
      4   32  1.0  28.880         0     0.0     1.0   3866.85520
```

```
[44]: # categorical data type
      oe.categories_
```

```
[44]: [array(['female', 'male'], dtype=object),
       array(['no', 'yes'], dtype=object),
       array(['northeast', 'northwest', 'southeast', 'southwest'], dtype=object)]
```