CHANDAN KUMAR

ID: GO_STP_13267

1. OneHotEncoding

OneHotEncoding is a process to convert string data into numeric data, but data should be categorical type.

2. Multicollinearity

It is a phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with accuracy.

It neither reduce the predictive power nor reliablility of the model as whole, at least within the sample data set; it only affects calculations regarding individual predictors

3. Dummy Variable

It is a scenario where there are attributes which are highly correlated and one varable predicts the value of others. When we use one hot encoding for handling the categorical data, then one attribute can be predicted with the help of other dummy variables.

4. Nominal Variable

It describes a variable with categories that do not have order or sequence.

Example: blood group, sex

5. Ordinal Variable

It describes the variable with order or sequence.

Example: Rating("poor", "bad", "neutral", "good", "very good")

```
1 import pandas as pd
2 url = "https://data.princeton.edu/wws509/datasets/salary.dat"
3 df = pd.read_csv(url, delim_whitespace = True)
```

```
1 df.head()
```

	sx	rk	yr	dg	yd	sl

```
1 df.columns
```

```
Index(['sx', 'rk', 'yr', 'dg', 'yd', 'sl'], dtype='object')
```

```
1 df.describe()
```

	yr	yd	sl
count	52.000000	52.000000	52.000000
mean	7.480769	16.115385	23797.653846
std	5.507536	10.222340	5917.289154
min	0.000000	1.000000	15000.000000
25%	3.000000	6.750000	18246.750000
50%	7.000000	15.500000	23719.000000
75%	11.000000	23.250000	27258.500000
max	25.000000	35.000000	38045.000000

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   sx      52 non-null     object
 1   rk      52 non-null     object
 2   yr      52 non-null     int64 
 3   dg      52 non-null     object
 4   yd      52 non-null     int64 
 5   sl      52 non-null     int64 
dtypes: int64(3), object(3)
memory usage: 2.6+ KB
```

```
1 df.corr()
```

	yr	yd	sl
yr	1.000000	0.638776	0.700669
yd	0.638776	1.000000	0.674854
sl	0.700669	0.674854	1.000000

```
1 df.isnull().sum()
```

```
sx    0
rk    0
yr    0
```

```
dg    0
yd    0
sl    0
dtype: int64
```

```
1 from sklearn.preprocessing import LabelEncoder
```

```
1 le = LabelEncoder()
2 df.sx = le.fit_transform(df.sx)
3 df.head()
```

	sx	rk	yr	dg	yd	sl
0	1	full	25	doctorate	35	36350
1	1	full	13	doctorate	22	35350
2	1	full	10	doctorate	23	28200
3	0	full	7	doctorate	27	26775
4	1	full	19	masters	30	33696

```
1 df.rk = le.fit_transform(df.rk)
2 df.head()
```

	sx	rk	yr	dg	yd	sl
0	1	2	25	doctorate	35	36350
1	1	2	13	doctorate	22	35350
2	1	2	10	doctorate	23	28200
3	0	2	7	doctorate	27	26775
4	1	2	19	masters	30	33696

```
1 df.dg = le.fit_transform(df.dg)
2 df.head()
```

	sx	rk	yr	dg	yd	sl
0	1	2	25	0	35	36350
1	1	2	13	0	22	35350
2	1	2	10	0	23	28200
3	0	2	7	0	27	26775
4	1	2	19	1	30	33696

✓ 0s completed at 5:11 PM