ASSIGNMENT / TASK 10

Discuss the concept of One-Hot-Encoding, Multicollinearity and the Dummy Variable Trap. What is Nominal and Ordinal Variables ?

ONE HOT ENCODING: One-hot encoding is essentially the representation of categorical variables as binary vectors. These categorical values are first mapped to integer values. Each integer value is then represented as a binary vector that is all 0s (except the index of the integer which is marked as 1).

when we want to convert string values into numerical values we first use level hot encoding to convert string values into some labels having some relation and then one hot encoding.

MULTiCOLLINEARITY: Multicollinearity is when two or more independent variables in a regression are highly related to one another, such that they do not provide unique or independent information to the regression.

DUMMY VARIABLE TRAP: We can draw one column from another column as they are highly corelated, for eg male and female, if we know female we already male, so we can drop one of the columns from male of female we always keep the columns that are corelated with each other, so this is called dummy variable trap.

NOMINAL VARIABLE: Categorical data are variables that contain label values rather than numeric values. The number of possible values is often limited to a fixed set. Categorical variables are often called nominal.

for eg: A “pet” variable with the values: “dog” and “cat“. A “color” variable with the values: “red“, “green“, and “blue“.

ORDINAL VARIABLE: When some categories may have a natural relationship to each other, such as a natural ordering then such variables are called ordinal variables.

for eg: A “place” variable with the values: “first“, “second“, and “third“. The “place” variable here does have a natural ordering of values. This type of categorical variable is called an ordinal variable because the values can be ordered or ranked.

Salary Dataset of 52 professors having categorical columns. Apply dummy variables concept and one-hot-encoding on categorical

columns.

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


url = "https://data.princeton.edu/wws509/datasets/salary.dat"
df = pd.read_csv(url, delim_whitespace = True)
df.head(10)
```

	sx	rk	yr	dg	yd	sl
0	male	full	25	doctorate	35	36350
1	male	full	13	doctorate	22	35350
2	male	full	10	doctorate	23	28200
3	female	full	7	doctorate	27	26775
4	male	full	19	masters	30	33696
5	male	full	16	doctorate	21	28516
6	female	full	0	masters	32	24900
7	male	full	16	doctorate	18	31909
8	male	full	13	masters	30	31850
9	male	full	13	masters	31	32850

```
dummies=pd.get_dummies(df.sx)
dummies.head()
```

	female	male
0	0	1
1	0	1
2	0	1
3	1	0
4	0	1

```
new_df=pd.concat([df,dummies],axis="columns")
new_df.head()
```

	sx	rk	yr	dg	yd	sl	female	male
0	male	full	25	doctorate	35	36350	0	1
1	male	full	13	doctorate	22	35350	0	1
2	male	full	10	doctorate	23	28200	0	1

```
new_df1=new_df.drop(['sx'],axis="columns")
```

```
new_df1.head()
```

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

```
dummies=pd.get_dummies(new_df1.rk)
dummies.head()
```

	assistant	associate	full
0	0	0	1
1	0	0	1
2	0	0	1
3	0	0	1
4	0	0	1

```
new_df2=pd.concat([new_df1,dummies],axis="columns")
new_df2.head()
```

```
new_df3=new_df2.drop(["rk"],axis="columns")
```

```
new_df3.head()
```

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

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
new_df4=new_df3
new_df4.dg=le.fit_transform(new_df4.dg)
new_df4.head()
```

	yr	dg	yd	sl	female	male	assistant	associate	full
0	25	0	35	36350	0	1	0	0	1
1	13	0	22	35350	0	1	0	0	1
2	10	0	23	28200	0	1	0	0	1
3	7	0	27	26775	1	0	0	0	1
4	19	1	30	33696	0	1	0	0	1

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
ct=ColumnTransformer([('dg',OneHotEncoder(),[1])],remainder="passthrough")
```

```
newct=ct.fit_transform(new_df4)#newct is a numpy array here
```

```
final_df=pd.DataFrame(newct,columns=["doctorate","masters","yr","yd","sl","female","male","as
final_df.head()
```

	doctorate	masters	yr	yd	sl	female	male	assistant	associate	full
0	1.0	0.0	25.0	35.0	36350.0	0.0	1.0	0.0	0.0	1.0
1	1.0	0.0	13.0	22.0	35350.0	0.0	1.0	0.0	0.0	1.0
2	1.0	0.0	10.0	23.0	28200.0	0.0	1.0	0.0	0.0	1.0
3	1.0	0.0	7.0	27.0	26775.0	1.0	0.0	0.0	0.0	1.0
4	0.0	1.0	19.0	30.0	33696.0	0.0	1.0	0.0	0.0	1.0

✓ 0s completed at 14:33