In [1]:

```
import pandas as pd
```

In [4]:

```
df=pd.read_csv(r"C:\Users\sagar\Desktop\dataset\PARTB-DATASETS\heart.csv",sep=',')
```

In [5]:

```
df.head()
```

Out[5]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

In [6]:

```
#Data Cleaning
df.isnull()
```

Out[6]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa
0	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...
298	False	False	False	False	False	False	False	False	False	False	False	False
299	False	False	False	False	False	False	False	False	False	False	False	False
300	False	False	False	False	False	False	False	False	False	False	False	False
301	False	False	False	False	False	False	False	False	False	False	False	False
302	False	False	False	False	False	False	False	False	False	False	False	False

303 rows × 14 columns

In [7]:

```
df.dropna(subset=['output'])
```

Out[7]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	outp
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	

303 rows × 14 columns

In [8]:

```
df.columns
```

Out[8]:

```
Index(['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg', 'thalachh',
       'exng', 'oldpeak', 'slp', 'caa', 'thall', 'output'],
      dtype='object')
```

In [10]:

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

Out[10]:

```
age         0
sex         0
cp          0
trtbps      0
chol        0
fbs         0
restecg     0
thalachh    0
exng        0
oldpeak     0
slp         0
caa         0
thall       0
output      0
dtype: int64
```

In [11]:

```
#no requirement of data cleaning
```

In [12]:

```
#Data Integration
from sklearn.impute import SimpleImputer
import numpy as np
```

In [13]:

```
df.columns
```

Out[13]:

```
Index(['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg', 'thalachh',
       'exng', 'oldpeak', 'slp', 'caa', 'thall', 'output'],
      dtype='object')
```

In [14]:

```
columns = ['sex', 'cp', 'exng', 'restecg']
```

In [15]:

```
imp = SimpleImputer(missing_values=np.NaN, strategy='mean')
```

In [16]:

```
df[columns] = imp.fit_transform(df[columns])
```

In [17]:

```
df[columns]
```

Out[17]:

	sex	cp	exng	restecg
0	1.0	3.0	0.0	0.0
1	1.0	2.0	0.0	1.0
2	0.0	1.0	0.0	0.0
3	1.0	1.0	0.0	1.0
4	0.0	0.0	1.0	1.0
...	...	...	...	...
298	0.0	0.0	1.0	1.0
299	1.0	3.0	0.0	1.0
300	1.0	0.0	0.0	1.0
301	1.0	0.0	1.0	1.0
302	0.0	1.0	0.0	0.0

303 rows × 4 columns

In [18]:

```
df[columns].isnull()
```

Out[18]:

	sex	cp	exng	restecg
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
...	...	...	...	...
298	False	False	False	False
299	False	False	False	False
300	False	False	False	False
301	False	False	False	False
302	False	False	False	False

303 rows × 4 columns

In [19]:

```
#Data Transformation
df.columns
```

Out[19]:

```
Index(['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg', 'thalachh',
       'exng', 'oldpeak', 'slp', 'caa', 'thall', 'output'],
      dtype='object')
```

In [21]:

```
df['output']
```

Out[21]:

```
0      1
1      1
2      1
3      1
4      1
      ..
298    0
299    0
300    0
301    0
302    0
Name: output, Length: 303, dtype: int64
```

In [22]:

```
df.head()
```

Out[22]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1.0	3.0	145	233	1	0.0	150	0.0	2.3	0	0	1	1
1	37	1.0	2.0	130	250	0	1.0	187	0.0	3.5	0	0	2	1
2	41	0.0	1.0	130	204	0	0.0	172	0.0	1.4	2	0	2	1
3	56	1.0	1.0	120	236	0	1.0	178	0.0	0.8	2	0	2	1
4	57	0.0	0.0	120	354	0	1.0	163	1.0	0.6	2	0	2	1

In [29]:

```
```

Out[29]:

```
0      1
1      1
2      1
3      1
4      1
      ..
298    0
299    0
300    0
301    0
302    0
Name: output, Length: 303, dtype: int64
```

In [ ]:

```
```

In [31]:

```
#numerical to categorical
df['output'].replace(0,np.nan) #replacing the value of NMHC(GT) first value with NAN
```

Out[31]:

```
0      1.0
1      1.0
2      1.0
3      1.0
4      1.0
      ...
298    NaN
299    NaN
300    NaN
301    NaN
302    NaN
Name: output, Length: 303, dtype: float64
```

In [44]:

```
############ Error Correcting
df.nunique()
```

Out[44]:

```
age          41
sex           2
cp            4
trtbps       49
chol        152
fbs           2
restecg       3
thalachh     91
exng          2
oldpeak      40
slp           3
caa           5
thall         4
output        2
dtype: int64
```

In [45]:

```
df['cp'].unique()
```

Out[45]:

```
array([3., 2., 1., 0.])
```

In [46]:

```
import numpy as np
```

In [47]:

```
p=df.loc[df['cp']==4, 'cp']=np.NaN
```

In [48]:

```
p
```

Out[48]:

```
nan
```

In [66]:

```
X = df[['oldpeak']]
Y = df[['slp']]
```

In [67]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
```

In [68]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.25 ,random_state=6)
```

In [69]:

```
X_train.shape
```

Out[69]:

```
(227, 1)
```

In [70]:

```
from sklearn.linear_model import LinearRegression

reg=LinearRegression()
reg.fit(X_train,Y_train)
predictions=reg.predict(X_test)
predictions
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
~\AppData\Local\Temp\ipykernel_9460\464781628.py in <module>
      2
      3 reg=LinearRegression()
----> 4 reg.fit(X_train,Y_train)
      5 predictions=reg.predict(X_test)
      6 predictions

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_base.py i
n fit(self, X, y, sample_weight)
    660         accept_sparse = False if self.positive else ["csr", "csc",
"coo"]
    661
--> 662         X, y = self._validate_data(
    663             X, y, accept_sparse=accept_sparse, y_numeric=True, mul
ti_output=True
    664         )

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py in _validate_da
ta(self, X, y, reset, validate_separately, **check_params)
    579                 y = check_array(y, **check_y_params)
    580             else:
--> 581                 X, y = check_X_y(X, y, **check_params)
    582             out = X, y
    583

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_X_y(X, y, accept_sparse, accept_large_sparse, dtype, order, copy, fo
rce_all_finite, ensure_2d, allow_nd, multi_output, ensure_min_samples, ens
ure_min_features, y_numeric, estimator)
    979     y = _check_y(y, multi_output=multi_output, y_numeric=y_numeri
c)
    980
--> 981     check_consistent_length(X, y)
    982
    983     return X, y

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_consistent_length(*arrays)
    330     uniques = np.unique(lengths)
    331     if len(uniques) > 1:
--> 332         raise ValueError(
    333             "Found input variables with inconsistent numbers of sa
mples: %r"
    334             % [int(l) for l in lengths]

ValueError: Found input variables with inconsistent numbers of samples: [2
27, 242]
```

In [65]:

```
import seaborn as sbn
sbn.regplot(x=X_test['sex'],y=predictions,scatter_kws={'s':10})
plt.scatter(X_test['sex'],Y_test, marker='+')
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
~\AppData\Local\Temp\ipykernel_9460\55632628.py in <module>
      1 import seaborn as sbn
----> 2 sbn.regplot(x=X_test['sex'],y=predictions,scatter_kws={'s':10})
      3 plt.scatter(X_test['sex'],Y_test, marker='+')

NameError: name 'predictions' is not defined
```

In [ ]: