

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
In [2]: df=pd.read_csv("C:\Users\VSHEREYANSH\machine_learning1\heart-attack\heart.csv")
df
```

```
Out[2]:
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows x 14 columns

```
In [3]: df. thall.max()
```

```
Out[3]: 3
```

```
In [4]: df.trtbps.max()
```

```
Out[4]: 200
```

```
In [5]: df.chol.max()
```

```
Out[5]: 564
```

```
In [6]: df.fbs.max()
```

```
Out[6]: 1
```

```
In [ ]:
```

```
In [7]: df.isnull()
```

```
Out[7]:
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	False	False	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	False	False
2	False	False	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	False	False
4	False	False	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	False	False
299	False	False	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	False	False
301	False	False	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	False	False

303 rows x 14 columns

```
In [8]: df.isnull().sum()
```

```
Out[8]: 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 [9]: df.duplicated().sum()
```

```
Out[9]: 1
```

```
In [10]: df.drop_duplicates()
```

```
Out[10]:
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

302 rows x 14 columns

```
In [11]: df.describe().T
```

```
Out[11]:
```

	count	mean	std	min	25%	50%	75%	max
age	303.0	54.366337	9.082101	29.0	47.5	55.0	61.0	77.0
sex	303.0	0.683168	0.466011	0.0	0.0	1.0	1.0	1.0
cp	303.0	0.966997	1.020252	0.0	0.0	1.0	2.0	3.0
trtbps	303.0	131.623762	17.538143	94.0	120.0	130.0	140.0	200.0
chol	303.0	246.264026	51.830751	126.0	211.0	240.0	274.5	564.0
fbs	303.0	1.048515	0.356198	0.0	0.0	0.0	0.0	1.0
restecg	303.0	0.528053	0.528053	0.0	0.0	1.0	1.0	2.0
thalachh	303.0	149.646895	22.905161	71.0	133.5	153.0	166.0	202.0
exng	303.0	0.326733	0.469794	0.0	0.0	0.0	1.0	1.0
oldpeak	303.0	1.039604	1.010075	0.0	0.0	0.8	1.6	6.2
slp	303.0	0.198604	0.420205	0.0	1.0	1.0	2.0	2.0
caa	303.0	0.729372	0.227606	0.0	0.0	0.0	1.0	1.0
thall	303.0	0.318531	0.613277	0.0	2.0	2.0	3.0	3.0
output	303.0	0.544554	0.498835	0.0	0.0	1.0	1.0	1.0

```
In [12]: plt.plot(df.age, df['output'])
```

```
Out[12]: <matplotlib.lines.Line2D at 0x208d133a208>
```

```
In [13]: plt.xlabel('age')
plt.ylabel('output')
plt.plot(df.age, df['output'])
```

```
Out[13]: <matplotlib.lines.Line2D at 0x208d13b4688>
```

```
In [14]: plt.xlabel('sex')
plt.ylabel('output')
plt.plot(df.sex, df['output'])
```

```
Out[14]: <matplotlib.lines.Line2D at 0x208d1423c48>
```

```
In [15]: plt.plot(df)
plt.show()
```

```
In [16]: df.head()
```

```
Out[16]:
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

```
In [17]: input_df = df.drop('output', axis = 1)
input_df.head()
```

```
Out[17]:
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall
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

```
In [18]: output_df = df[['output']]
output_df.head()
```

```
Out[18]:
```

	output
0	1
1	1
2	1
3	1
4	1
...	...
298	0
299	0
300	0
301	0
302	0

## Model building!!!

```
In [19]: from sklearn.model_selection import train_test_split
```

```
In [20]: input_train, input_test, output_train, output_test = train_test_split(input_df, output_df, t
est_size = 0.2)
print(len(input_train))
print(len(input_test))
print(len(output_train))
print(len(output_test))
```

```
242
61
242
61
```

```
In [21]: from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

```
In [22]: reg = LinearRegression()
```

```
reg.fit(input_train, output_train)
```

```
Out[22]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
In [23]: reg.predict(input_test)
```

```
Out[23]: array([[ 1.02429037],
 [ 0.7984587],
 [ 0.1160971 ],
 [ 0.68716018],
 [ 0.04909044],
 [ 0.00241119],
 [ 0.07931784],
 [ 0.2626222 ],
 [ 0.22776235],
 [ 0.09486883],
 [ 0.18599428],
 [ 1.06239162],
 [ 0.29165129],
 [ 0.41963537],
 [ 0.18522554],
 [ 0.15315917],
 [ 0.76589429],
 [ 0.71981594],
 [ 0.04808965],
 [ 0.61777921],
 [ 0.74108452],
 [ 0.68691461],
 [ 0.62383516],
 [ 0.09584968],
 [ 0.36846651],
 [ 0.69106787],
 [ 0.04574142],
 [ 0.91958461],
 [ 0.62676105],
 [ 1.11074041],
 [ 0.58375843],
 [ 0.87259667],
 [ 0.51694296],
 [ 0.21225941],
 [ 0.9975739 ],
 [ 0.52227614],
 [ 0.75322244],
 [ 0.58510897],
 [ 0.25158928],
 [ 0.20186753],
 [ 0.49435482],
 [ 0.77882939],
 [ 0.9920266 ],
 [ 0.09947758],
 [ 0.11889517],
 [ 0.20986327],
 [ 0.89393658],
 [ 0.48398982],
 [ 0.21897365],
 [ 1.3318272 ],
 [ 0.85248883],
 [ 0.14587721],
 [ 0.91262251],
 [ 0.74031945],
 [ 0.64293246],
 [ 0.20914397],
 [ 0.83318415],
 [ 0.12454956],
 [ 0.52946201],
 [ 0.77902986],
 [ 0.99962871]])
```

```
In [24]: reg.score(input_test, output_test)
```

```
Out[24]: 0.4674924418136177
```

```
In [25]: df.head()
```

```
Out[25]:
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

```
In [26]: df.output.unique()
```