

```

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
df=pd.read_csv('/content/heart.csv')
h=df.head(10)
t=df.tail(10)
print(df.shape)
for i in range(293,283,-1):
    df.drop([i],axis=0,inplace=True)
for i in range(10):
    df.drop([i],axis=0,inplace=True)
dfmt=pd.concat([h,t],axis=0)
dfmt.to_csv("test_file.csv")
data=pd.read_csv('/content/test_file.csv')
print(data)

```

```

(303, 14)

```

	Unnamed: 0	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	\
0	0	63	1	3	145	233	1	0	150	0	
1	1	37	1	2	130	250	0	1	187	0	
2	2	41	0	1	130	204	0	0	172	0	
3	3	56	1	1	120	236	0	1	178	0	
4	4	57	0	0	120	354	0	1	163	1	
5	5	57	1	0	140	192	0	1	148	0	
6	6	56	0	1	140	294	0	0	153	0	
7	7	44	1	1	120	263	0	1	173	0	
8	8	52	1	2	172	199	1	1	162	0	
9	9	57	1	2	150	168	0	1	174	0	
10	293	67	1	2	152	212	0	0	150	0	
11	294	44	1	0	120	169	0	1	144	1	
12	295	63	1	0	140	187	0	0	144	1	
13	296	63	0	0	124	197	0	1	136	1	
14	297	59	1	0	164	176	1	0	90	0	
15	298	57	0	0	140	241	0	1	123	1	
16	299	45	1	3	110	264	0	1	132	0	
17	300	68	1	0	144	193	1	1	141	0	
18	301	57	1	0	130	131	0	1	115	1	
19	302	57	0	1	130	236	0	0	174	0	

	oldpeak	slp	caa	thall	output
0	2.3	0	0	1	1
1	3.5	0	0	2	1
2	1.4	2	0	2	1
3	0.8	2	0	2	1
4	0.6	2	0	2	1
5	0.4	1	0	1	1
6	1.3	1	0	2	1
7	0.0	2	0	3	1
8	0.5	2	0	3	1
9	1.6	2	0	2	1
10	0.8	1	0	3	0
11	2.8	0	0	1	0
12	4.0	2	2	3	0
13	0.0	1	0	2	0
14	1.0	1	2	1	0
15	0.2	1	0	3	0
16	1.2	1	0	3	0
17	3.4	1	2	3	0
18	1.2	1	1	3	0
19	0.0	1	1	2	0

```

data=data.sample(frac=1)
data.head(10)
x=data[["chol"]]
y=data["output"]

```

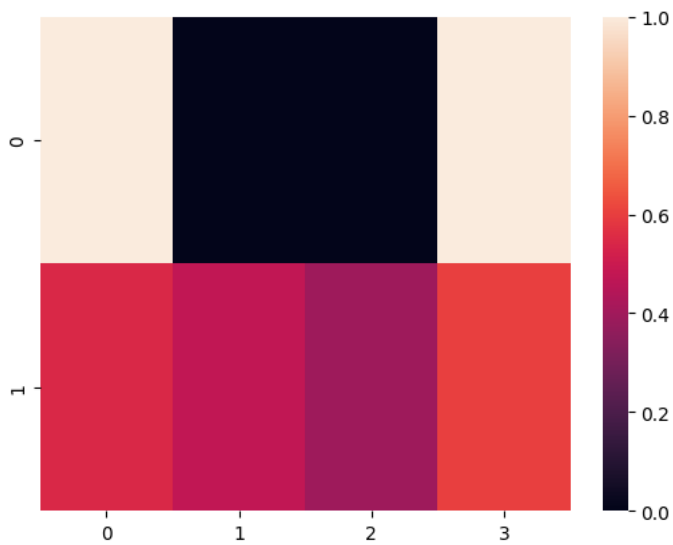
```

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
model = LinearRegression()
model.fit(X_train, y_train)
acc=model.score(X_test,y_test).round(2)
y_pred = model.predict(X_test)

```

```
print(acc, '\n', y_pred)
sns.heatmap([y_test, y_pred])
```

```
0.25
[0.54336634 0.47430904 0.39209797 0.59926987]
<Axes: >
```



```
model = LogisticRegression()
model.fit(X_train, y_train)
acc=model.score(X_test,y_test).round(2)
y_pred = model.predict(X_test)
print(acc, '\n', y_pred)
```

```
1.0
[1 0 0 1]
```

```
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test,y_pred)
sns.heatmap(cm,annot=True)
```

