In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df=pd.read_csv('heart.csv')
df
```

Out[2]:

	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 [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trtbps    303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalachh  303 non-null    int64  
 8   exng      303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64  
 11  caa       303 non-null    int64  
 12  thall     303 non-null    int64  
 13  output    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

In [4]:

```
df.describe()
```

Out[4]:

	age	sex	cp	trtbps	chol	fbs	restecg	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	3
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	1
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	1
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	1
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	1
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	2

In [5]:

```
sns.pairplot(df)
```

Out[5]:

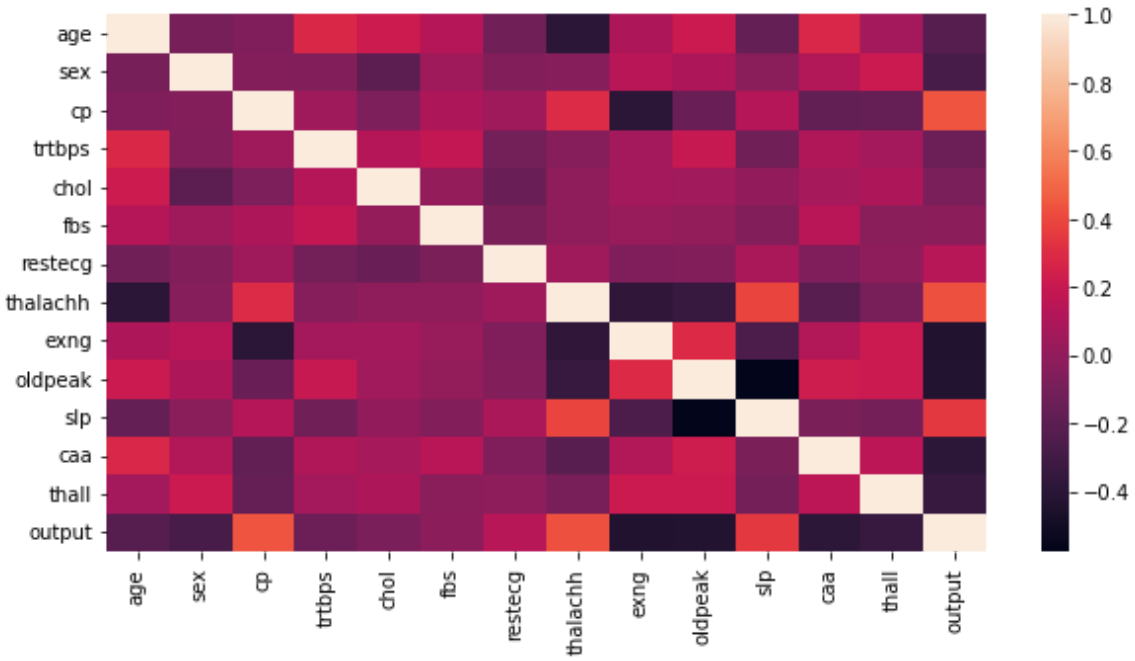
```
<seaborn.axisgrid.PairGrid at 0x22a7c75bb20>
```

In [6]:

```
plt.figure(figsize=(10,5))
sns.heatmap(df.corr())
```

Out[6]:

```
<AxesSubplot:>
```



In [7]:

```
df.corr()
```

Out[7]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762
trtbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123
thalachh	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000
exng	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187
slp	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784
caa	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177
thall	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439
output	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741

In [8]:

```
df.corr().style.background_gradient()
```

Out[8]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762
trtbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123
thalachh	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000
exng	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187
slp	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784
caa	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177
thall	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439
output	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trtbps    303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalachh  303 non-null    int64  
 8   exng      303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64  
 11  caa       303 non-null    int64  
 12  thall     303 non-null    int64  
 13  output    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

In [10]:

```
df.head()
```

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

In [11]:

```
#x=df.iloc[:,-1]
#y=df.iloc[:,0]

x=df.drop("age",axis=1)
y=df['age']
```

In [12]:

```
x
```

Out[12]:

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

303 rows × 13 columns

In [13]:

```
y
```

Out[13]:

```
0      63
1      37
2      41
3      56
4      57
       ..
298    57
299    45
300    68
301    57
302    57
Name: age, Length: 303, dtype: int64
```

In [14]:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

In [15]:

```
xtrain.shape
```

Out[15]:

```
(212, 13)
```

In [16]:

```
ytrain.shape
```

Out[16]:

```
(212,)
```

In [17]:

```
from sklearn.linear_model import LinearRegression
linreg=LinearRegression()
linreg.fit(xtrain,ytrain)
ypred=linreg.predict(xtest)
```

In [18]:

```
from sklearn.metrics import r2_score
```

In [19]:

```
print(r2_score(ytest,ypred))
```

```
0.29364229506672135
```

In [20]:

```
train=linreg.score(xtrain,ytrain)
test=linreg.score(xtest,ytest)
```

In [21]:

```
print(f'training result:-{train}')
print(f' testing result:-{test}')
```

```
training result:-0.2936108095627338
 testing result:-0.29364229506672135
```

In [22]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [23]:

```
l2=Ridge(alpha=10)
l2.fit(xtrain,ytrain)
ypred=l2.predict(xtest)
```

In [24]:

```
train=linreg.score(xtrain,ytrain)
test=linreg.score(xtest,ytest)
print(f'training result:-{train}')
print(f' testing result:-{test}')
```

```
training result:-0.2936108095627338
 testing result:-0.29364229506672135
```

Hypermarameter Tuning

In [25]:

```
for i in range(1,50):
    l2 = Ridge(alpha=i)
    l2.fit(xtrain,ytrain)

    train = l2.score(xtrain,ytrain)
    test =l2.score(xtest,ytest)

    print(f"{i} {train} {test}")
```

```
1 0.2935971581266008 0.29416184376846066
2 0.29355983327456703 0.29462038323707773
3 0.29350341111572387 0.2950271475568905
4 0.29343153051322923 0.2953896656821392
5 0.2933471075586994 0.2957141289477023
6 0.2932524953544907 0.29600566848564014
7 0.29314960446787963 0.2962685671680434
8 0.2930399947247613 0.296506423319285
9 0.2929249458591162 0.2967222784549918
10 0.2928055123861598 0.29691871788109314
11 0.2926825665809599 0.2970979505998649
12 0.2925568324022715 0.29726187328356224
13 0.29242891246145053 0.29741212186944166
14 0.2922993096047719 0.2975501134559879
15 0.29216844429144484 0.2976770805402541
16 0.29203666866638633 0.2977940991627872
17 0.29190427801700336 0.29790211217298324
18 0.29177152014644003 0.298001948561177
19 0.29163860307755685 0.29809433960114085
20 0.29150570141213783 0.29817993239141605
21 0.2913729616011034 0.2982593012640292
22 0.29124050632855125 0.2983329574359702
23 0.2911084381713598 0.29840135720583716
24 0.2909768426640127 0.2984649089406255
25 0.29084579087309814 0.29852397905208883
26 0.29071534156605405 0.2985788971258261
27 0.29058554304291406 0.2986299603371795
28 0.29045643468721094 0.29867743726461626
29 0.2903280482820787 0.29872157119232534
30 0.29020040912944545 0.29876258297835645
31 0.29007353700362204 0.2988006735520574
32 0.28994744696523855 0.298836026094248
33 0.28982215005710366 0.2988688079450804
34 0.2896976538999958 0.29889917227752094
35 0.2895739632034503 0.2989272595685638
36 0.2894510802041842 0.29895319889544103
37 0.2893290050427928 0.2989771090800525
38 0.2892077360876981 0.2989990997014328
39 0.28908727021392977 0.29901927199324396
40 0.2889676030431747 0.29903771964085324
41 0.2888487291505596 0.29905452949054434
42 0.2887306422428132 0.2990697821816719
43 0.288613335311784 0.2990835527111183
44 0.2884968007666968 0.29909591093814614
45 0.2883810305480574 0.29910692203669
46 0.28826601622569215 0.2991166469012041
47 0.2881517490830672 0.29912514251140976
48 0.28803822018972125 0.2991324622605972
49 0.2879254204634073 0.29913865625156966
```

In [26]:

```
l2 = Ridge(alpha=39)
l2.fit(xtrain,ytrain)
ypred = l2.predict(xtest)
```

In [27]:

```
from sklearn.linear_model import LinearRegression
linreg = LinearRegression()
l2.fit(xtrain,ytrain)
ypred = l2.predict(xtest)
```

In [28]:

```
print(r2_score(ytest,ypred))
```

```
0.29901927199324396
```

svm

In [29]:

```
df.head()
```

Out[29]:

	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 [31]:

```
df.corr().style.background_gradient()
```

Out[31]:

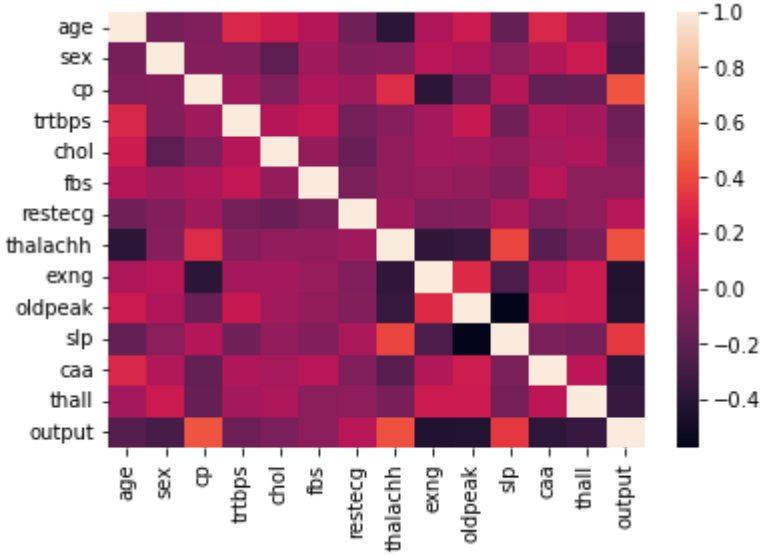
	age	sex	cp	trtbps	chol	fbs	restecg	thalachh
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762
trtbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123
thalachh	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000
exng	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187
slp	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784
caa	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177
thall	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439
output	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741

In [32]:

```
sns.heatmap(df.corr())
```

Out[32]:

```
<AxesSubplot:>
```



In [ ]:

```
```