

untitled4

May 7, 2024

```
[35]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn import metrics
```

```
[3]: df = pd.read_csv('train.csv')
```

```
[4]: df
```

```
[4]:
```

	ID	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	\
0	1	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	
1	2	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	
2	4	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	
3	5	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	
4	7	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	
..	
328	500	0.17783	0.0	9.69	0	0.585	5.569	73.5	2.3999	6	391	
329	502	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	
330	503	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	
331	504	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	
332	506	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	
	ptratio	black	lstat	medv								
0	15.3	396.90	4.98	24.0								
1	17.8	396.90	9.14	21.6								
2	18.7	394.63	2.94	33.4								
3	18.7	396.90	5.33	36.2								
4	15.2	395.60	12.43	22.9								
..								
328	19.2	395.77	15.10	17.5								
329	21.0	391.99	9.67	22.4								
330	21.0	396.90	9.08	20.6								

```
331      21.0  396.90   5.64  23.9
332      21.0  396.90   7.88  11.9
```

```
[333 rows x 15 columns]
```

```
[5]: df.isnull().sum()
```

```
[5]: ID          0
     crim        0
     zn          0
     indus       0
     chas        0
     nox         0
     rm          0
     age         0
     dis         0
     rad         0
     tax         0
     ptratio     0
     black       0
     lstat       0
     medv        0
     dtype: int64
```

```
[6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 333 entries, 0 to 332
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   ID          333 non-null   int64
 1   crim        333 non-null   float64
 2   zn          333 non-null   float64
 3   indus       333 non-null   float64
 4   chas        333 non-null   int64
 5   nox         333 non-null   float64
 6   rm          333 non-null   float64
 7   age         333 non-null   float64
 8   dis         333 non-null   float64
 9   rad         333 non-null   int64
10   tax         333 non-null   int64
11   ptratio     333 non-null   float64
12   black       333 non-null   float64
13   lstat       333 non-null   float64
14   medv        333 non-null   float64
dtypes: float64(11), int64(4)
```

memory usage: 39.2 KB

```
[9]: x=df[['ID','crim','zn','indus','chas','nox','rm','age','dis','rad','tax','ptratio','black','ls
```

```
[12]: y=df[['medv']]
```

```
[14]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.  
↳25,random_state=42)
```

```
[15]: model = LinearRegression()
```

```
[16]: model
```

```
[16]: LinearRegression()
```

```
[17]: model.fit(x_train,y_train)
```

```
[17]: LinearRegression()
```

```
[18]: pred = model.predict(x_test)  
pred
```

```
[18]: array([[25.42935235],  
[22.96355416],  
[22.98361313],  
[32.78417799],  
[25.00393879],  
[14.02939811],  
[17.33416663],  
[30.26235415],  
[15.66094194],  
[25.17645124],  
[26.57561017],  
[19.87047671],  
[20.0909698 ],  
[34.65458829],  
[21.54826488],  
[34.56267042],  
[22.2485164 ],  
[16.882139  ],  
[25.36436429],  
[16.87727359],  
[36.17705665],  
[31.39060697],  
[22.8003689 ],  
[28.13000873],  
[17.20049765],
```

[42.37151734],
[7.87497041],
[-0.36689721],
[31.09480595],
[8.4983594],
[19.11933223],
[20.39461378],
[27.46125616],
[15.1123262],
[19.49023076],
[12.24379213],
[27.56523215],
[4.49445516],
[17.2050638],
[22.53286675],
[24.31038054],
[22.18772176],
[25.17245639],
[39.17706326],
[36.30722879],
[21.36801908],
[11.07627164],
[21.45992975],
[13.86557089],
[20.5226888],
[13.18677931],
[27.99316461],
[21.49698121],
[13.63192997],
[33.57300717],
[1.31418106],
[20.80538627],
[27.16216548],
[25.27755635],
[28.15722941],
[17.87759693],
[23.94807788],
[18.11868011],
[28.38039855],
[23.49353731],
[16.58474878],
[26.43944232],
[20.85487585],
[23.73111221],
[16.67730384],
[14.83654746],
[26.14692568],

```
[21.12274293],  
[12.8361667 ],  
[24.2684098 ],  
[33.5090989 ],  
[22.02205101],  
[23.54787016],  
[22.92262304],  
[16.50548743],  
[21.16265595],  
[28.70685258],  
[40.24134644],  
[15.29564422]])
```

```
[22]: model.score(x_test,y_test)
```

```
[22]: 0.7257587357992887
```

```
[23]: model.score(x_train,y_train)
```

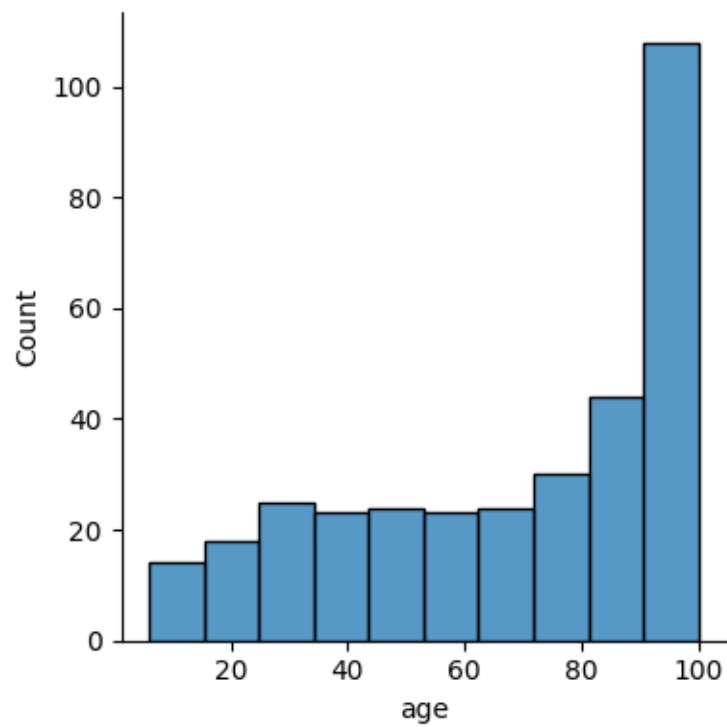
```
[23]: 0.7322764285677805
```

```
[27]: print('MSE', metrics.mean_squared_error(y_test,pred))
```

```
MSE 23.26809178579522
```

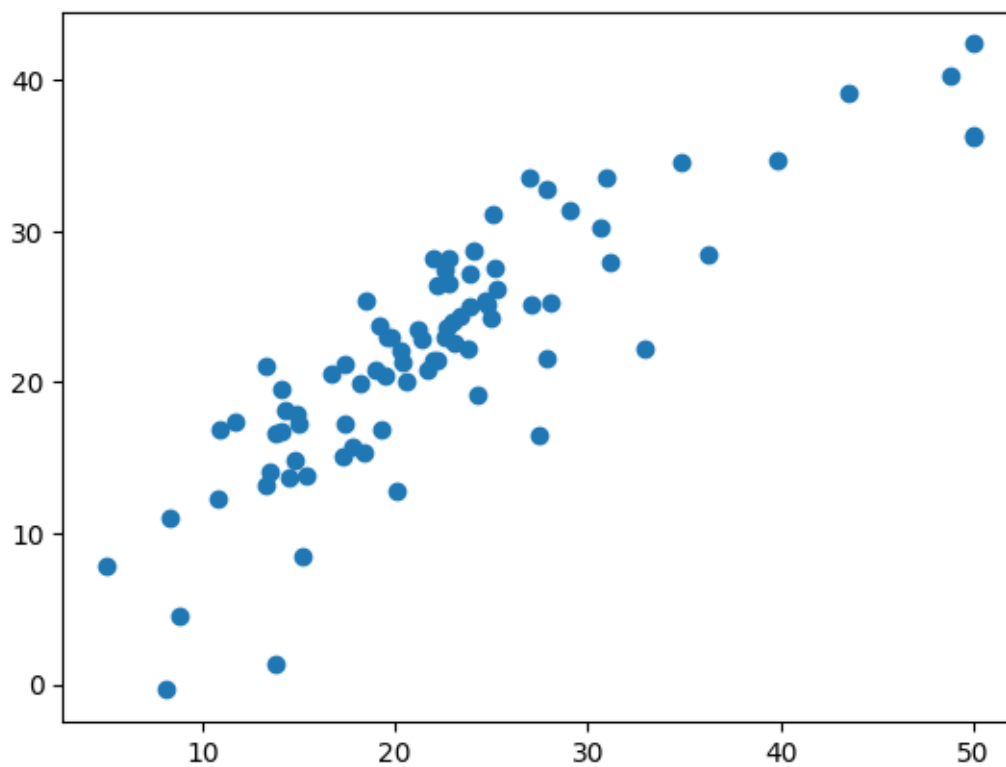
```
[28]: sns.displot(x='age',data=df,height=4)
```

```
[28]: <seaborn.axisgrid.FacetGrid at 0x2d24d12f5c0>
```



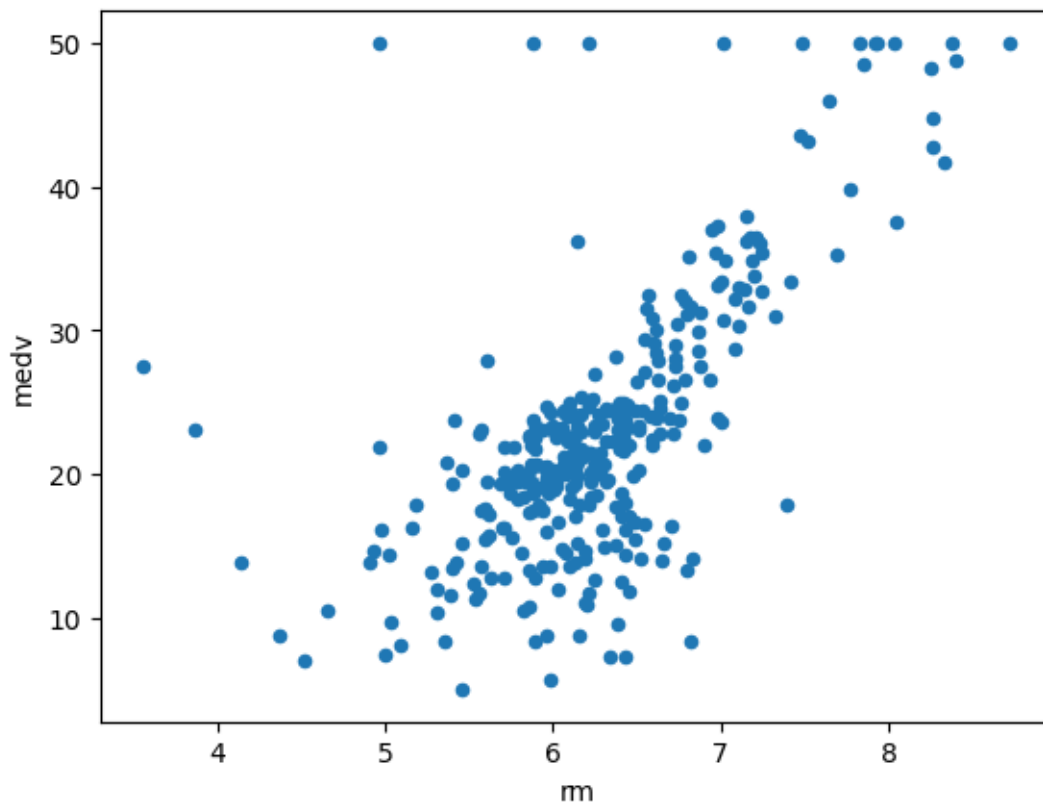
```
[36]: plt.scatter(y_test, pred)
```

```
[36]: <matplotlib.collections.PathCollection at 0x2d25796cdd0>
```



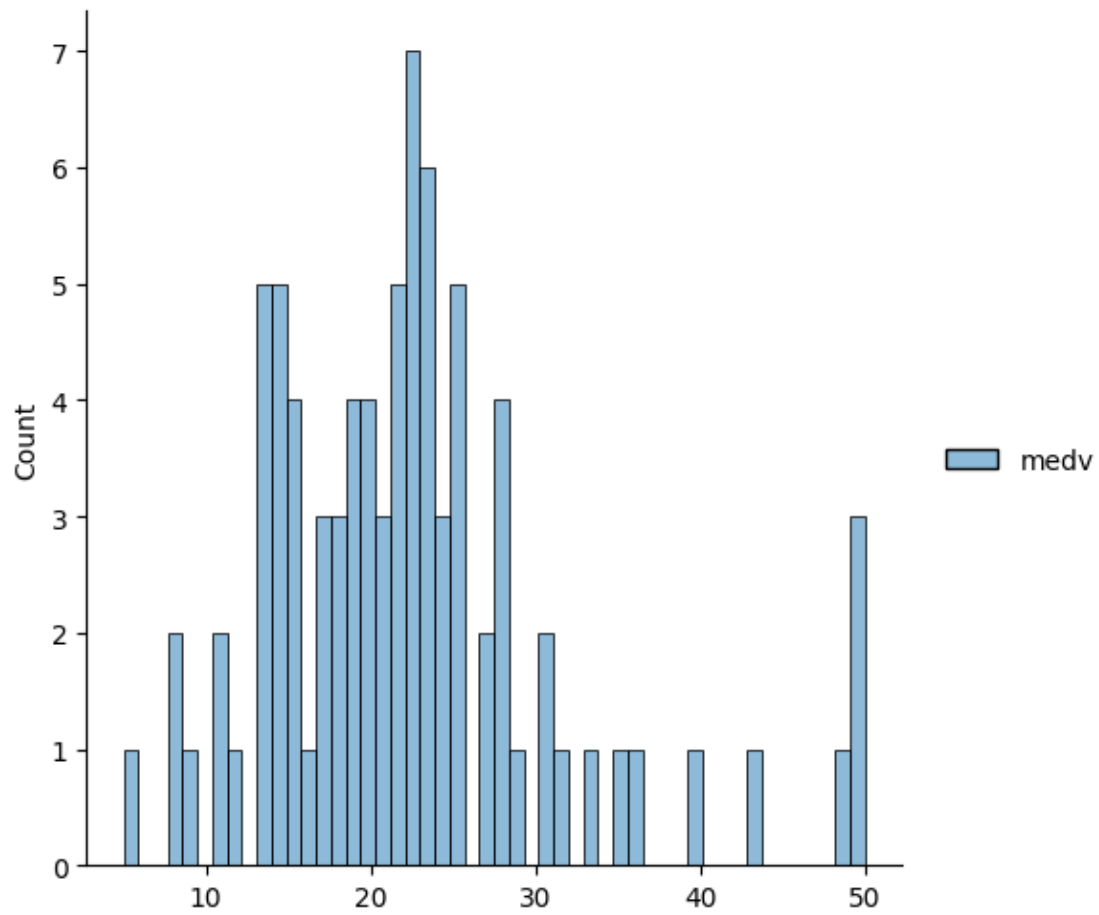
```
[37]: df.plot.scatter('rm', 'medv')
```

```
[37]: <Axes: xlabel='rm', ylabel='medv'>
```



```
[38]: sns.displot((y_test),bins=50)
```

```
[38]: <seaborn.axisgrid.FacetGrid at 0x2d256bca9c0>
```

```
[ ]: coe
```