RK1

April 19, 2021

1 RK1 Alekhin Sergey IU6-61

1.0.1 Library import

[14]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import warnings
from sklearn import datasets
from sklearn.datasets import load_boston
from sklearn import linear_model
from sklearn.cluster import KMeans
from sklearn import metrics
from pandas import DataFrame
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

[2]:
```
boston = load_boston()
data = pd.DataFrame(boston.data, columns=boston.feature_names)
data['TARGET'] = boston.target
```

[3]:
```
data.head()
```

[3]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	TARGET
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0				

```
 4     18.7  396.90   5.33    36.2
```

[4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    float64
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    float64
 9   TAX      506 non-null    float64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  TARGET   506 non-null    float64
dtypes: float64(14)
memory usage: 55.5 KB
```

[5]:
```
data.describe()
```

[5]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	\
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	

	AGE	DIS	RAD	TAX	PTRATIO	B	\
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	
mean	68.574901	3.795043	9.549407	408.237154	18.455534	356.674032	
std	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864	
min	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	
25%	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500	
50%	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000	
75%	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000	
max	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	

```
             LSTAT      TARGET
count  506.000000  506.000000
mean    12.653063   22.532806
std      7.141062    9.197104
min      1.730000    5.000000
25%      6.950000   17.025000
50%     11.360000   21.200000
75%     16.955000   25.000000
max     37.970000   50.000000
```

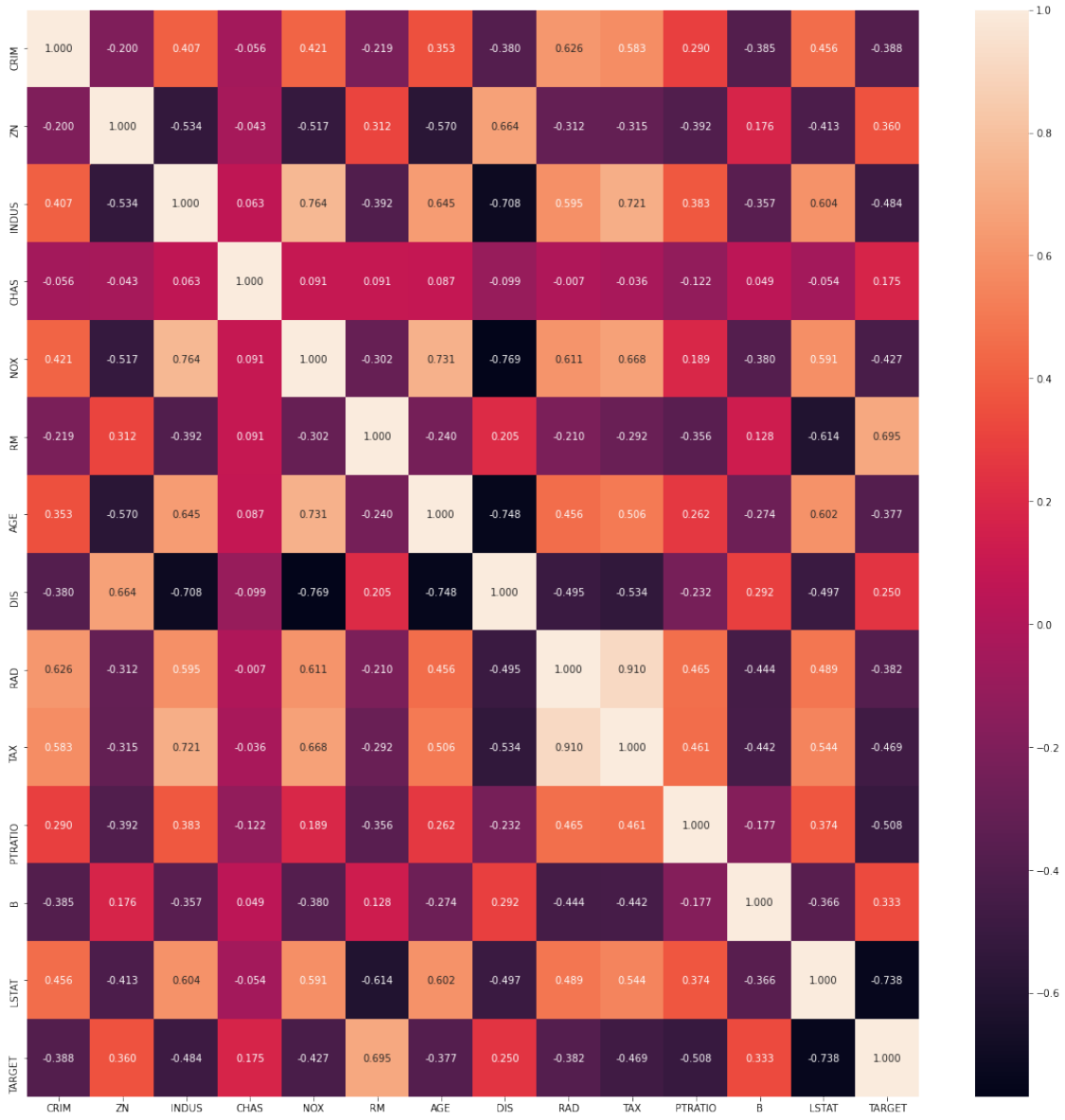
[6]:
```
corr_matrix = data.corr()
```

[7]:
```
corr_matrix['TARGET']
```

[7]:
```
CRIM      -0.388305
ZN         0.360445
INDUS     -0.483725
CHAS       0.175260
NOX       -0.427321
RM         0.695360
AGE       -0.376955
DIS        0.249929
RAD       -0.381626
TAX       -0.468536
PTRATIO   -0.507787
B          0.333461
LSTAT     -0.737663
TARGET     1.000000
Name: TARGET, dtype: float64
```

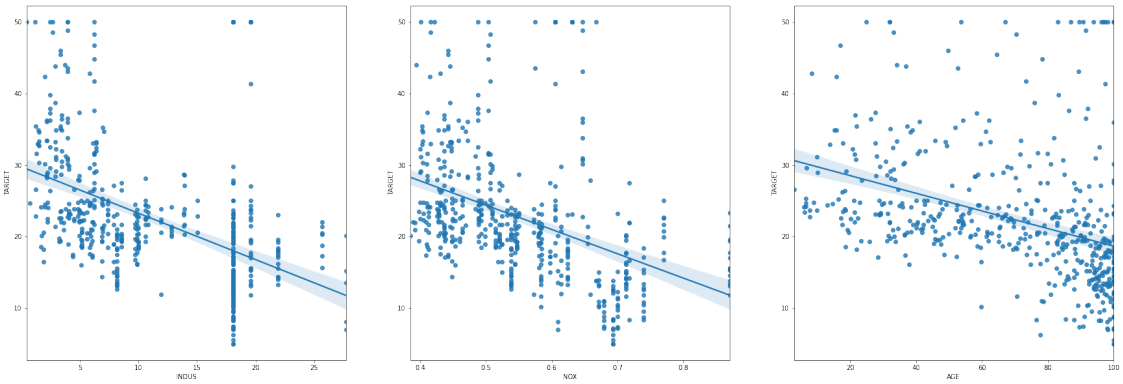
[8]:
```
plt.figure(figsize=(20,20))
sns.heatmap(corr_matrix, annot=True, fmt='.3f')
```

[8]:
```
<AxesSubplot:>
```



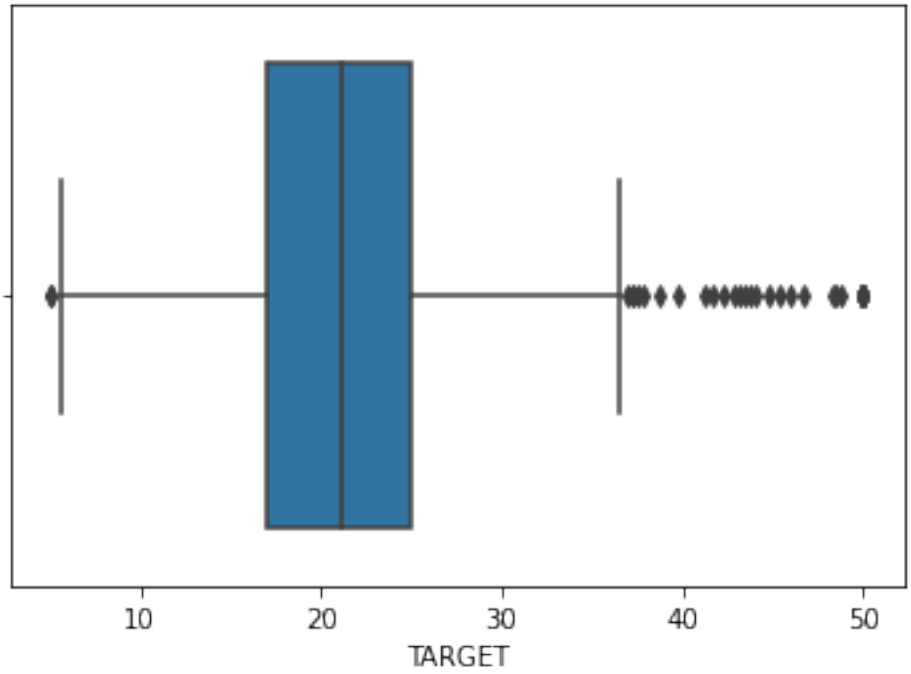
[9]:
```
fig, axs = plt.subplots(ncols=3, figsize=(30,10))
sns.regplot(x=data['INDUS'], y=data['TARGET'], ax = axs[0])
sns.regplot(x=data['NOX'], y=data['TARGET'], ax = axs[1])
sns.regplot(x=data['AGE'], y=data['TARGET'], ax = axs[2])
```

[9]: <AxesSubplot:xlabel='AGE', ylabel='TARGET'>



[10]:
```
sns.boxplot(x=data['TARGET'])
```

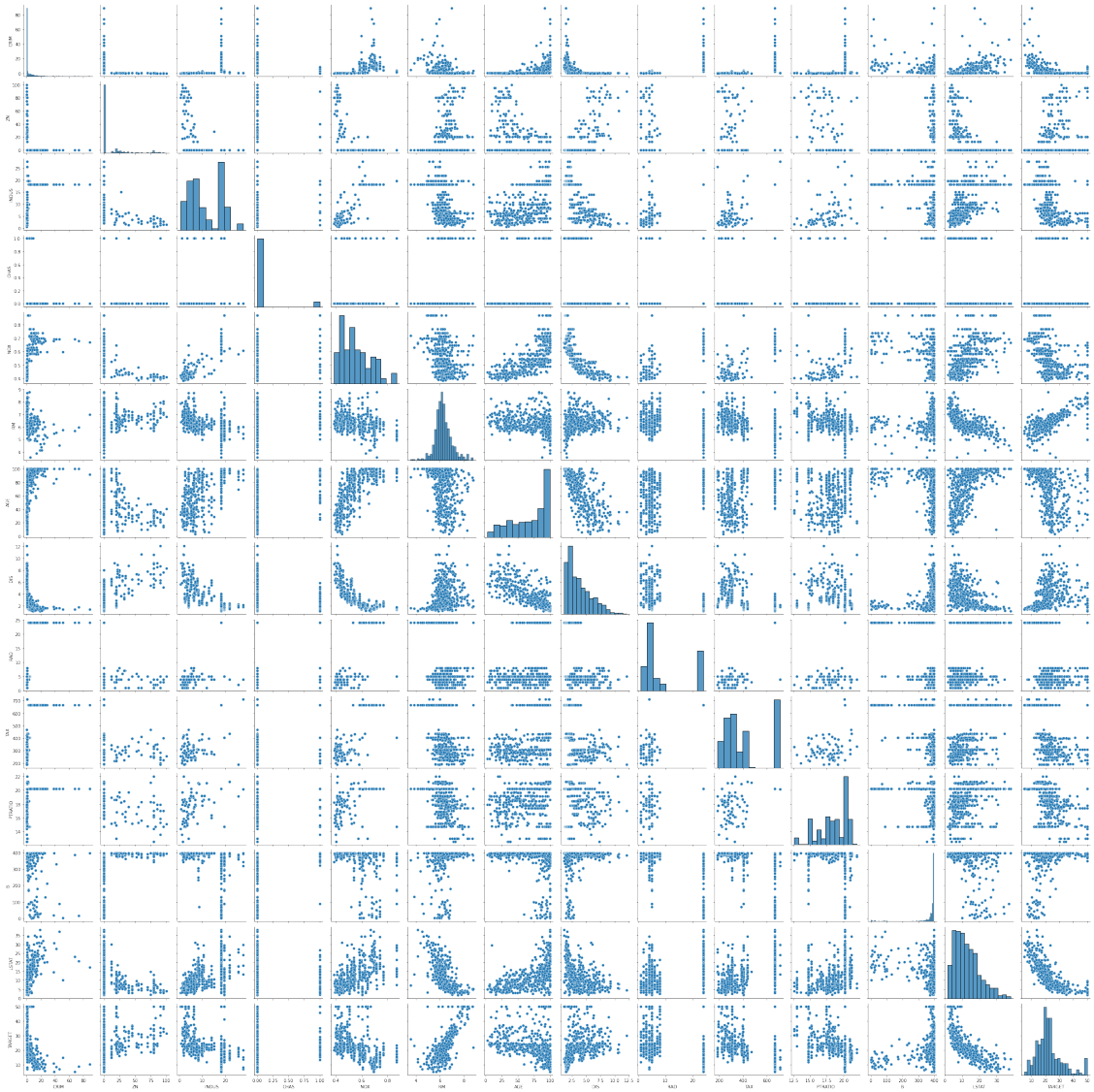
[10]: <AxesSubplot:xlabel='TARGET'>



[11]:
```
plt.figure(figsize=(12,6))
sns.pairplot(data)
```

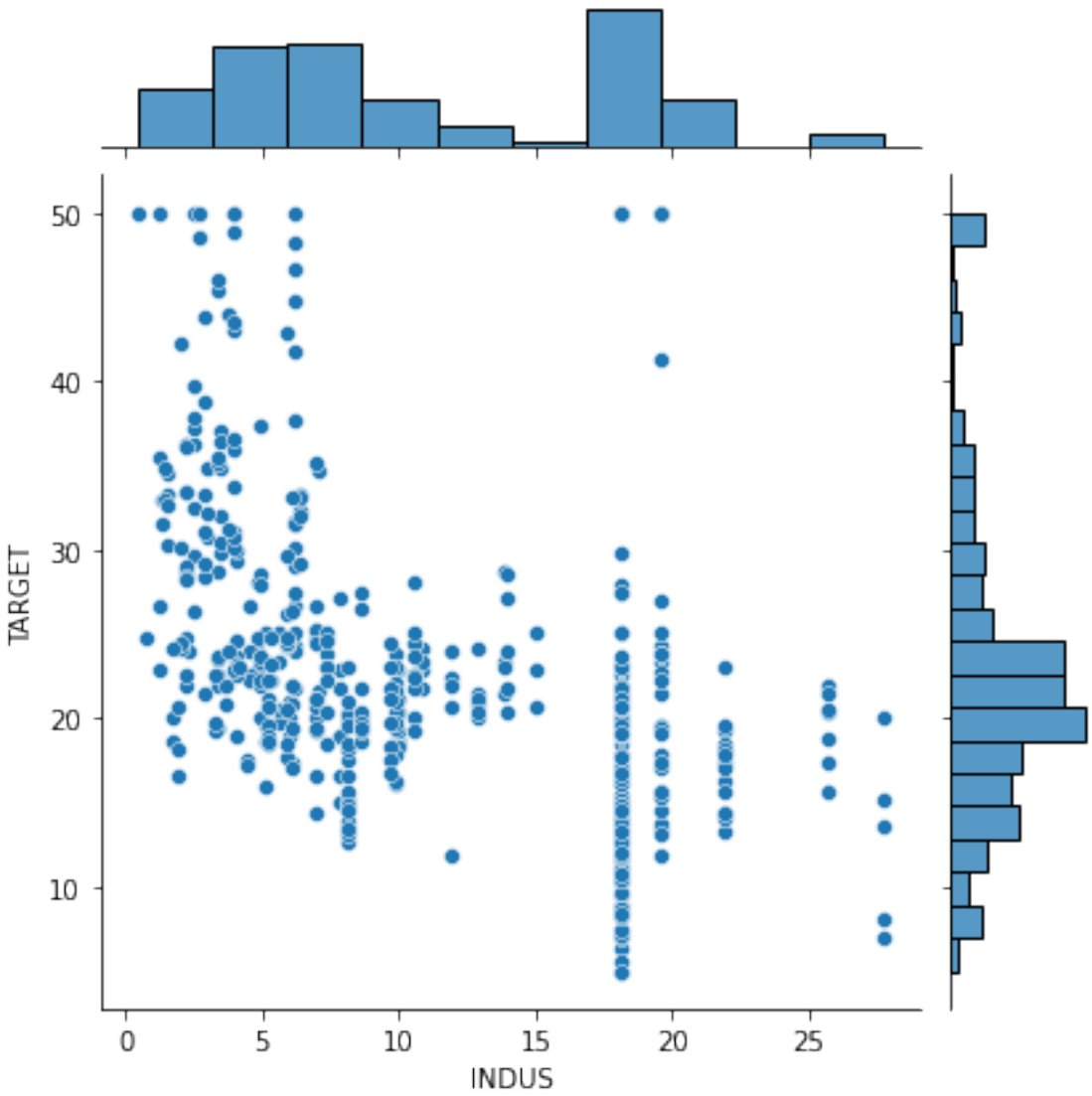
[11]: <seaborn.axisgrid.PairGrid at 0x7f9a3f5f8d00>

<Figure size 864x432 with 0 Axes>



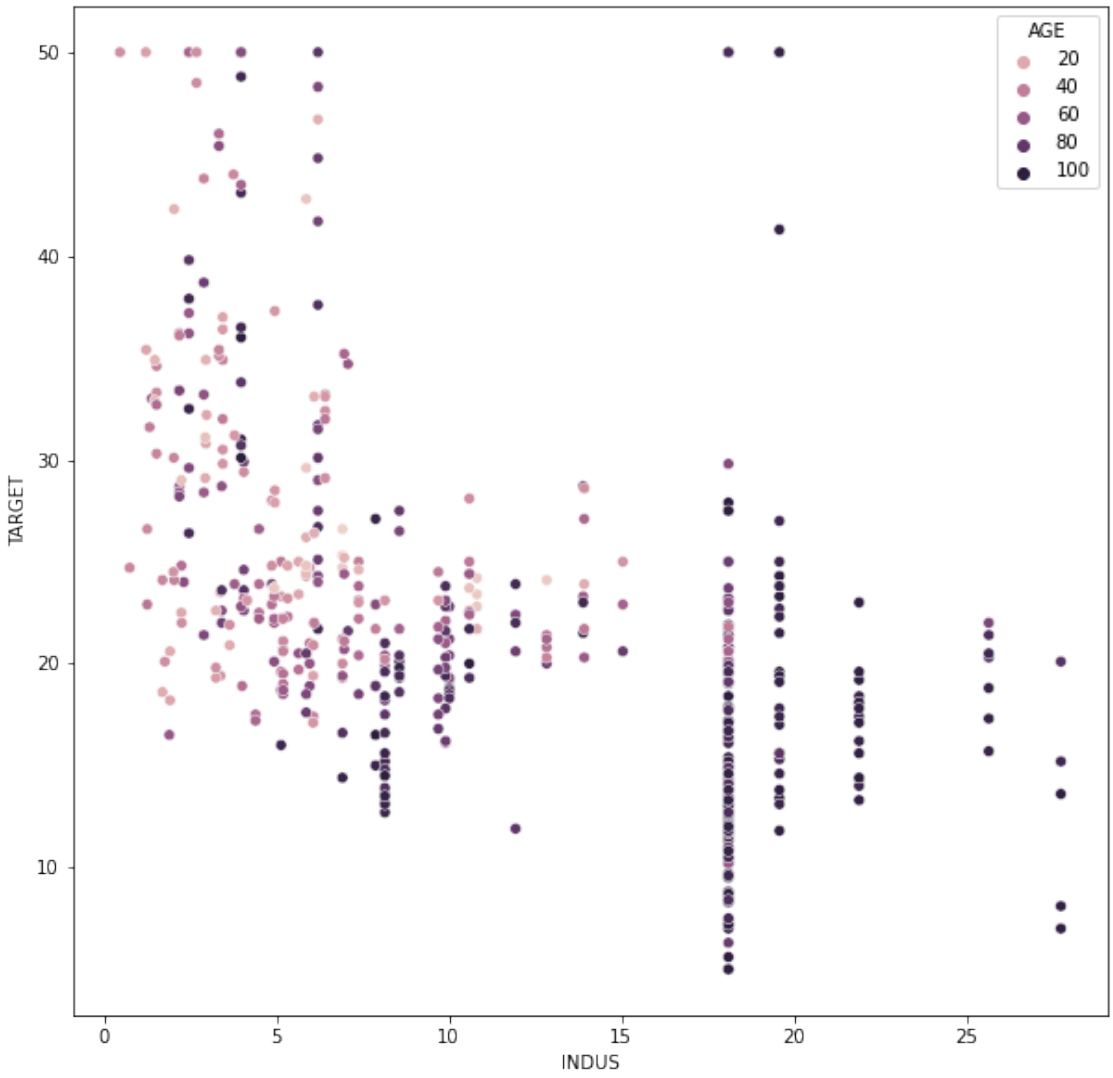
[12]:
```
sns.jointplot(x = "INDUS", y = "TARGET", kind="scatter", data = data)
```

[12]: <seaborn.axisgrid.JointGrid at 0x7f9a2c709df0>



[13]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='INDUS', y='TARGET', data=data, hue='AGE')
```

[13]: <AxesSubplot:xlabel='INDUS', ylabel='TARGET'>


AGE
20
40
60
80
100
TARGET
INDUS
50
40
30
20
10
0
5
10
15
20
25