rachittoshniwal / **machineLearning**

Code Issues Pull requests Actions Projects Wiki Security

master

machineLearning / standard scaler.ipynb

rachittoshniwal Add files via upload History

1 contributor

617 lines (617 sloc) | 194 KB

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import seaborn as sns
from sklearn.preprocessing import StandardScaler
```

In [2]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.pipeline import Pipeline
```

In [3]:
```
from sklearn.datasets import fetch_california_housing
```

In [4]:
```
X, y = fetch_california_housing(as_frame=True, return_X_y=True)
```

In [6]:
```
X.shape
```

Out[6]: (20640, 8)

In [7]:
```
X.head()
```

Out[7]:

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	L
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-

In [15]:
```
X = X.iloc[:, :-2]
```

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

Out[16]:
```
0    4.526
1    3.585
2    3.521
3    3.413
4    3.422
Name: MedHouseVal, dtype: float64
```

In [17]:
```
X.describe()
```

Out[17]:

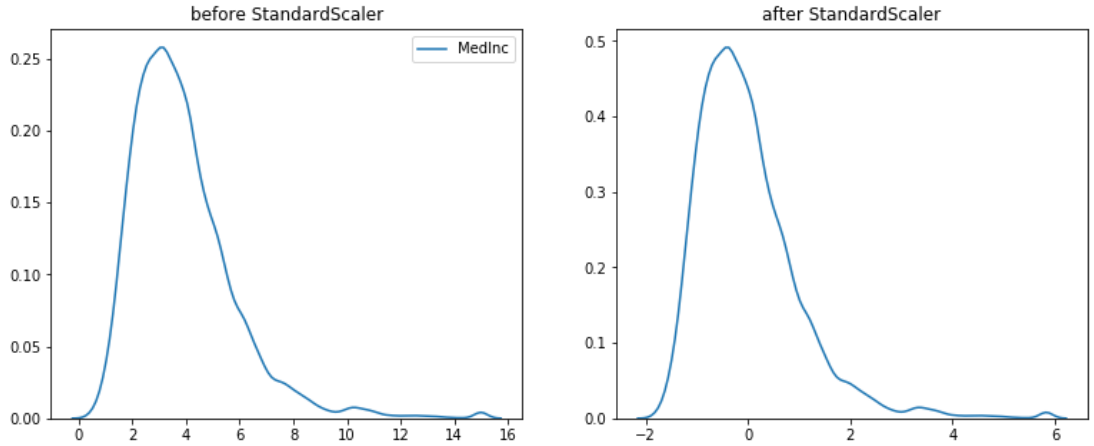
	MedInc	HouseAge	AveRooms	AveBedrms	Population	A
count	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	2
mean	3.870671	28.639486	5.429000	1.096675	1425.476744	3
std	1.899822	12.585558	2.474173	0.473911	1132.462122	1

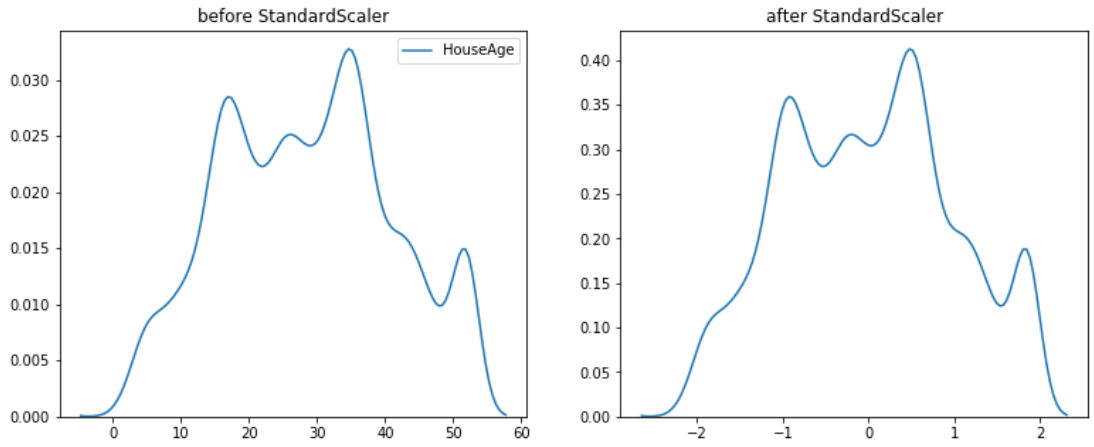
std	1.899822	12.585558	2.474173	0.473911	1132.462122	1
min	0.499900	1.000000	0.846154	0.333333	3.000000	0
25%	2.563400	18.000000	4.440716	1.006079	787.000000	2
50%	3.534800	29.000000	5.229129	1.048780	1166.000000	2
75%	4.743250	37.000000	6.052381	1.099526	1725.000000	3
max	15.000100	52.000000	141.909091	34.066667	35682.000000	1

In [18]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2
, random_state=0)
```

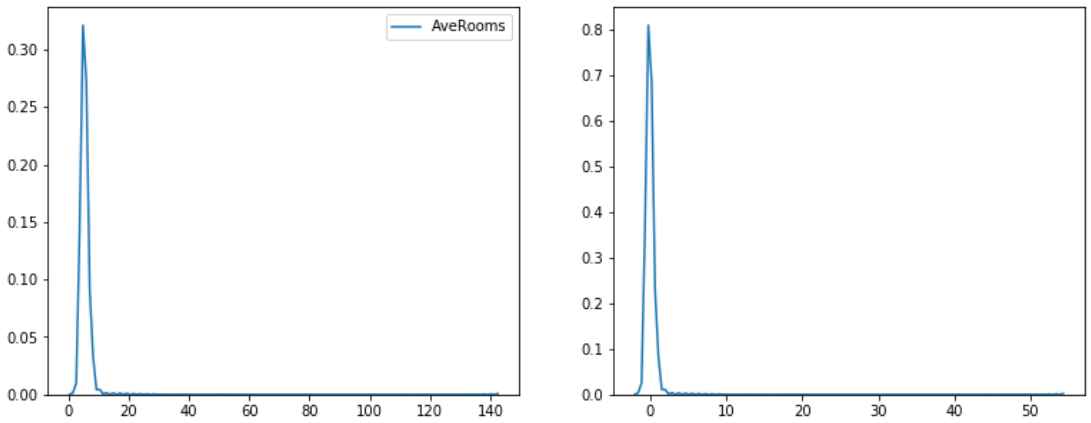
In [19]:
```
def plots(df, var, t):
    plt.figure(figsize=(13,5))
    plt.subplot(121)
    sns.kdeplot(df[var])
    plt.title('before ' + str(t).split('(')[0])

    plt.subplot(122)
    p1 = t.fit_transform(df[[var]]).flatten()
    sns.kdeplot(p1)
    plt.title('after ' + str(t).split('(')[0])
```

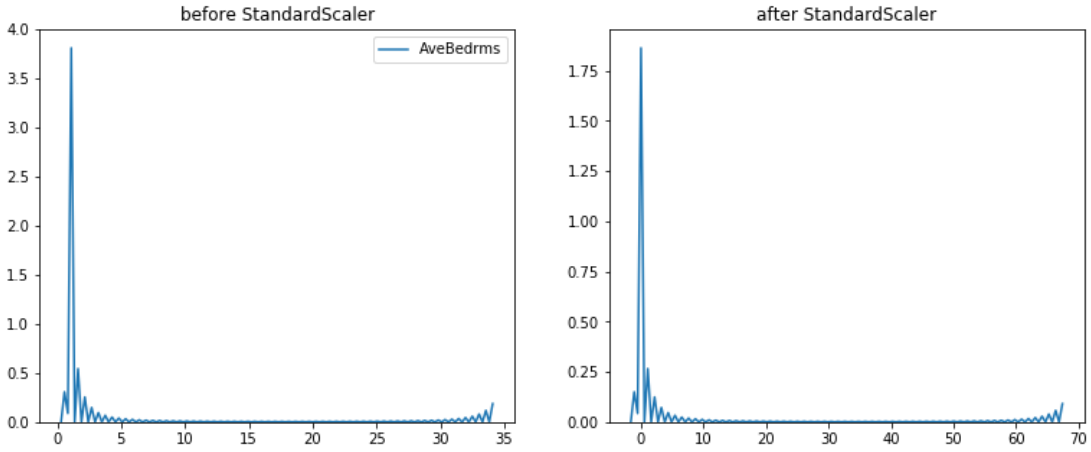
In [20]:
```
for col in X_train.columns:
    plots(X_train, col, StandardScaler())
```

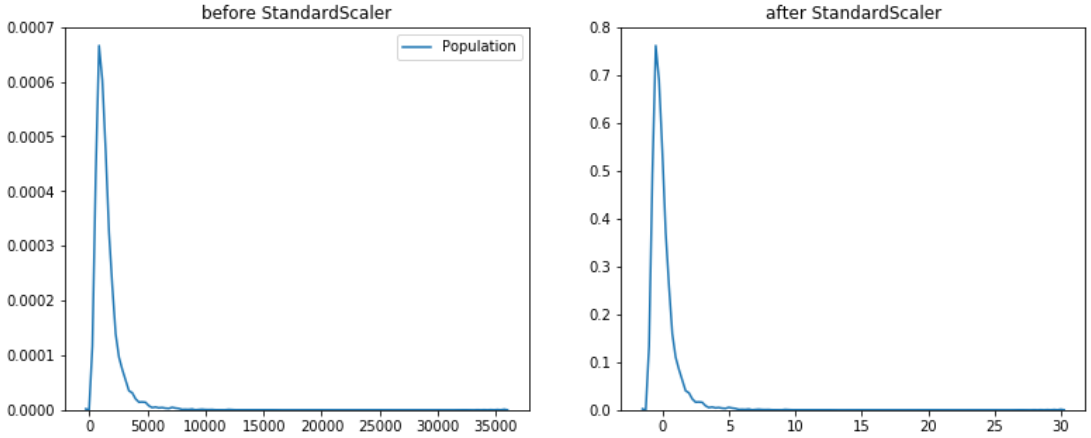


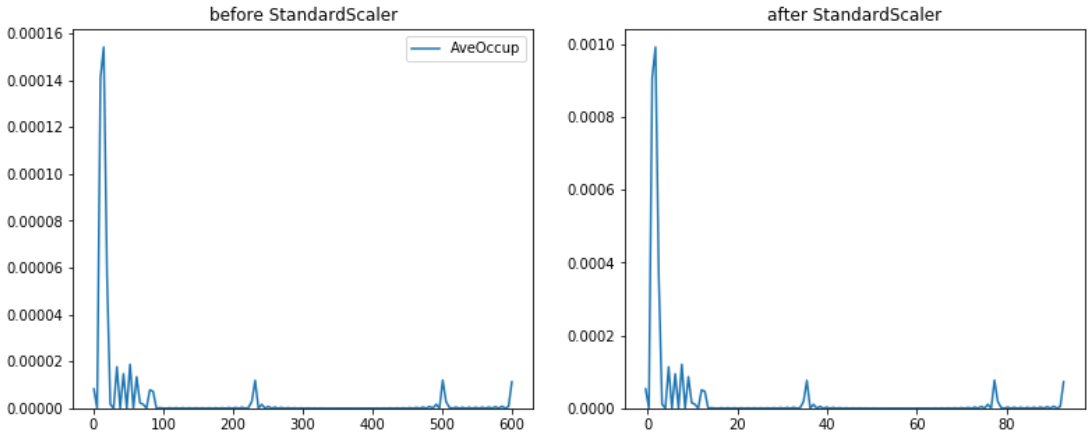


before StandardScaler after StandardScaler









In [ ]:

In [21]:
```
def model accuracy scaled(mod):
```

```
    model_scaled = Pipeline([
        ('scale', StandardScaler()),
        ('model', mod)
    ])
    model_scaled.fit(X_train, y_train)
    return model_scaled.score(X_test, y_test)

def model_accuracy_unscaled(mod):
    model_unscaled = Pipeline([
        ('model', mod)
    ])
    model_unscaled.fit(X_train, y_train)
    return model_unscaled.score(X_test, y_test)
```

In [22]:
```
model_accuracy_scaled(KNeighborsRegressor())
```

Out[22]: 0.5892398258820833

In [23]:
```
model_accuracy_unscaled(KNeighborsRegressor())
```

Out[23]: 0.17191143873653625

In [ ]:
```

```

In [26]:
```
model_accuracy_scaled(RandomForestRegressor(random_state=0))
```

Out[26]: 0.6687193378314035

In [27]:
```
model_accuracy_unscaled(RandomForestRegressor(random_state=0))
```

Out[27]: 0.6687567614986214