

```
In [18]: import numpy as np
import pandas as pd

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

from sklearn.datasets import load_iris, load_boston
from sklearn.metrics import mean_squared_error, median_absolute_error
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score,
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler, MinMaxScaler

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [13]: boston_dataset = load_boston()
df_boston = pd.DataFrame(boston_dataset.data, columns=boston_dataset.feature_names)
df_boston.head()
```

```
/home/anzhela/1cloud/1cloud/lib/python3.8/site-packages/sklearn/uti
ls/deprecation.py:87: FutureWarning: Function load_boston is deprec
ated; `load_boston` is deprecated in 1.0 and will be removed in 1.
2.
```

The Boston housing prices dataset has an ethical problem. You can refer to the documentation of this function for further details.

The scikit-learn maintainers therefore strongly discourage the use of this dataset unless the purpose of the code is to study and educate about ethical issues in data science and machine learning.

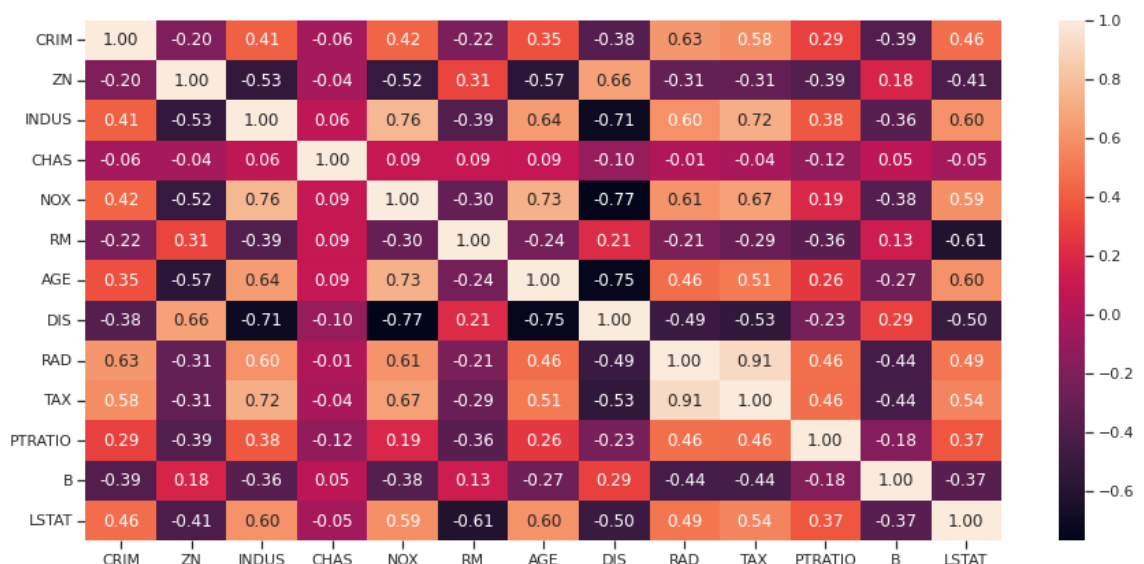
In this special case, you can fetch the dataset from the original source::

Out[13]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
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	
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	
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	
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	
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	

In [14]: `fig, ax = plt.subplots(figsize=(15,7))  
sns.heatmap(df_boston .corr(method='pearson'), ax=ax, annot=True, fm`

Out[14]: <AxesSubplot:>



In [50]: `feature_name = ['AGE', 'NOX', 'INDUS', 'RM']`

In [51]: `df_boston_X_train, df_boston_X_test, df_boston_y_train, df_boston_y_test = train_test_split(df_boston[feature_name].values, boston_dataset.target, test_size=0.2)`

```
In [52]: lnr_rg = Pipeline([('scaler', StandardScaler()), ('regressor', LinearRegression())]
lnr_rg.fit(df_boston_X_train, df_boston_y_train.reshape(-1, 1))
```

```
Out[52]: Pipeline(steps=[('scaler', StandardScaler()),
                          ('regressor', LinearRegression())])
```

```
In [53]: boston_predict = lnr_rg.predict(df_boston_X_test)

mean_squared_error(df_boston_y_test, boston_predict), \
median_absolute_error(df_boston_y_test, boston_predict), \
r2_score(df_boston_y_test, boston_predict)
```

```
Out[53]: (34.25519827548655, 3.5979098669414125, 0.6262572694620032)
```

```
In [54]: rand_fr = RandomForestRegressor()
rand_fr.fit(df_boston_X_train, df_boston_y_train.reshape(-1, 1))

/tmp/ipykernel_4476/1083503232.py:2: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  rand_fr.fit(df_boston_X_train, df_boston_y_train.reshape(-1, 1))
```

```
Out[54]: RandomForestRegressor()
```

```
In [55]: rand_fr_predict = rand_fr.predict(df_boston_X_test)

mean_squared_error(df_boston_y_test, rand_fr_predict), \
median_absolute_error(df_boston_y_test, rand_fr_predict), \
r2_score(df_boston_y_test, rand_fr_predict)
```

```
Out[55]: (15.438122309210518, 1.875, 0.831561740211765)
```

```
In [ ]:
```