In [1]:
```
import numpy as np
import pandas as pd
```

In [3]:
```
from sklearn.datasets import load_boston
boston = load_boston()
```

In [4]:
```
data = pd.DataFrame(boston.data)
```

In [5]:
```
data.head()
```

Out[5]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **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 |
| **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 |
| **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 |
| **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 |
| **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 | 5.33 |

In [7]:
```
data.columns = boston.feature_names
```

In [8]:
```
data['PRICE'] = boston.target
```

In [9]:
```
data.head(n=10)
```

Out[9]:

| | CRIM | ZN | INDUS | CHAS | NOX | RM | AGE | DIS | RAD | TAX | PTRATIO | B | LST |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **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. |
| **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. |
| **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. |
| **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. |
| **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 | 5. |
| **5** | 0.02985 | 0.0 | 2.18 | 0.0 | 0.458 | 6.430 | 58.7 | 6.0622 | 3.0 | 222.0 | 18.7 | 394.12 | 5. |
| **6** | 0.08829 | 12.5 | 7.87 | 0.0 | 0.524 | 6.012 | 66.6 | 5.5605 | 5.0 | 311.0 | 15.2 | 395.60 | 12. |
| **7** | 0.14455 | 12.5 | 7.87 | 0.0 | 0.524 | 6.172 | 96.1 | 5.9505 | 5.0 | 311.0 | 15.2 | 396.90 | 19. |
| **8** | 0.21124 | 12.5 | 7.87 | 0.0 | 0.524 | 5.631 | 100.0 | 6.0821 | 5.0 | 311.0 | 15.2 | 386.63 | 29. |
| **9** | 0.17004 | 12.5 | 7.87 | 0.0 | 0.524 | 6.004 | 85.9 | 6.5921 | 5.0 | 311.0 | 15.2 | 386.71 | 17. |

In [10]:
```
print(data.shape)
```

```
(506, 14)
```

In [11]:
```python
data.isnull().sum()
```

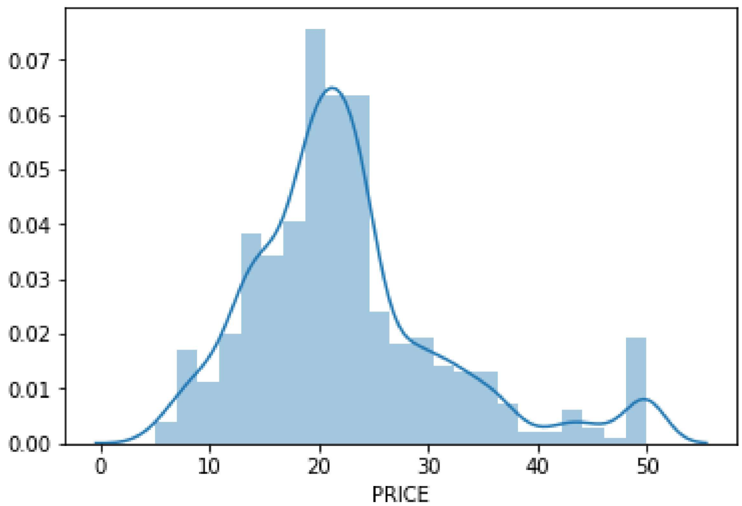
Out[11]:
```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
PRICE      0
dtype: int64
```

In [12]:
```python
data.describe
```

Out[12]:
```
<bound method NDFrame.describe of           CRIM    ZN  INDUS  CHAS    NOX
RM    AGE     DIS   RAD    TAX  \
0     0.00632  18.0   2.31   0.0  0.538  6.575   65.2  4.0900   1.0  296.0
1     0.02731   0.0   7.07   0.0  0.469  6.421   78.9  4.9671   2.0  242.0
2     0.02729   0.0   7.07   0.0  0.469  7.185   61.1  4.9671   2.0  242.0
3     0.03237   0.0   2.18   0.0  0.458  6.998   45.8  6.0622   3.0  222.0
4     0.06905   0.0   2.18   0.0  0.458  7.147   54.2  6.0622   3.0  222.0
5     0.02985   0.0   2.18   0.0  0.458  6.430   58.7  6.0622   3.0  222.0
6     0.08829  12.5   7.87   0.0  0.524  6.012   66.6  5.5605   5.0  311.0
7     0.14455  12.5   7.87   0.0  0.524  6.172   96.1  5.9505   5.0  311.0
8     0.21124  12.5   7.87   0.0  0.524  5.631  100.0  6.0821   5.0  311.0
9     0.17004  12.5   7.87   0.0  0.524  6.004   85.9  6.5921   5.0  311.0
10    0.22489  12.5   7.87   0.0  0.524  6.377   94.3  6.3467   5.0  311.0
11    0.11747  12.5   7.87   0.0  0.524  6.009   82.9  6.2267   5.0  311.0
12    0.09378  12.5   7.87   0.0  0.524  5.889   39.0  5.4509   5.0  311.0
13    0.62976   0.0   8.14   0.0  0.538  5.949   61.8  4.7075   4.0  307.0
14    0.63796   0.0   8.14   0.0  0.538  6.096   84.5  4.4619   4.0  307.0
15    0.62739   0.0   8.14   0.0  0.538  5.834   56.5  4.4986   4.0  307.0
16    1.05393   0.0   8.14   0.0  0.538  5.935   29.3  4.4986   4.0  307.0
```

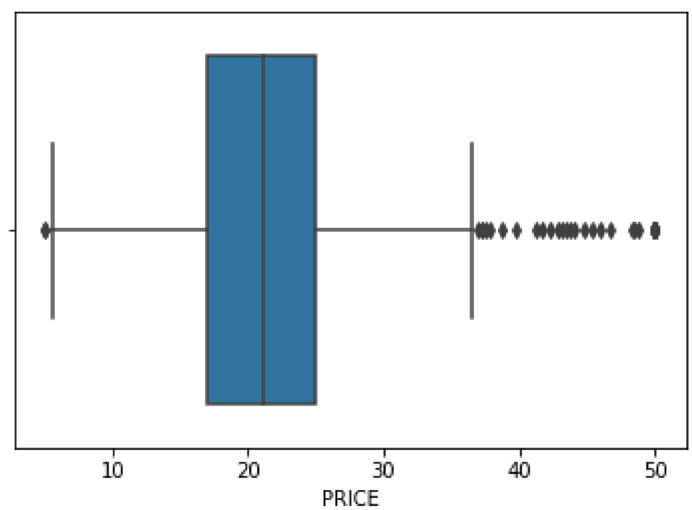
In [16]:
```
import seaborn as sns
sns.distplot(data.PRICE)
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1096facef0>




In [17]:
```
sns.boxplot(data.PRICE)
```

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1094edf4a8>



In [18]:
```
correlation = data.corr()
correlation.loc['PRICE']
```

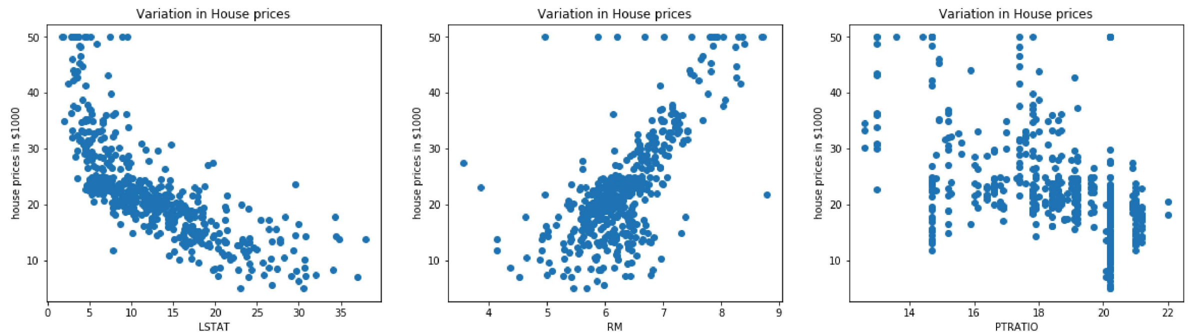
Out[18]:
```
CRIM      -0.388305
ZN         0.360445
INDUS     -0.483725
CHAS       0.175260
NOX       -0.427321
RM         0.695360
AGE       -0.376955
DIS        0.249929
RAD       -0.381626
TAX       -0.468536
PTRATIO   -0.507787
B          0.333461
LSTAT     -0.737663
PRICE      1.000000
Name: PRICE, dtype: float64
```

In [19]:
```
import matplotlib.pyplot as plt
```

In [20]:
```
fig,axes = plt.subplots(figsize=(15,12))
sns.heatmap(correlation, square = True, annot = True)
```

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1094e57438>

In [24]:
```
plt.figure(figsize = (20,5))
features = ['LSTAT' , 'RM', 'PTRATIO']
for i, col in enumerate(features):
    plt.subplot(1, len(features), i+1)
    x = data[col]
    y = data.PRICE
    plt.scatter(x,y, marker='o')
    plt.title("Variation in House prices")
    plt.xlabel(col)
    plt.ylabel("house prices in $1000")
```




In [27]:
```
x = data.iloc[:,:-1]
y = data.PRICE
```

In [35]:
```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.3, random_st
```

In [37]:
```
mean = xtrain.mean(axis=0)
std = xtrain.std(axis=0)
```

In [39]:
```
from sklearn.linear_model import LinearRegression
```

In [40]:
```
regressor = LinearRegression()
```

In [41]:
```
regressor.fit(xtrain, ytrain)
```

Out[41]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [42]:
```
ypred = regressor.predict(xtest)
```

In [43]:
```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
```

In [45]:
```
rmse = (np.sqrt(mean_squared_error(ytest, ypred)))
rmse
```

Out[45]:
```
5.214975145375418
```

In [47]:
```
r2 = r2_score(ytest, ypred)
r2
```

Out[47]: 0.6733825506400176

In [88]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)
```

In [89]:
```
import keras
```

In [90]:
```
from keras.layers import Dense, Activation, Dropout
from keras.models import Sequential
```

In [91]:
```
model = Sequential()
model.add(Dense(128, activation='relu', input_dim=13))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(1))
```

In [93]:
```
model.compile(optimizer = 'adam', loss = 'mean_squared_error', metrics=['mae']
```

In [94]:
```
!pip install ann_visualizer
```

```
Requirement already satisfied: ann_visualizer in /home/rmdstic/anaconda3/lib/
python3.7/site-packages (2.5)
```

In [95]:
```
!pip install graphviz
```

```
Requirement already satisfied: graphviz in /home/rmdstic/anaconda3/lib/python
3.7/site-packages (0.20.1)
```

In [96]:
```
from ann_visualizer.visualize import ann_viz;
```

In [97]:
```
ann_viz(model, title="DEMO ANN");
```

In [98]:
```
history = model.fit(xtrain, ytrain, epochs=100, validation_split=0.05)
```

```
Epoch 1/100
11/11 [==============================] - 2s 17ms/step - loss: 585.4051 - m
ae: 22.3685 - val_loss: 623.2092 - val_mae: 23.6422
Epoch 2/100
11/11 [==============================] - 0s 5ms/step - loss: 535.7755 - ma
e: 21.1923 - val_loss: 541.6809 - val_mae: 21.9797
Epoch 3/100
11/11 [==============================] - 0s 4ms/step - loss: 430.3748 - ma
e: 18.5709 - val_loss: 368.0314 - val_mae: 17.9560
Epoch 4/100
11/11 [==============================] - 0s 4ms/step - loss: 250.3356 - ma
e: 13.3832 - val_loss: 107.7644 - val_mae: 9.4365
Epoch 5/100
11/11 [==============================] - 0s 4ms/step - loss: 95.5615 - ma
e: 7.8445 - val_loss: 31.0145 - val_mae: 4.6807
Epoch 6/100
11/11 [==============================] - 0s 4ms/step - loss: 64.2435 - ma
e: 6.1589 - val_loss: 20.6496 - val_mae: 3.8790
Epoch 7/100
```

In [100]:
```
pip install plotly
```

```
Collecting plotly
  Downloading https://files.pythonhosted.org/packages/a8/07/72953cf70e3bd3a24
cbc3e743e6f8539abe6e3e6d83c3c0c83426eaffd39/plotly-5.18.0-py3-none-any.whl (h
ttps://files.pythonhosted.org/packages/a8/07/72953cf70e3bd3a24cbc3e743e6f8539
abe6e3e6d83c3c0c83426eaffd39/plotly-5.18.0-py3-none-any.whl) (15.6MB)
    100% |████████████████████████████████| 15.6MB 2.6MB/s eta 0:00:01
Requirement already satisfied: packaging in /home/rmdstic/anaconda3/lib/pytho
n3.7/site-packages (from plotly) (19.0)
Collecting tenacity>=6.2.0 (from plotly)
  Downloading https://files.pythonhosted.org/packages/f4/f1/990741d5bb2487d52
9d20a433210ffa136a367751e454214013b441c4575/tenacity-8.2.3-py3-none-any.whl
(https://files.pythonhosted.org/packages/f4/f1/990741d5bb2487d529d20a433210ff
a136a367751e454214013b441c4575/tenacity-8.2.3-py3-none-any.whl)
Requirement already satisfied: six in /home/rmdstic/anaconda3/lib/python3.7/s
ite-packages (from packaging->plotly) (1.12.0)
Requirement already satisfied: pyparsing>=2.0.2 in /home/rmdstic/anaconda3/li
b/python3.7/site-packages (from packaging->plotly) (2.3.1)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.18.0 tenacity-8.2.3
Note: you may need to restart the kernel to use updated packages.
```

In [102]:
```
from plotly.subplots import make_subplots
```

In [103]:
```
import plotly.graph_objects as go
```

In [105]:

```
fig = go.Figure()
fig.add_trace(go.Scattergl(y=history.history['loss'], name='Train'))
fig.add_trace(go.Scattergl(y=history.history['val_loss'], name='Vaild'))
fig.update_layout(height=500, width=700, xaxis_title = 'Epoch', yaxis_title='L
fig.show()
```

In [106]:
```
fig = go.Figure()
fig.add_trace(go.Scattergl(y=history.history['mae'], name='Train'))
fig.add_trace(go.Scattergl(y=history.history['val_mae'], name='Valid'))
fig.update_layout(height=500, width=700, xaxis_title = 'Epoch', yaxis_title='M
fig.show()
```

In [107]:
```
y_pred = model.predict(xtest)
```

In [108]:
```
mse_nn, mae_nn = model.evaluate(xtest, ytest)
print('Mean Squared Error on test data: ', mse_nn)
print('Mean Squared Error on test data: ', mae_nn)
```

```
5/5 [==============================] - 0s 1ms/step - loss: 7002.5273 - mae: 7
1.1850
Mean Squared Error on test data:  7002.52734375
Mean Squared Error on test data:  71.18496704101562
```

In [109]:
```
from sklearn.metrics import mean_absolute_error
```

In [110]:
```
lr_model = LinearRegression()
```

In [111]:
```
lr_model.fit(xtrain, ytrain)
```

Out[111]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [112]:
```
y_pred_lr = lr_model.predict(xtest)
```

In [113]:
```
mse_lr = mean_squared_error(ytest, y_pred_lr)
mae_lr = mean_absolute_error(ytest, y_pred_lr)
```

In [114]:
```
print('Mean Squared Error on Test Data: ', mse_lr)
print('Mean Absolute Error on Test Data: ', mae_lr)
```

```
Mean Squared Error on Test Data:  454.4737836456808
Mean Absolute Error on Test Data:  18.963407281929687
```

In [115]:
```
from sklearn.metrics import r2_score
```

In [121]:
```
r2 = r2_score(ytest, ypred)
print(r2)
```

```
0.6733825506400176
```

In [122]:
```
from sklearn.metrics import mean_squared_error
```

In [124]:
```
rmse = np.sqrt(mean_squared_error(ytest, ypred))
print(rmse)
```

```
5.214975145375418
```

In [128]:
```
import sklearn
new_data = sklearn.preprocessing.StandardScaler().fit_transform(([[0.1,10.0, 5
```

In [129]:
```
prediction = model.predict(new_data)
```

In [131]:
```
print('Prdeicted house price: ', prediction)
```

```
Prdeicted house price:  [[9.389934]]
```

In [ ]:

In [ ]: