

Linear regression by using DNN: Implement Boston housing price prediction problem by Linear regression using DNN.(Boston house price prediction dataset)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
BostonTrain = pd.read_csv("boston_test.csv")
```

```
BostonTrain.head()
```

```
BostonTrain.info()
BostonTrain.describe()
```

```
X = BostonTrain.iloc[:,1:-1].values
Y = BostonTrain.iloc[:, -1].values
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.4)
```

```
X_train.shape
```

```
y_train.shape
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
print(X_train[0].shape)
```

```
model = Sequential()
model.add(Dense(128, activation='relu', input_shape=X_train[0].shape))
model.add(Dense(64, activation = 'relu'))
model.add(Dense(1, activation = 'linear'))
model.compile(optimizer = 'adam', loss = 'mse', metrics = ['mae'])
model.summary()
```

```
model.fit(X_train, y_train, epochs = 100, batch_size = 1, validation_data =
(X_test, y_test))
```

```
X_test[8]
```

```
print("Actual Value: ", y_test[8])
sample = np.array([X_test[8]])
print("Predicted Value: ", model.predict(sample)[0][0])
```

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
y_pred = model.predict(X_test)
y_true = y_test.values if hasattr(y_test, 'values') else y_test
mse = mean_squared_error(y_true, y_pred)
mae = mean_absolute_error(y_true, y_pred)
r2 = r2_score(y_true, y_pred)
print("Mean Squared Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("R2 Score:", r2)
```

