**Create a neural network for the given 'housepricedata.csv' to predict the whether price of the house is above or below median value or not.**

```
#add housepricedata.csv file to your root folder
#add package tensorflow, scikit, pandas
#add keras if not loaded with tensorflow

import tensorflow as tf
import keras
import pandas
import sklearn
import pandas as pd
df = pd.read_csv('housepricedata.csv')
print(df.head())

dataset = df.values
X = dataset[:,0:10]
Y = dataset[:,10]

from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler()
X_scale = min_max_scaler.fit_transform(X)
print(X_scale)

#Spliting the data for training, testing and validation
from sklearn.model_selection import train_test_split
X_train, X_val_and_test, Y_train, Y_val_and_test = train_test_split(X_scale,
Y, test_size=0.3)
X_val, X_test, Y_val, Y_test = train_test_split(X_val_and_test,
Y_val_and_test, test_size=0.5)

#Trains the neural network with Keras
from keras.models import Sequential
from keras.layers import Dense
```

```
model = Sequential([
Dense(32, activation='relu', input_shape=(10,)), Dense(32,
activation='relu'),
Dense(1, activation='sigmoid'),
])
model.compile(optimizer='sgd', loss='binary_crossentropy',
metrics=['accuracy'])
hist = model.fit(X_train, Y_train, batch_size=32, epochs=100,
validation_data=(X_val, Y_val))

#Evaluate the model
print("\nAccuracy:",model.evaluate(X_test, Y_test)[1])

import matplotlib.pyplot as plt
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()
```