Predicting House Price Using Machine Learning

Objective:-

Predicting house prices using machine learning is a common and valuable real time project.It is typically involves a supervised regression task.

Algorithm:-

Step 1:Data Collection

Step 2:Data Preprocessing

Step 3:Feature Selection/Engineering

Step 4:Choose a Machine Learning Algorithm

Step 5:Train the Model

Step 6:Model Evaluation

Step 7:Hyper Parameter Tuning

Step 8:Model Deployment

Step 9:Predict House Prices

Step 10:Regular Updates

Program:-

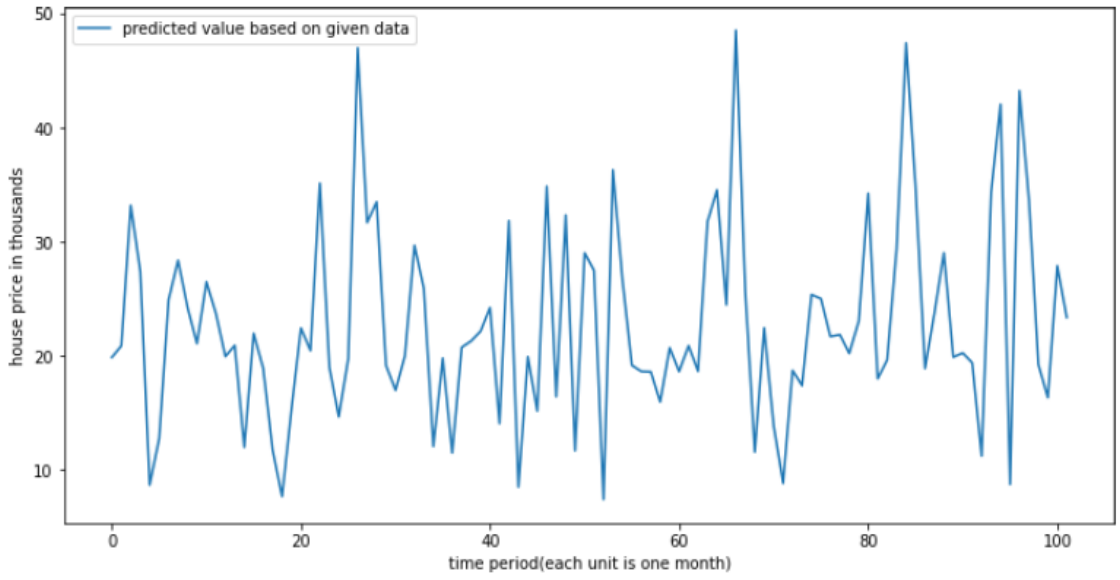
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

HouseDF = pd.read_csv('USA_Housing.csv')
HouseDF.head()
HouseDF=HouseDF.reset_index()
HouseDF.head()
HouseDF.info()
HouseDF.describe()
HouseDF.columns
sns.pairplot(HouseDF)
sns.distplot(HouseDF['Price'])
sns.heatmap(HouseDF.corr(), annot=True)
X = HouseDF[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number
of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population']]
y = HouseDF['Price']
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,
random_state=101)
from sklearn.linear_model import minmaxscaler
lm = minmaxscaler(feature_range=(0,1))
lm.fit_transform(X_train,y_train)
print(lm.intercept_)
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
from keras.layers import Dense,Dropout,LSTM
from keras.models import Sequential
model = Sequential()
model.add(LSTM(units = 50,activation = 'relu',return_sequences = True
,input_shape = (x_train.shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(units = 60,activation = 'relu',return_sequences = True))
model.add(Dropout(0.3))
model.add(LSTM(units = 80,activation = 'relu',return_sequences = True))
```

```
model.add(Dropout(0.4))
model.add(LSTM(units = 120,activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(units = 1))
model.compile(optimizer='adam', loss = 'mean_squared_error')
model.fit(x_train, y_train,epochs=50)
print(lm.intercept_)
coeff_df = pd.DataFrame(lm.coeff_,X.columns,columns=['Coefficient'])
coeff_df
predictions = lm.predict(X_test)
scale_factor = 1/0.02099517
y_predicted = y_predicted*scale_factory
y_test = y_test * scale_factor
plt.scatter(y_test,predictions)
sns.distplot((y_test-predictions),bins=50);
plt.figure(figsize=(12,6))
plt.plot(y_test,'b',label = 'Original Price')
plt.plot(y_predicted,'r',label = 'Predicted Price')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

Output:-

Graph:-


PREDICTED VALUE OF HOUSE PRICE BASED ON TEST SAMPLE DATA
predicted value based on given data
house price in thousands
time period(each unit is one month)
50
40
30
20
10
0
20
40
60
80
100


Actual Price vs Predicted Price
Predicted Prices
Actual Prices
50
40
30
20
10
10
20
30
40
50

Conclusion:-

Thus the machine learning model to predict the house price based on given dataset is executed successfully using xg regressor (a upgraded/ slighted boosted form of regular linear regression, this gives lesser error). This model further helps people understand whether this place is more suited for them based on heatmap correlation. It also helps people looking to sell a house at best time for greater profit. Any house price in any location can be predicted with minimum error by giving appropriate dataset.