IBM NAAN MUDHALVAN

Artificial Intelligence Group – 3

HOUSE PRICE PREDICTION USING MACHINE LEARNING

Team members:

Name **:** Sanjai.J

Reg no**:** 721921243090

A Project Report on

Predicting House Prices using Machine Learning

Phase 2

INNOVATION

Dataset Link: **https://www.kaggle.com/datasets/vedavyasv/usa-housing**

Source Code

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
import xgboost as xg
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
/opt/conda/lib/python3.10/site-packages/scipy/_init_.py:146: UserWarning: A NumPy
version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version
1.23.5
warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Loading Dataset

```
dataset = pd.read_csv('E:/USA_Housing.csv')
```

Model 1 - Linear Regression

```
In [1]:
model_lr=LinearRegression()
In [2]:
model_lr.fit(X_train_scal, Y_train)
Out[2]:
```

Predicting Prices

```
In [3]:
Prediction1 = model_lr.predict(X_test_scal)
```

Evaluation of Predicted Data

```
In [4]:
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)), Y_test, label='Actual Trend')
plt.plot(np.arange(len(Y_test)), Prediction1, label='Predicted Trend')
plt.xlabel('Data')
plt.ylabel('Trend')
plt.legend()
plt.title('Actual vs Predicted')
```

Out[4]:
Text(0.5, 1.0, 'Actual vs Predicted')
In [5]:
sns.histplot((Y_test-Prediction1), bins=50)
Out[5]:
<Axes: xlabel='Price', ylabel='Count'>
In [6]:
print(r2_score(Y_test, Prediction1))
print(mean_absolute_error(Y_test, Prediction1))
print(mean_squared_error(Y_test, Prediction1))
Out[6]:
0.9182928179392918
82295.49779231755
10469084772.975954

Model 2 - Support Vector Regressor

In [7]:
model_svr = SVR()
In [8]:
model_svr.fit(X_train_scal, Y_train)
Out[8]:
Predicting Prices
In [9]:
Prediction2 = model_svr.predict(X_test_scal)
Evaluation of Predicted Data
In [10]:
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)), Y_test, label='Actual Trend')
plt.plot(np.arange(len(Y_test)), Prediction2, label='Predicted Trend')
plt.xlabel('Data')
plt.ylabel('Trend')
plt.legend()
plt.title('Actual vs Predicted')
Out[10]:
Text(0.5, 1.0, 'Actual vs Predicted')
In [11]:
sns.histplot((Y_test-Prediction2), bins=50)
Out[12]:
<Axes: xlabel='Price', ylabel='Count'>
In [12]:
print(r2_score(Y_test, Prediction2))
print(mean_absolute_error(Y_test, Prediction2))
print(mean_squared_error(Y_test, Prediction2))
-0.0006222175925689744

286137.81086908665
128209033251.4034

Model 3 - Lasso Regression

In [13]:
```
model_lar = Lasso(alpha=1)
```
In [14]:
```
model_lar.fit(X_train_scal,Y_train)
```
Out[14]:

Predicting Prices

In [15]:
```
Prediction3 = model_lar.predict(X_test_scal)
```
Evaluation of Predicted Data

In [16]:
```
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)), Y_test, label='Actual Trend')
plt.plot(np.arange(len(Y_test)), Prediction3, label='Predicted Trend')
plt.xlabel('Data')
plt.ylabel('Trend')
plt.legend()
plt.title('Actual vs Predicted')
```
Out[16]:
```
Text(0.5, 1.0, 'Actual vs Predicted')
```
In [17]:
```
sns.histplot((Y_test-Prediction3), bins=50)
```
Out[17]:
```
<Axes: xlabel='Price', ylabel='Count'>
```
In [18]:
```
print(r2_score(Y_test, Prediction2))
print(mean_absolute_error(Y_test, Prediction2))
print(mean_squared_error(Y_test, Prediction2))
```
```
-0.0006222175925689744
286137.81086908665
128209033251.4034
```

Model 4 - Random Forest Regressor

In [19]:
```
model_rf = RandomForestRegressor(n_estimators=50)
```

In [20]:
```
model_rf.fit(X_train_scal, Y_train)
```

ALGORITHM BRIEF OUTLINE

1. Import the python libraries that are required for house price prediction using linear regression. Example: numpy is used for convention of data to 2d or 3d array format which is required for linear regression model ,matplotlib for plotting the graph , pandas for reading the data from source and manipulation that data, etc.

2. First Get the value from source and give it to a data frame and then manipulate this data to required form using head(),indexing, drop().

3. Next we have to train a model, its always best to spilt the data into training data and test data for modelling.

4. Its always good to use shape() to avoid null spaces which will cause error during modelling process.

5. Its good to normalize the value since the values are in very large quantity for house prices , for this we may use minmaxscaler to reduce the gap between prices so that its easy and less time consuming for comparing and values.range usually specified is between 0 to 1 using fit transform.

6. Then we have to make few imports from keras: like sequential for initializing the network,lstm to add lstm layer, dropout to prevent overfitting of lstm layers, dense to add a densely connected network layer for output unit.

7. In lstm layer declaration its best to declare the unit, activiation,return sequence.

8. To compile this model its always best to use adam optimizer and set the loss as required for the specific data.

9. We can fit the model to run for a number of epochs. Epochs are the number of times the learning algorithm will work through the entire training set.

10. Then we convert the values back to normal form by using inverse minimal scale by scale factor.
11. Then we give a test data(present data)to the trained model to get the predicted value(future data).

12. Then we can use matplotlib to plot a graph comparing the test and predicted value to see the increase/decrease rate of values in each time of the year in a particular place. Based on this people will know when its best time to sell or buy a place in a given location.

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.

THANKING YOU