Surya Group of Institutions
VIKRAVANDI -605 652

நான் முதல்வன்
உலகை வெல்லும் இளைய தமிழகம்

Name: Anjugam C
Reg No: 422221106002
Department: ECE
Year/Sem: III / V
NM ID: au422221106002

Prediction House Price Using Machine Learning

Problem Statement: The housing market is an important and complex sector that impacts people's lives in many ways. For many individuals and families, buying a house is one of the biggest investments they will make in their lifetime. Therefore, it is essential to accurately predict the prices of houses so that buyers and sellers can make informed decisions. This project aims to use machine learning techniques to predict house prices based on various features such as location, square footage, number of bedrooms and bathrooms, and other relevant factors.

Project Steps

Phase 1: Problem Definition and Design Thinking

Problem Definition: The problem is to predict house prices using machine learning techniques. The objective is to develop a model that accurately predicts the prices of houses based on a set of features such as location, square footage, number of bedrooms and bathrooms, and other relevant factors. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

Design Thinking:

1. **Data Source:** Choose a dataset containing information about houses, including features like location, square footage, bedrooms, bathrooms, and price.
2. **Data Preprocessing**: Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.
3. **Feature Selection:** Select the most relevant features for predicting house prices.
4. **Model Selection:** Choose a suitable regression algorithm (e.g., Linear Regression, Random Forest Regressor) for predicting house prices.
5. **Model Training:** Train the selected model using the preprocessed data.
6. **Evaluation:** Evaluate the model's performance using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.

Data Source:

So to deal with this kind of issues Today we will be preparing a MACHINE LEARNING Based model, trained on the House Price Prediction Dataset.

You can download the dataset from the link

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

Importing Libraries and Dataset

Here we are using

- **Pandas** – To load the Dataframe
- **Matplotlib** – To visualize the data features i.e. barplot
- **Seaborn** – To see the correlation between features using heatmap

Python3

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

dataset = pd.read_excel("HousePricePrediction.xlsx")

# Printing first 5 records of the dataset
print(dataset.head(5))
```

Output:

```
   MSSubClass MSZoning  LotArea LotConfig BldgType  OverallCond  YearBuilt
0          60       RL     8450    Inside     1Fam            5       2003
1          20       RL     9600       FR2     1Fam            8       1976
2          60       RL    11250    Inside     1Fam            5       2001
3          70       RL     9550    Corner     1Fam            5       1915
4          60       RL    14260       FR2     1Fam            5       2000

   YearRemodAdd Exterior1st  BsmtFinSF2  TotalBsmtSF  SalePrice
0          2003     VinylSd         0.0        856.0   208500.0
1          1976     MetalSd         0.0       1262.0   181500.0
2          2002     VinylSd         0.0        920.0   223500.0
3          1970     Wd Sdng         0.0        756.0   140000.0
4          2000     VinylSd         0.0       1145.0   250000.0
```

Python3

```
dataset.shape
```

Output:

```
(2919,13)
```

Data Preprocessing:

Now, we categorize the features depending on their datatype (int, float, object) and then calculate the number of them.

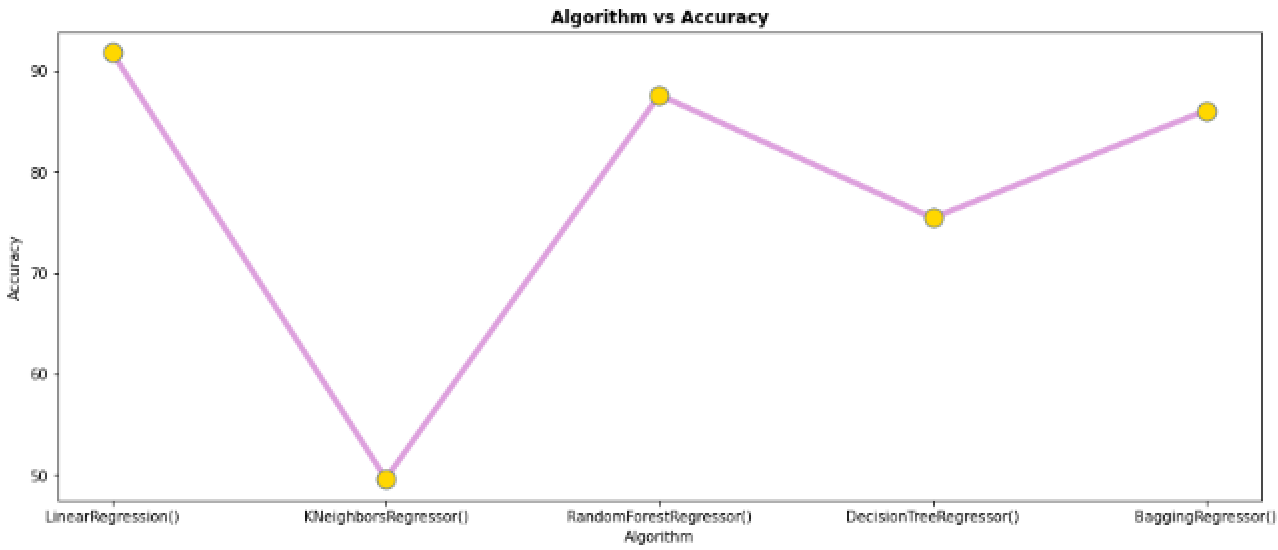
Python3

```
obj = (dataset.dtypes == 'object')
object_cols = list(obj[obj].index)
```

Model Selection:

```
lr = LinearRegression()
dt = DecisionTreeRegressor()
rn = RandomForestRegressor()
knn = KNeighborsRegressor()
sgd = SGDRegressor()
br = BaggingRegressor()
li = [lr,knn,rn,dt,br]
di = {}
for i in li:
    i.fit(X_train,y_train)
    ypred = i.predict(X_test)
    print(i,":",r2_score(ypred,y_test)*100)
    di.update({str(i):i.score(X_test,y_test)*100})
plt.figure(figsize=(15, 6))
plt.title("Algorithm vs Accuracy", fontweight='bold')
plt.xlabel("Algorithm")
plt.ylabel("Accuracy")
plt.plot(di.keys(),di.values(),marker='o',color='plum',linewidth=4,markersize=13,
         markerfacecolor='gold',markeredgecolor='slategray')
plt.show()
```

```
LinearRegression() : 91.00355108791786
KNeighborsRegressor() : 20.484074470563286
RandomForestRegressor() : 83.83229998741602
DecisionTreeRegressor() : 73.30037072629774
BaggingRegressor() : 81.95467285948747
```



In [15]:
```
print(lr.intercept_)
```

```
-2640159.7968526953
```

In [16]:
```
lr.coef_
```

Out[16]:
```
array([2.15282755e+01, 1.64883282e+05, 1.22368678e+05, 2.23380186e+03,
       1.51504200e+01])
```

MODEL BULIDING AND EVALUTION OF PREDICATED DATA

```
 model_lr=LinearRegression()
odel_lr.fit(X_train_scal,
Y_train) Prediction1 =
model_lr.predict(X_test_scal)
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')
 ns.histplot((Y_test-
 Prediction1), bins=50)
```

```
print(r2_score(Y_test, Prediction2))

print(mean_absolute_error(Y_test, Prediction2))
print(mean_squared_error(Y_test, Prediction2))
```

```
print(r2_score(Y_test,
Prediction1))
print(mean_absolute_error(Y_test,
Prediction1))
print(mean_squared_error(Y_test,
Prediction1))
```

```
Model_rf = RandomForestRegressor(n_estimators=50)
```

```
model_rf.fit(X_train_scal, Y_train)
```

CONCLUSION:

Thus the machine learning model to predict the house price based on given dataset is executed successfully using random forester (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.