NAME:S.SATHISHKUMAR

REG NO:422221104037

COLLEGE CODE:4222

TEAM:T9

Predicting house price using machine learning

Introduction

The problem falls under the category of supervised learning algorithms. The dataset we'll be using is boston housing Dataset. The dataset comprises 13 input features and one target feature. The input features include features that may or may not impact the price.

Dataset

The Boston data frame has 506 rows and 14 columns. Each row comprises one data-point and contains details about a plot. Various features affect the pricing of a house.

The Boston housing dataset has 14 features, out of which we'll use 13 to train the model. The 14th feature is the price, which we'll use as our target variable.

STEPS INVOLVED:

1. Importing the required packages into our python environment
2. Importing the house price data and do some EDA on it
3. Data Visualization on the house price data
4. Feature Selection & Data Split
5. Modeling the data using the algorithms
6. Evaluating the built model using the evaluation metrics

Approach taken

We'll use the Random Forest regression algorithm to predict the price of the houses. In this article, we'll consider machine learning algorithms as a black box that fits the data.

This article focuses more on the machine learning pipeline. For more information on the Random Forest algorithm.

We'll begin by loading the data. Since we're using an inbuilt dataset, we'll be calling the load boston function from the sklearn.datasets module. We load the data into the data variable.

Once the data is loaded, we separate the data and target attributes of the data variable. We store them in variables data and target, respectively.

Once we have the data and target values in 2 different variables, we can divide the data into two parts: the testing data and training data.

The theory behind dividing the dataset into two parts is to ensure the model doesn't overfit the training data. Otherwise, the model will perform well on the training data and perform poorly on the test data.

This means that the model has learned the training data so well that it cannot generalize new data points. This should be avoided.

Once we have the dataset split into training and testing sets, we can pre-process the data. Pre-processing involves scaling the values and converting the categorical values into numerical values.

For example, there is a variable in the given dataset that indicates whether the Charles river is close to the house or not. This variable takes the values Near and Far.

We need to convert this into a numerical value. To do this, we can use the labelEncoder function available in the pre-processing module of sklearn. This will replace the column with numerical values of 0 and 1, respectively. 0 indicates Near, and 1 indicates Far.

Once we perform the pre-processing of the dataset, we can fit the data to the model. We begin with instantiating an object of the RandomForestRegressor class. We use the fit method to fit the data to the model.

Once the model is fit, we evaluate the model’s performance on the test set we got earlier. We use the predict method present in the RandomForestRegressor class.

The predict method takes in the test input data and predicts an output. Using the predicted output and the actual output known from the dataset, we compute the test accuracy

Code

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

Data Preprocessing

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

```
obj = (dataset.dtypes == 'object')
object_cols = list(obj[obj].index)
print("Categorical variables:",len(object_cols))

int_ = (dataset.dtypes == 'int')
num_cols = list(int_[int_].index)
print("Integer variables:",len(num_cols))

fl = (dataset.dtypes == 'float')
fl_cols = list(fl[fl].index)
print("Float variables:",len(fl_cols))
```

W

PANDAS

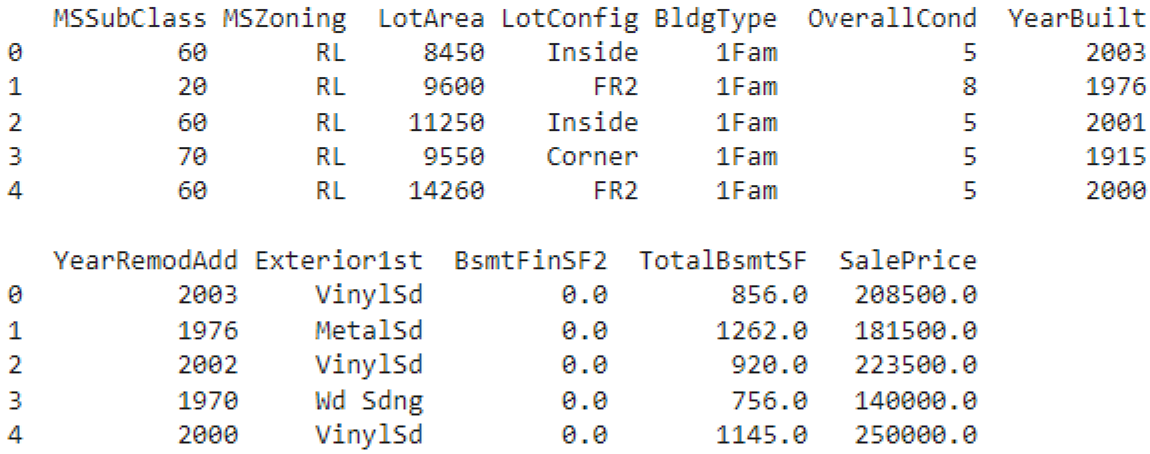
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

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
```