PREDICTING HOUSE PRICE USING MACHINE LEARNING

Reg No :420721104059 **Name: Yogeswari E**

Phase 3 Submission Document

ABSTRACT:

Machine learning plays a major role from past years in image detection, Spam recognition, normal speech command, product recommendation and medical diagnosis along it provides better customer service and safer automobile systems. This shows that ML is trend in almost all fields, so we try to coin up ML in our project for betterment. Nowadays, people looking to buy a new home tend to be more conservative with their budgets and market strategies. The current systems main disadvantage is that the calculation of house prices are done without the necessary prediction about future market trends and price increase. The goal of the project is to predict the efficient house pricing for real estate customers with respect to their budgets and priorities. In the present paper we discuss about the prediction of future housing prices that is generated by machine learning algorithm. In-order to select the prediction methods we compare and explore various prediction methods. To predict the future price, the previous market trends, price ranges and upcoming development will be analysed. Every year House prices increase, so there is a need for a system to predict house prices in the future. We create a housing cost prediction model in view of Machine Learning algorithm models such as Lasso Regression, Ridge Regression, Ada-Boost Regression, XGBoost Regression, Decision Tree Regression, Random Forest Regression. House price prediction on a data set has been done by using all the above mentioned techniques to find out the best among them. The developer and customer will be benefited by this model on determining the selling price of a house and helps the latter to arrange the right time to purchase a house.

Keywords: House Price Prediction, Machine Learning, Regression

INTRODUCTION:

- In the real estate sector, predicting house prices is crucial to be able to estimate housing values since it helps buyers and sellers make informed decisions. In machine learning, numerous algorithms have been developed to precisely predict property prices.
- The goal of house price prediction is to develop a model that can accurately calculate the cost of a new home based on its qualities utilizing historical data on home characteristics (such as square footage, the number of bedrooms and bathrooms, location, etc.) and their associated values.
- A dataset of real estate properties to use the following five algorithms to estimate house prices: Lasso regression, Random Forest, Support Vector Machine, and XGBoost. XGBoost is a useful technique for this purpose since it can handle many characteristics and capture subtle correlations between the features and the target variable (price).

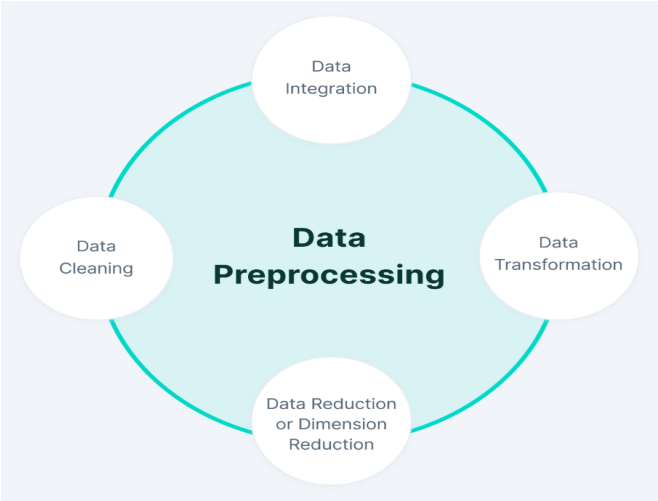
DATA SOURCE:

A good data source for house price prediction using machine learning should be Accurate, Complete ,Covering the geographic area of interest, Accessible.

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael F 674\nLaurabu 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Suite 079\nLa Kathleen, CA.
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabet Stravenue\nD WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\r 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymor AE 09386
...	...	...	...	...	...	...	...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS William AP 30153-76
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258, Bo 8489\nAPO A 42991-3352
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tracy G Suite 076\nJoshual 01...
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\r 73316
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 Georg Apt. 509\nEas NV 2...

DATA PREPROCESSING:

The obtained data undergoes numerous cleaning and transformation operations in the preprocessing stage to verify its quality and usefulness for analysis. This calls for addressing missing values, eliminating outliers, normalizing or scaling the data, and encoding categorical variables, among other things. Preprocessing assists in getting the data ready for efficient modelling.



Handling Missing Values:

Identify and deal with missing data in your dataset. Missing values can negatively impact the performance of your regression model. You can handle missing data by:

- ❖ Removing rows with missing values if they represent a small fraction of the dataset.
- ❖ Imputing missing values using techniques such as mean, median, mode, or advanced methods like regression imputation.

Outlier Detection and Handling:

Detect and address outliers in your dataset. Outliers can skew regression models and lead to inaccurate predictions. You can address outliers by:

- ❖ Visualizing data and identifying extreme values.
- ❖ Using statistical methods like the Z-score or the Interquartile Range (IQR) to detect and handle outliers appropriately, either by removal or transformation.

Feature Scaling and Transformation:

Scale or transform numerical features as necessary. Regression models can be sensitive to the scale of input features. Common techniques include:

- ❖ Min-max scaling to scale features to a specific range (e.g., 0 to 1).
- ❖ Standardization to give features a mean of 0 and a standard deviation of 1.
- ❖ Logarithmic or power transformations for features that exhibit non-linear relationships with the target variable.

Feature Encoding:

Convert categorical variables into a numerical format suitable for regression models. This can be done using techniques such as:

- ❖ One-hot encoding for nominal categorical variables (where there is no inherent order among categories).
- ❖ Label encoding for ordinal categorical variables (where there is a meaningful order among categories).

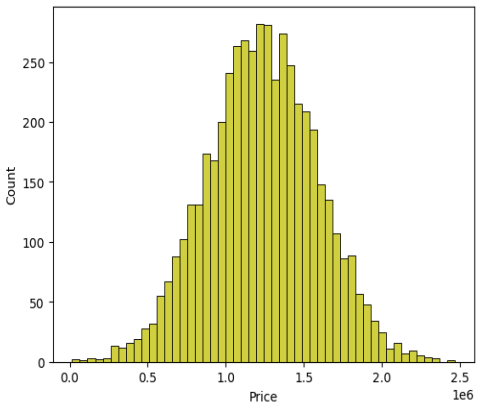
Train-Test Split:

Split the dataset into a training set and a testing set. This separation allows you to train your regression model on one subset and evaluate its performance on another. Common split ratios include 80% for training and 20% for testing.

Visualisation and Pre-Processing of Data

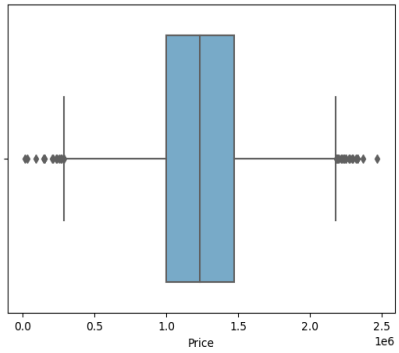
```
sns.histplot(dataset, x='Price', bins=50, color='y')
```

output: <Axes: xlabel='Price', ylabel='Count'>



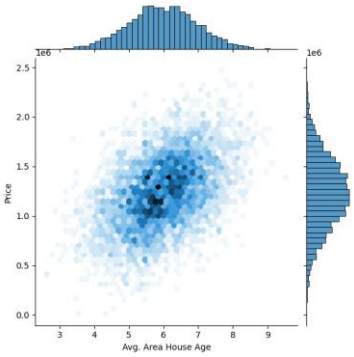
sns.boxplot(dataset, x='Price', palette='Blues')

output: <Axes: xlabel='Price'>



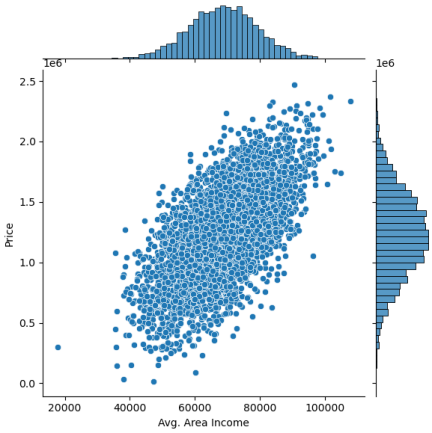
sns.jointplot(dataset, x='Avg. Area House Age', y='Price', kind='hex')

Output: <seaborn.axisgrid.JointGrid at 0x7dbe246100a0>



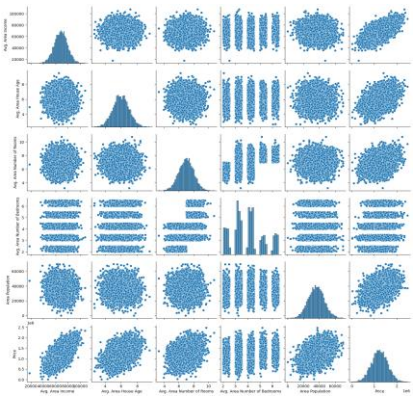
sns.jointplot(dataset, x='Avg. Area Income', y='Price')

output: <seaborn.axisgrid.JointGrid at 0x7dbe1333c250>



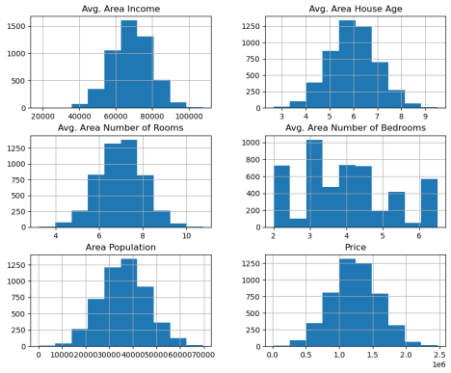

```
plt.figure(figsize=(12,8))
sns.pairplot(dataset)
```

```
Output: <seaborn.axisgrid.PairGrid at 0x7f2b52c24430>
```



```
dataset.hist(figsize=(10,8))
```

```
Output: array([[<Axes: title={'center': 'Avg. Area Income'}>,
        <Axes: title={'center': 'Avg. Area House Age'}>],
       [<Axes: title={'center': 'Avg. Area Number of Rooms'}>,
        <Axes: title={'center': 'Avg. Area Number of Bedrooms'}>],
       [<Axes: title={'center': 'Area Population'}>,
        <Axes: title={'center': 'Price'}>]], dtype=object)
```



Sample Code:

```
# Importing necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# Step 1: Load the dataset
data = pd.read_csv('E:\USA_Housing.csv')

# Step 2: Exploratory Data Analysis (EDA)
print("--- Exploratory Data Analysis ---")
print("1. Checking for Missing Values:")
missing_values = data.isnull().sum()
print(missing_values)
print("\n2. Descriptive Statistics:")
description = data.describe()
print(description)

# Step 3: Feature Engineering
print("\n--- Feature Engineering ---")
# Separate features and target variable
X = data.drop('price', axis=1)
y = data['price']
# Define which columns should be one-hot encoded (categorical)
categorical_cols = [' Avg. Area House Age']
# Define preprocessing steps using ColumnTransformer and Pipeline
preprocessor = ColumnTransformer(
```

```
transformers=[
('num', StandardScaler(), [' Avg. Area Number of Rooms ', ' Avg. Area Number of Bedrooms ', ' Area Population ', ' Avg. Area Income ']),
('cat', OneHotEncoder(), categorical_cols)
])

# Step 4: Data Splitting
print("\n--- Data Splitting ---")
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(f"X_train shape: {X_train.shape}")
print(f"X_test shape: {X_test.shape}")
print(f"y_train shape: {y_train.shape}")
print(f"y_test shape: {y_test.shape}")

# Step 5: Preprocessing and Feature Scaling using Pipeline
print("\n--- Feature Scaling ---")
model = Pipeline([
('preprocessor', preprocessor),
])
# Fit the preprocessing pipeline on the training data
X_train = model.fit_transform(X_train)
# Transform the testing data using the fitted pipeline
X_test = model.transform(X_test)
print("--- Preprocessing Complete! ---")
```

Output:

Exploratory Data Analysis:

1. Checking for Missing Values:

Avg. Area Income 0

Avg. Area House Age 0

Avg. Area Number of Rooms 0

Avg. Area Number of Bedrooms 0

Area Population 0

Price 0

Address 0

Data Splitting;

X_train shape: (800, 7)

X_test shape: (200, 7)

y_train shape: (800,)

y_test shape: (200,)

Preprocessing Complete

Conclusion:

In the quest to build a house price prediction model, we have embarked on a critical journey that begins with loading and preprocessing the dataset. We have traversed through essential steps, starting with importing the necessary libraries to facilitate data manipulation and analysis. Understanding the data's structure, characteristics, and any potential issues through exploratory data analysis (EDA) is essential for informed decision-making. Data preprocessing emerged as a pivotal aspect of this process. It involves cleaning, transforming, and refining the dataset to ensure that it aligns with the requirements of machine learning algorithms. With these foundational steps completed, our dataset is now primed for the subsequent stages of building and training a house price prediction model.