## HOUSE PRICE PREDICTION USING MACHINE

## LEARNING

Project: House Price Prediction

’

###### OBJECTIVE:

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.

Phase 1: Problem Definition and Design Thinking

##### Data Source:

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

**Dataset Link:** **[https://www.kaggle.com/datasets/vedavyasv/usa-housing](https://www.kaggle.com/datasets/vedavyasv/usa-housing" \t "https://courses.myclass.skillup.online/courses/course-v1:IBM+AI101+2023_B5/courseware/d8660830b7ec4f2e8158584fd8319a7d/6edc648a54684cf2a2882b39112be886/[object Object])**

#### project img2

##### Data Preprocessing:

Data Preprocessing is the first critical first step in any machine learning project. It involves Cleaning the data and preprocess the data, handle missing values, and .convert categorical features into numerical representations.

###### Duplicate Removal:

Duplicate rows can introduce bias into the model. We will identify and remove duplicate typically by sorting the dataset based on unique identifier(e.g. property, ID) and then eliminating consecutive rows with same identifier

1. Handling Missing Values:

Missing data are common and needs to be addressed. We will utilize suitable methods such as:

* **Mean Imputation:** Replace the missing values with the mean of the features for the remaining rows. This is appropriate foe numerical features.
* **Median Imputation:** If data contain outliers , median imputation can be more robust as it is less sensitive to extreme values.

#### Feature Selection:

Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.

#### **Model Selection:**

Linear Regression:

#importing libraries

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression, SGDRegressor

from sklearn.linear\_model import ElasticNet

#1.Model Preparation

X = df.drop(['Price'],axis=1)

y = df['Price']

print(X.shape)

dataf numeric cols]= imputer numeric.fit transform(data[numeric cols])

dataf categorical cols]= imputer categorical. fit transform(data[categorical cols])

#2. Convert Categorical Features to Numerical

#We'll use Label Encoding for simplicity here. You can also use one-hot encoding for nominal categorical features.

label encoder Label Encoder()

for col in categorical cols:

data[col] = label\_encoder.fit\_transform(data[col])

#3. Split Data into Features (X) and Target (y)

X= data.drop(columns=['Price']) # Features

y= data['Price'] # Target

#4. Normalize the Data

scaler StandardScaler()

X scaled = scaler.fit transform(X)

6:00 / 16:07

#Split data into training and testing sets (adjust test\_size as needed)

X train, X test, y train, y test train test split(X scaled, y, test size=0.2, random state 42)

# Display the preprocessed data

print("\nPreprocessed Data:")

print(X\_train[:5]) # Display first 5 rows of preprocessed features

print(y train[:5]) # Display first 5 rows of target values

Forest Regression:

# Model

rfr = RandomForestRegressor(max\_depth=10, random\_state=42)

rfr.fit(X\_train, y\_train)

# Prediction

pred5 = rfr.predict(X\_test)

# Performance Measure

rfr\_mse = mean\_squared\_error(y\_test, pred5)

rfr\_rmse = np.sqrt(lr\_mse)

rfr\_r2 = r2\_score(y\_test, pred5)

# Show Measures

result = '''

MSE : {}

RMSE : {}

R^2 : {}

'''.format(lr\_mse, lr\_rmse, lr\_r2)

print(result)

# Model Comparision

names.append("RFR")

mses.append(rfr\_mse)

rmses.append(rfr\_rmse)

r2s.append(rfr\_r2)

**OUTPUT:**

Dataset Preview:

Avg. Area Income Avg. Area House Age Avg. Area Number of Rooms \

0 79545.458574 5.682861 7.009188

1 79248.642455 6.002900 6.730821

2 61287.067179 5.865890 8.512727

3 63345.240046 7.188236 5.586729

4 59982.197226 5.040555 7.839388

Avg. Area Number of Bedrooms Area Population Price \

0 4.09 23086.800503 1.059034e+06

1 3.09 40173.072174 1.505891e+06

2 5.13 36882.159400 1.058988e+06

3 3.26 34310.242831 1.260617e+06

4 4.23 26354.109472 6.309435e+05

##### Conclusion:

A modular framework for house price prediction using machine learning is a flexible and extensible solution that can be used to develop comprehensive prediction systems for a variety of stakeholders.