PREDICTING HOUSE PRICES USING MACHINE LEARNING

PHASE 4 : DEVELOPMENT PART 2

AI_PHASE_4

NM:PROJECT

NAME	VISHVAJITH V
REGESTER NUMBER	61772221T310
DATE	01 - 11 - 2023

COLLEGE : GCE-SALEM

01-Nov-23

SYNAPSIS :

INTRODUCTION :

- Machine learning involves training a computer to recognize patterns and make predictions based on data.
- In the case of house price prediction, we can use historical data on various features of a house, such as its location, size, and amenities, to train a machine learning model.
- Once the model is trained, it can analyze new data on a given house and make a prediction of its market value.

DATASET LINK :

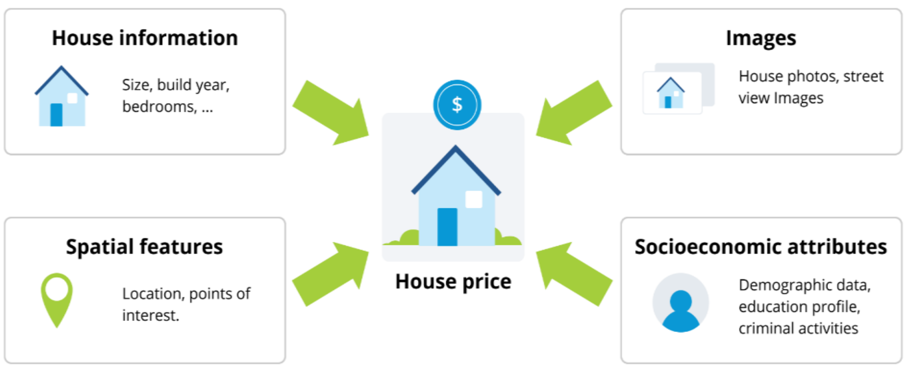
https://www.kaggle.com/datasets/houseprice/csv-housing

PROBLEM STATEMENT :

- 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.

OVER VIEW :

- **Data Collection**
  - Gather historical house price data
  - Collect features like square footage, bedrooms, location, etc.
- **Data Preprocessing**
  - Handle missing data
  - Fill missing values
- **Encode categorical variables**
  - Convert categorical data to numerical
  - Split data into training and testing sets
- **Choose a Model**
  - Select Linear Regression
- **Train the Model**
  - Use training data to train the Linear Regression model
- **Model Evaluation**
  - Use testing data to evaluate model performance
  - Calculate MAE, MSE, RMSE, etc…
  - Visualize results (scatter plots, etc…
- **Make Predictions**
  - Use the trained model to predict house prices


House information
Size, build year, bedrooms, ...
Images
House photos, street view Images
$
House price
Spatial features
Location, points of interest.
Socioeconomic attributes
Demographic data, education profile, criminal activities

ALGORITHM :

- **STEP 1** : Import the required libraries and modules, including pandas for data manipulation, scikit-learn for machine learning algorithms, and Linear Regression for the linear regression model.

- **STEP 2 :** Loading the required dataset with pd.read_csv and select the features we want to use for prediction (e.g., bedrooms, bathrooms, sqft_living, sqft_lot, floors, and zip code), as well as the target variable (price).

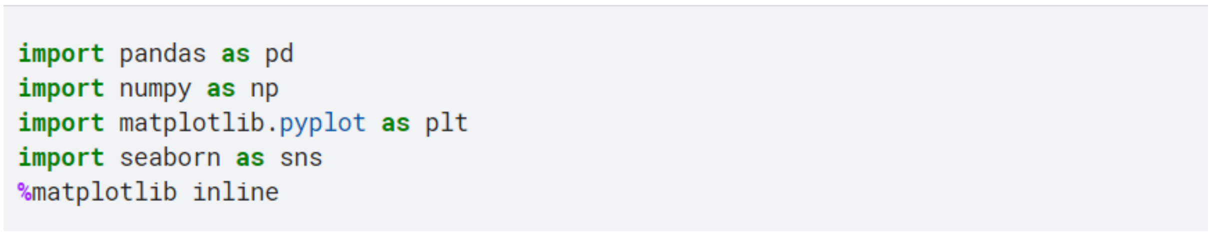
- **STEP 3 :** Split the data into a training set and a test set using the train_test_split function, with 80% of the data used for training and 20% for testing.

- **STEP 4:** Create an instance of the linear regression model using Linear Regression(). We then perform the model training by calling the function fit() with the training data.

- **STEP 5:** Demonstrate how to predict the price of a new house by creating a new data frame new house with the features of the house. We pass this data frame to the model's prediction function to obtain the predicted price.

IMPORTING NECESSARY LIBRARIES :

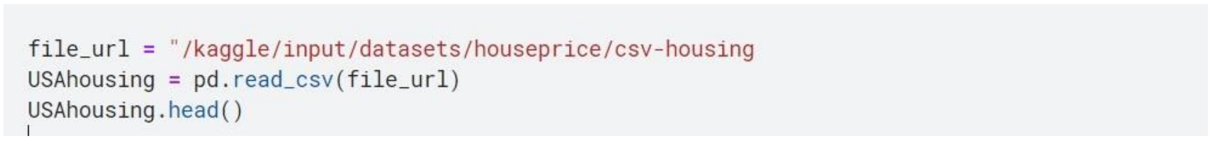
- pandas is used for data manipulation and loading.
- numpy is used for numerical operations.
- matplotlib and seaborn are used for data visualization.



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

LOAD THE DATASET:

- We use the pandas library to import a dataset named 'Housing.csv' by employing the pd.read_csv() function.
- The loaded data is stored in a data frame called df, which is a two dimensional table for data storage and manipulation.



```
file_url = "/kaggle/input/datasets/houseprice/csv-housing
USAhousing = pd.read_csv(file_url)
USAhousing.head()
```

ANALYZING THE PROGRAM:

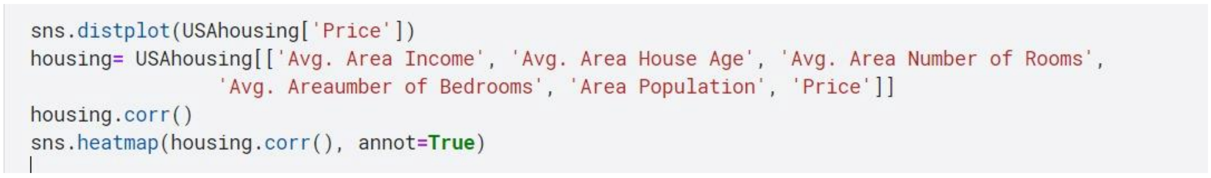
The data contains the following columns:

- 'Avg. Area Income': Avg. Income of residents of the city house is located in.
- 'Avg. Area House Age': Avg Age of Houses in same city
- 'Avg. Area Number of Rooms': Avg Number of Rooms for Houses in same city
- 'Avg. Area Number of Bedrooms': Avg Number of Bedrooms for Houses in same city

- 'Area Population': Population of city house is located in
- 'Price': Price that the house sold at
- 'Address': Address for the house

EXPLANATION :

1.Exploratory Data Analysis (EDA) :



```
sns.distplot(USAhousing['Price'])
housing= USAhousing[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
                'Avg. Areaumber of Bedrooms', 'Area Population', 'Price']]
housing.corr()
sns.heatmap(housing.corr(), annot=True)
```

➢ **Summary Statistics:**

- Calculate and review summary statistics for numerical features, such as mean, median, standard deviation, minimum, and maximum values.
- This helps in understanding the central tendencies and distributions of the variables.

➢ **Correlation Analysis:**

- Calculate and visualize the correlation matrix to understand the relationships between numerical features and the target variable (house prices).
- Identify features that are highly correlated with the target, as they are likely to be significant predictors.

➢ **Feature Relationships:**

- Explore relationships between different features to uncover patterns and potential interactions that can affect house prices.
- For example, you might examine how the number of bedrooms and bathrooms relates to house prices or how location affects price variations.

➢ **Feature Engineering:**

- Based on your EDA findings, you may decide to create new features or transformations to enhance the predictive power of the model.
- For instance, generating features like "age of the house" or "distance to important amenities" can be valuable.

➢ **Documentation:**

- Keep a detailed record of your EDA process, including the insights gained, decisions made, and any actions taken to preprocess the data.
- This documentation will help you and your team understand the dataset and its characteristics.

2.Creating and Training the Model:

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train,y_train)
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
```

- **Data Split:** Divide the dataset into training and testing sets to evaluate the model's performance.
- **Model Selection:** Choose a suitable machine learning algorithm for regression, like linear regression or decision trees.
- **Feature Selection:** Decide which features to include in the model based on their relevance and importance.
- **Model Training:** Train the model using the training data to learn the relationships between features and house prices.
- **Hyperparameter Tuning:** Fine-tune the model's hyperparameters for optimal performance using techniques like cross-validation.
- **Model Evaluation:** Assess the model's performance on the testing set using evaluation metrics such as MAE, MSE, or RMSE.
- **Model Deployment:** Integrate the trained model into an application or platform for practical use.
- **Continuous Monitoring:** Periodically update the model with new data to ensure its accuracy and relevance in predicting house prices.

CODE :

#START OF PROGRAM:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
file_url = "/kaggle/input/datasets/houseprice/csv-housing
USAhousing = pd.read_csv(file_url)
USAhousing.head()
USAhousing.info()
USAhousing.describe()
USAhousing.columns
sns.distplot(USAhousing['Price'])
housing= USAhousing[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
                'Avg. Areaumber of Bedrooms', 'Area Population', 'Price']]
housing.corr()
sns.heatmap(housing.corr(), annot=True)
X = housing[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
                'Avg. Area Number of Bedrooms', 'Area Population']]
y = housing['Price']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)  # 40% for test
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
```

```
lm.fit(X_train,y_train)
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
predictions = lm.predict(X_test)
plt.scatter(y_test, predictions, edgecolor='black
            from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

#End of Program

OUTPUT :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House Age           5000 non-null   float64
 2   Avg. Area Number of Rooms     5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null   float64
 5   Price                         5000 non-null   float64
 6   Address                       5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

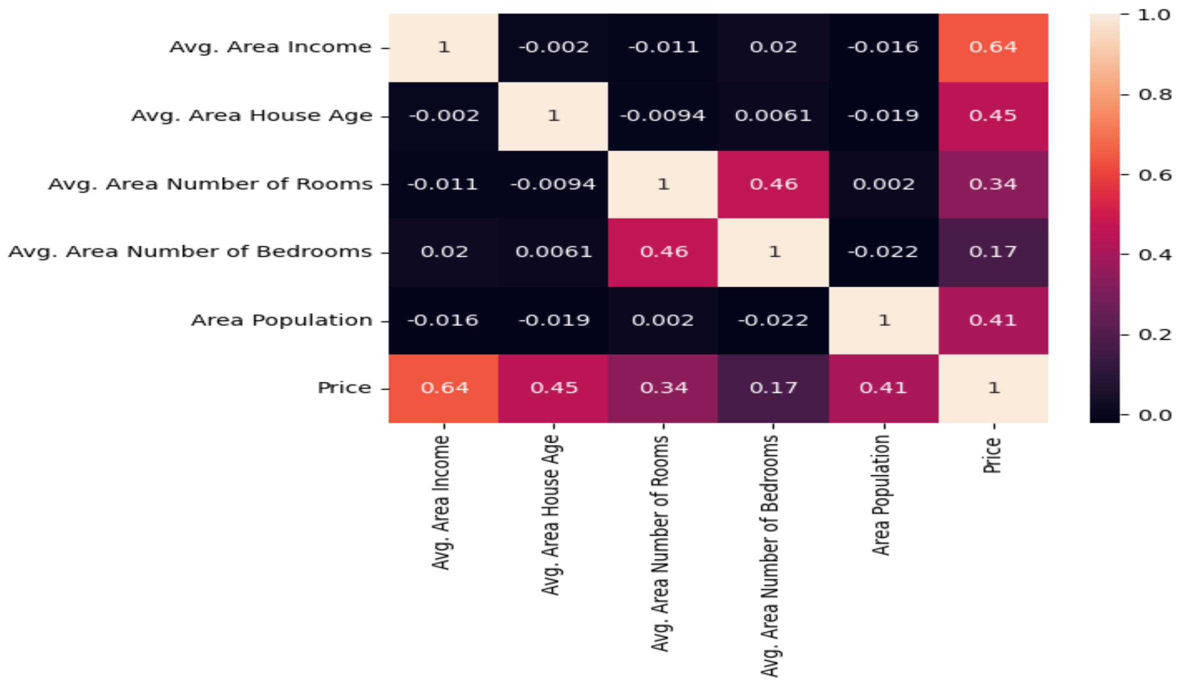
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

	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 Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386

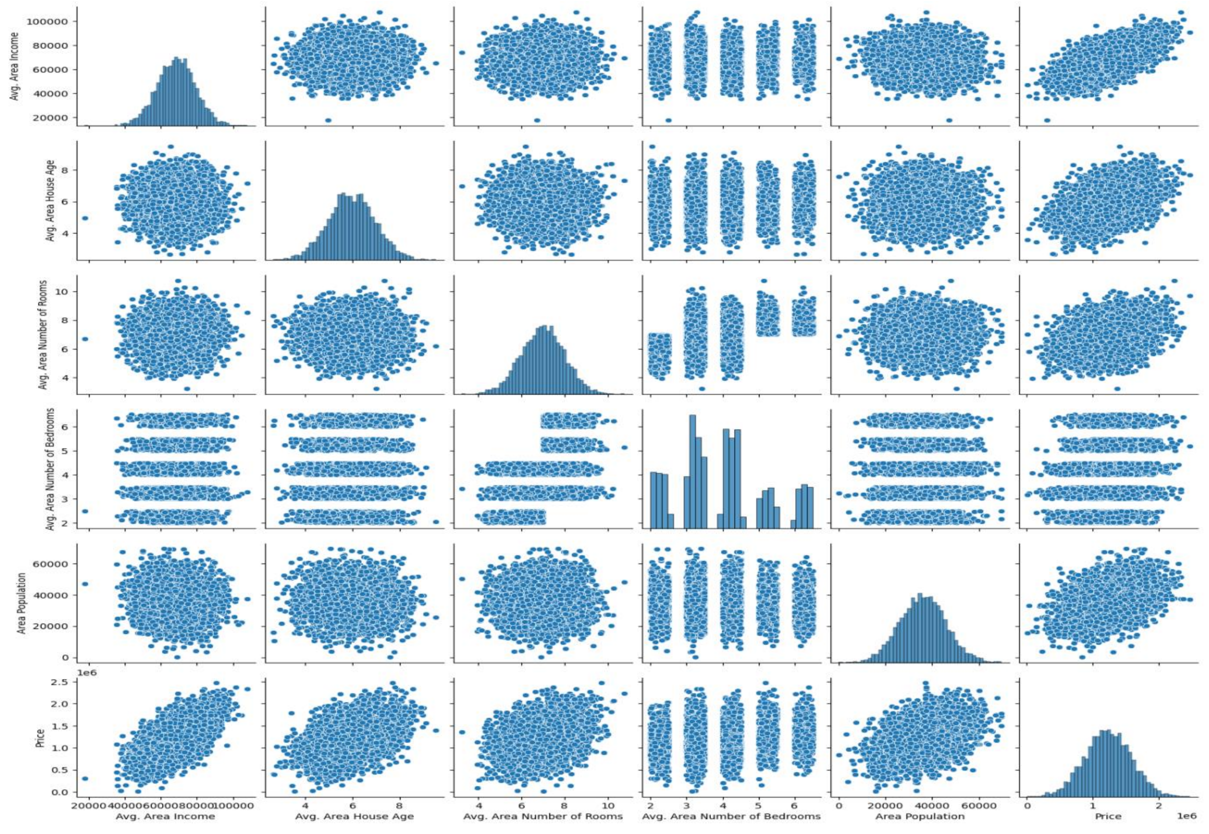
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
Avg. Area Income	1.000000	-0.002007	-0.011032	0.019788	-0.016234	0.639734
Avg. Area House Age	-0.002007	1.000000	-0.009428	0.006149	-0.018743	0.452543
Avg. Area Number of Rooms	-0.011032	-0.009428	1.000000	0.462695	0.002040	0.335664
Avg. Area Number of Bedrooms	0.019788	0.006149	0.462695	1.000000	-0.022168	0.171071
Area Population	-0.016234	-0.018743	0.002040	-0.022168	1.000000	0.408556
Price	0.639734	0.452543	0.335664	0.171071	0.408556	1.000000

	Coefficient
Avg. Area Income	21.528276
Avg. Area House Age	164883.282027
Avg. Area Number of Rooms	122368.678027
Avg. Area Number of Bedrooms	2233.801864
Area Population	15.150420

REFERENCE :



Avg. Area Income
Avg. Area House Age
Avg. Area Number of Rooms
Avg. Area Number of Bedrooms
Area Population
Price
1 -0.002 -0.011 0.02 -0.016 0.64
-0.002 1 -0.0094 0.0061 -0.019 0.45
-0.011 -0.0094 1 0.46 0.002 0.34
0.02 0.0061 0.46 1 -0.022 0.17
-0.016 -0.019 0.002 -0.022 1 0.41
0.64 0.45 0.34 0.17 0.41 1
1.0
0.8
0.6
0.4
0.2
0.0



Avg. Area Income
Avg. Area House Age
Avg. Area Number of Rooms
Avg. Area Number of Bedrooms
Area Population
Price
1e6

Data Cleaning :

Data Cleaning is the way to improvise the data or remove incorrect, corrupted or irrelevant data.

As in our dataset, there are some columns that are not important and irrelevant for the model training. So, we can drop that column before training.

There are 2 approaches to dealing with empty/null values

1. We can easily delete the column/row (if the feature or record is not much important).

2. Filling the empty slots with mean/mode/0/NA/etc. (depending on the dataset requirement).

As Id Column will not be participating in any prediction. So we can Drop it

Model and Accuracy :

- As we have to train the model to determine the continuous values, so we will be using these regression models.
- SVM-Support Vector Machine
- Random Forest Regressor
- Linear Regressor
- And To calculate loss we will be using the mean_absolute_percentage_error module.
- It can easily be imported by using sklearn library.
- SVM – Support vector Machine
- SVM can be used for both regression and classification model. It finds the hyperplane in the n-dimensional plane. To read more about svm refer this

CONCLUSION :

- In conclusion, using machine learning in Python is a powerful tool for predicting house prices.
- By gathering and cleaning data, visualizing patterns, and training and evaluating our models, we can make informed decisions in the dynamic world of real estate.
- By leveraging advanced algorithms and data analysis, we can make accurate predictions and inform decision-making processes.
- This approach empowers buyers, sellers, and investors to make informed choices in a dynamic and competitive market, ultimately maximizing their opportunities and outcomes.