Predicting House Prices Using Machine Learning

Predicting house prices using machine learning is a common and Valuable real-time project. Here’s a high-level outline of how you can Approach such a project.

➢ Data Connection

Gather a dataset that includes features like square Footage, number of bedrooms, bathrooms, location, and historical Sale prices.

➢ Data Preprocessing

- Handle missing data and outliers.
- Encode categorical variables (e.g., location) into numerical Format.
- Split the dataset into a training set and a test set for model Evaluation.

➢ Feature Selection/Engineering

- Select relevant features that have a significant impact on house Prices.
- Create new features if needed, like price per square foot or age Of the property.

➢ Model Selection

- Choose a regression model suitable for the task. Linear

Regression, decision trees, random forests, or gradient boosting are Common choices.

➢ Model Training

- Train the selected model using the training dataset.

➢ Model Evaluation

- Evaluate the model’s performance using appropriate metrics Like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).

➢ Hyperparameter Tuning

- Optimize model hyperparameters to improve performance.

➢ Deployment

- Deploy the trained model in a real-time environment, which can Be a web application or API.

➢ Testing and Monitoring

- Continuously test the model’s performance in a real-time setting And monitor for drift or degradation.

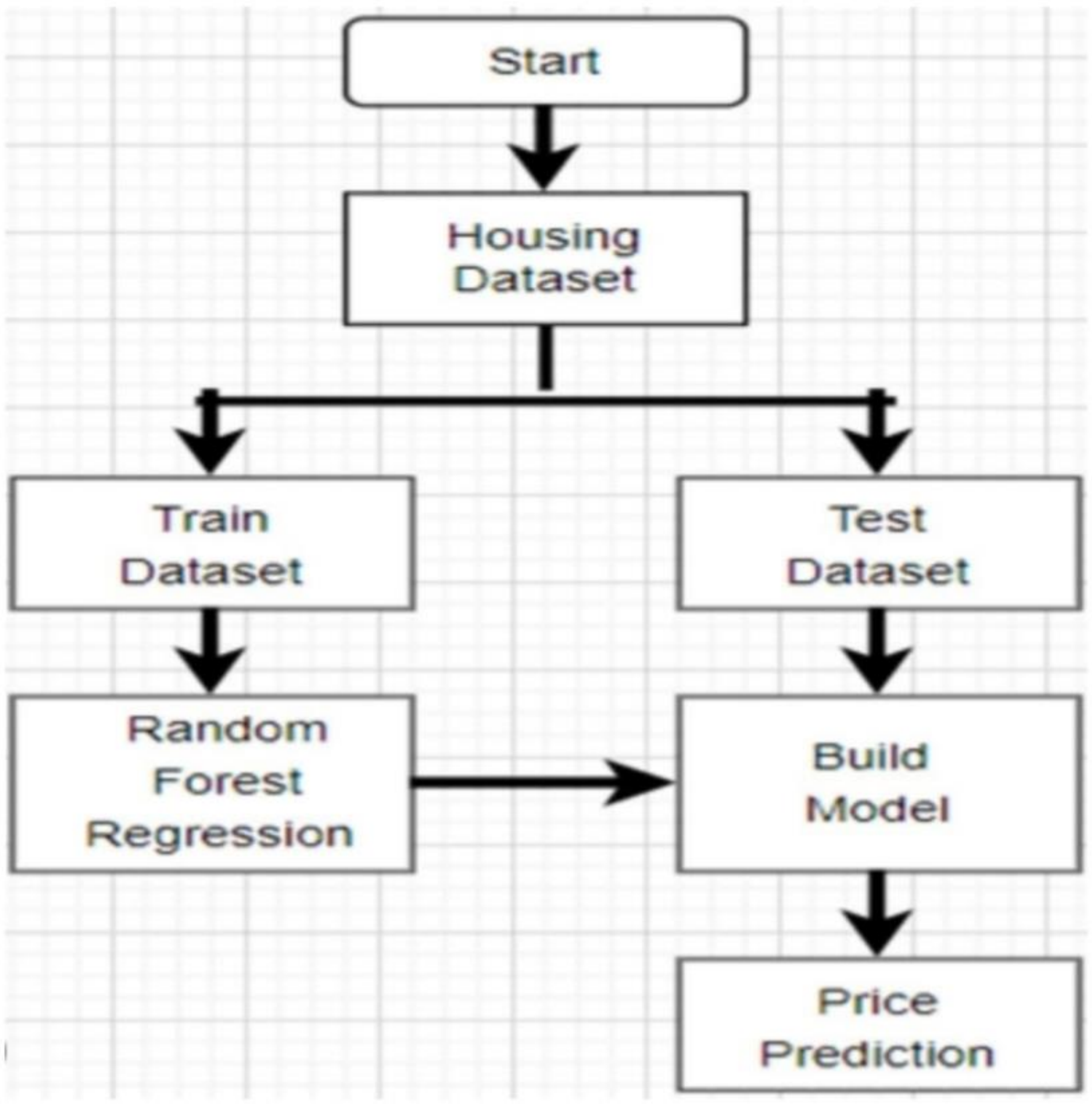
➢ User Interface

- Develop a user-friendly interface for users to input property Features and receive price predictions.

➢ Feedback Loop

- Collect user feedback and data to further improve

FLOW CHART :



Objective:-

Predicting house prices using machine learning

Typically involves a supervised regression task.

Step 1: Data Collection

- Gather a dataset that includes historical information About houses, such as features (square footage, number of Bedrooms, location, etc.) and their corresponding sale Prices.

Step 2: Data Preprocessing

- Handle missing data, outliers, and duplicates.

- Encode categorical features (e.g., one-hot encoding Or label encoding).

- Split the data into a training set and a Testing/validation set.

Step 3: Feature Selection/Engineering

- Select relevant features that influence house Prices.

- Create new features or transform existing ones if Necessary (e.g., feature scaling, normalization).

Step 4: Choose a Machine Learning Algorithm

- Select a regression algorithm. Common choices

Include Linear Regression, Decision Trees, Random Forest,Gradient Boosting, or even deep learning methods like Neural Networks.

Step 5: Train the Model

- Use the training data to train the selected Algorithm. The model learns the relationship between house Features and their prices.

Step 6: Model Evaluation

- Use the testing/validation set to evaluate the Model’s performance. Common regression metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), And Root Mean Squared Error (RMSE).

Step 7: Hyperparameter Tuning

- Fine-tune the model's hyperparameters to improve Its performance. Techniques like cross-validation can be helpful .

Step 8: Model Deployment

- Once you're satisfied with the model's

Performance, deploy it for predictions.

Step 9: Predict House Prices

- Input the features of a house you want to predict The price for into the trained model.

- The model will provide an estimated house price as The output.

Step 10: Regular Updates

- Periodically update the model with new data to Keep it accurate as the housing market changes

Program:-

```
Import pandas as pd

Import numpy as np

Import seaborn as sns

Import matplotlib.pyplot as plt
```

```
%matplotlib inline

HouseDF = pd.read_csv(‘USA_Housing.csv’)

HouseDF.head()

HouseDF=HouseDF.reset_index()

HouseDF.head()

HouseDF.info()

HouseDF.describe()
HouseDF.columns

Sns.pairplot(HouseDF)

Sns.distplot(HouseDF[‘Price’])

Sns.heatmap(HouseDF.corr(), annot=True)

X = HouseDF[[‘Avg. Area Income’, ‘Avg. Area
House Age’, ‘Avg. Area Number

Of Rooms’, ‘Avg. Area Number of Bedrooms’,
‘Area Population’]]

Y = HouseDF[‘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)

From sklearn.linear_model import
minmaxscaler

Lm = minmaxscaler(feature_range=(0,1))

Lm.fit_transform(X_train,y_train)
```

```
Print(lm.intercept_)

Coeff_df =
pd.DataFrame(lm.coef_,X.columns,columns
=['Coefficient'])

Coeff_df

From keras.layers import
Dense,Dropout,LSTM

From keras.models import Sequential

Model = Sequential()
```

```
Model.add(LSTM(units = 50,activation =
‘relu’,return_sequences = True

,input_shape = (x_train.shape[1], 1)))

Model.add(Dropout(0.2))

Model.add(LSTM(units = 60,activation =
‘relu’,return_sequences = True))

Model.add(Dropout(0.3))

Model.add(LSTM(units = 80,activation =
‘relu’,return_sequences = True))
model.add(Dropout(0.4))
model.add(LSTM(units = 120,activation =
'relu'))
model.add(Dropout(0.5))
model.add(Dense(units = 1))
model.compile(optimizer='adam', loss =
'mean_squared_error')
model.fit(x_train, y_train,epochs=50)
print(lm.intercept_)
coeff_df =
pd.DataFrame(lm.coeff_,X.columns,columns
=['Coefficient'])
coeff_df
predictions = lm.predict(X_test)
scale_factor = 1/0.02099517
y_predicted = y_predicted*scale_factory
y_test = y_test * scale_factor
```

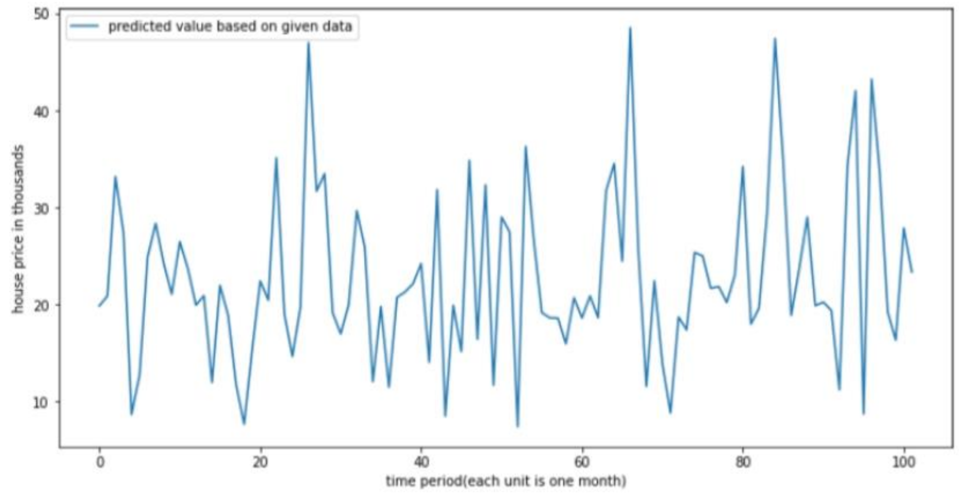
```
plt.scatter(y_test,predictions)
sns.distplot((y_test-predictions),bins=50);
plt.figure(figsize=(12,6))
plt.plot(y_test,'b',label = 'Original Price')
plt.plot(y_predicted,'r',label = 'Predicted
Price')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()
from sklearn import metrics
print('MAE:',
metrics.mean_absolute_error(y_test,
predictions))
```

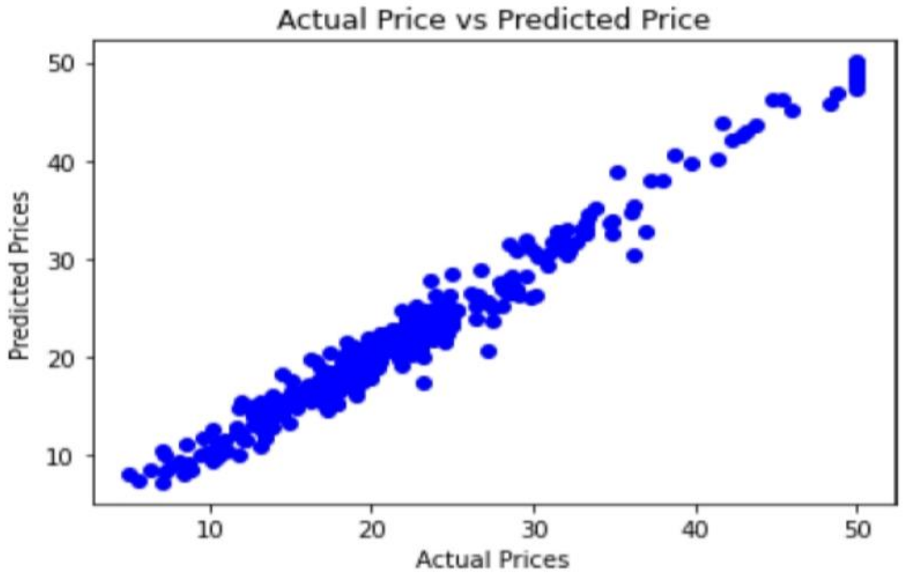
```
print('MSE:',
metrics.mean_squared_error(y_test,
predictions))
print('RMSE:',
np.sqrt(metrics.mean_squared_error(y_test
, predictions)))
```

OUTPUT:

Graph:-

PREDICTED VALUE OF HOUSE PRICE BASED ON TEST SAMPLE DATA





Conclusion:-

Executing the acquisitions
Identification of target market and properties
Building the plan
Evaluate resources

Thus the machine learning model to predict the house price

 Based on given dataset is executed successfully using xg Regressor (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.