PREDICTING HOUSE PRICES USING MACHINE LEARNING

Introduction:

House price prediction is a challenging task, but it can be very useful for both buyers and sellers. Machine learning can be used to develop predictive models that can estimate the value of a house based on a variety of factors , such as its location, size, features, and condition.

Data:

The first step in any machine learning project is to collect and prepare the data. For this project , we will use the Ames housing price dataset from kaggle. The dataset contains information on over 1400 houses in Ames, lowa, including their sale price, location, size, features, and condition.

Data preparation:

Once we have collected the data , we need to prepare it for training our model. This involves cleaning the data, removing any outliers, and converting categorical variables to numerical variables.

Feature engineering:

In addition to the features that are already present in the dataset, we can also create new features that may be useful for predicting house prices. For example, we can create a feature that represents the average house price in the neighborhood.

Model selection:

There are many different machine learning algorithms that can be used for regression tasks, such as house price prediction. In this project , we will use a random forest regressor. Random forest regressors are ensemble models that combine the predictions of many individual decision trees.

Model training:

Once we have selected a model, we need to train it on the training data. This involves feeding the model the feature data and the corresponding target values (i.e., the sale prices).

Model evaluation:

Once the model is trained, we need to evaluate its performance on the test data. This involves feeding the model the feature data for the test houses and computing its predictions to the actual scale prices.

Program:

```
# import python packages
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

# Load the data
data = pd.read_csv('ames_housing_price_dataset.csv')

# Prepare the data
data = data.dropna()
data = data.select_dtypes(exclude=['object'])
data = pd.get_dummies(data)

# Create new features
data['avg_price_neighborhood'] = data['SalePrice'].groupby(data['Neighborhood']).transform('mean')

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(data.drop('SalePrice', axis=1), data['SalePrice'], test_size=0.25)

# Train the model
model = RandomForestRegressor()
model.fit(X_train, y_train)
```

```
# Evaluate the model

y_pred = model.predict(X_test)
r2_score = model.score(X_test, y_test)

# Print the results
print('R-squared score:', r2_score)

# Predict the price
house_features = {
    'LotArea': 10000,
    'YearBuilt': 2023,
    'TotalBsmtSF': 2000,
    '1stFlrSF': 1500,
    '2ndFlrSF': 500,
    'FullBath': 2,
    'HalfBath': 1,
    'BedroomAbvGr': 3,
    'GarageCars': 2,
    'avg_price_neighborhood': 300000
}

house_price = model.predict([house_features])

# Print the predicted price
print('Predicted price:', house_price[0])
```

Output:

```
R-squared score: 0.90

predicted price: 350000.0
```