***AI***\_***phase4*** ***Predicting*** ***house*** ***prices*** ***using*** ***machine*** ***learning*** involves several phases: data preprocessing, feature selection, model training, and evaluation. Here’s a brief overview of each phase, and I’ll provide Python code for feature selection as an example:

\*\*Phase 1: Data Preprocessing\*\*

- Data cleaning and handling missing values.

- Encoding categorical variables.

- Scaling or normalizing numerical features.

- Splitting the data into training and testing sets.

\*\*Phase 2: Feature Selection\*\*

Feature selection is essential to choose the most relevant features for your model. Here’s an example using Python’s `scikit-learn` library with a hypothetical dataset:

```python

From sklearn.feature\_selection import SelectKBest

From sklearn.feature\_selection import f\_regression

# Assuming X contains your feature data and y contains target prices

X\_new = SelectKBest(score\_func=f\_regression, k=5).fit\_transform(X, y)

```

This code uses the F-regression method to select the top 5 features based on their relevance to predicting house prices.

\*\*Phase 3: Model Training\*\*

You can choose from various regression algorithms, such as Linear Regression, Random Forest, or Gradient Boosting. Here’s an example of training a simple Linear Regression model:

```python

From sklearn.linear\_model import LinearRegression

Model = LinearRegression()

Model.fit(X\_new, y)

```

\*\*Phase 4: Evaluation\*\*

You should evaluate the model’s performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or R-squared. Here's an example:

```python

From sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

Import numpy as np

Predictions = model.predict(X\_new)

Mae = mean\_absolute\_error(y, predictions)

Mse = mean\_squared\_error(y, predictions)

R2 = r2\_score(y, predictions)

Print(f”Mean Absolute Error: {mae}”)

Print(f”Mean Squared Error: {mse}”)

Print(f”R-squared: {r2}”)

``