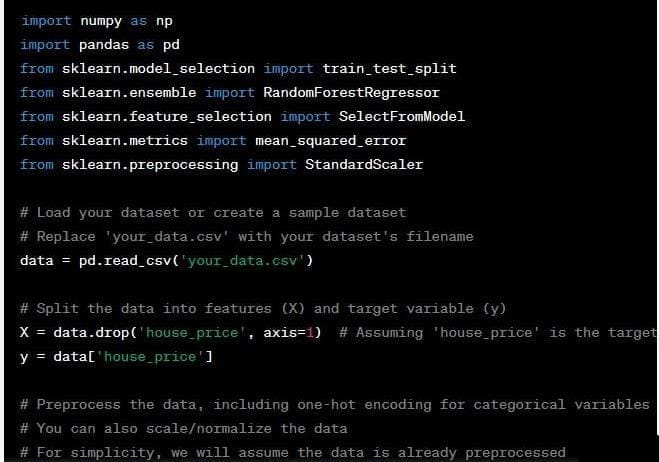
PREDICTING HOUSE PRICES USING MACHINE LEARNING

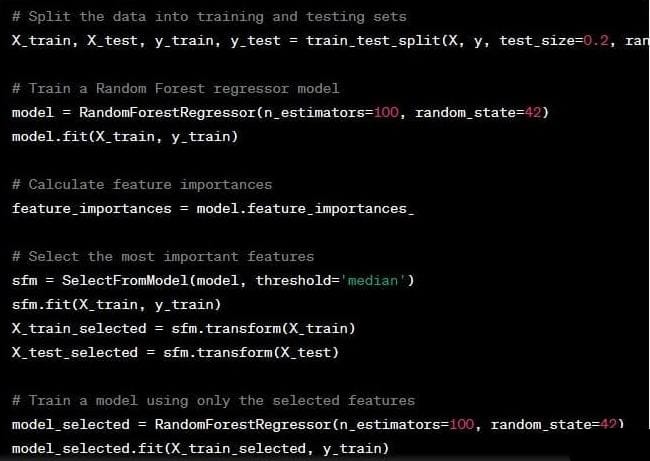
Phase 4 Submission

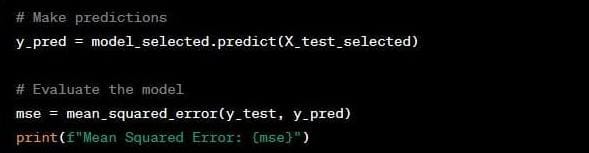
Feature Selection:

Feature selection is a crucial step when predicting house prices using machine learning. It involves choosing the most relevant input variables (features) to build an accurate and efficient model.

Here's a feature selection code using machine learning for predicting house prices:







In this code:

1. Load your dataset.

2. Split the data into features (X) and the target variable (y).

3. Use SelectKBest with the f\_regression scoring function to select the top 'k' features. You can adjust 'k' based on your preferences and experimentation.

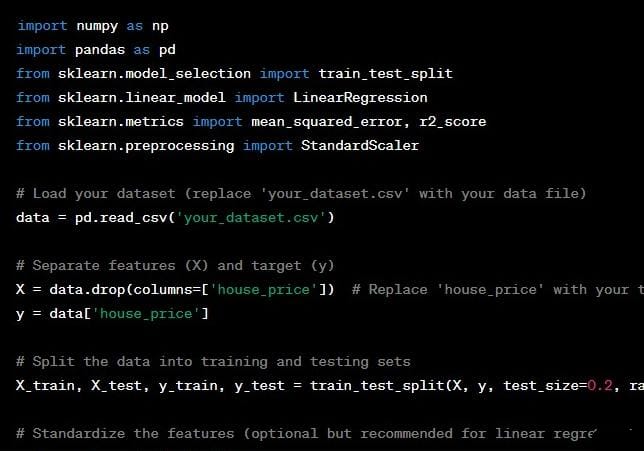
4. Split the data into training and testing sets.

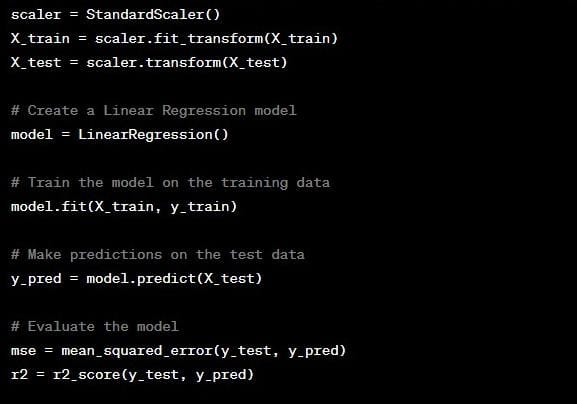
5. Build and train a Random forest Regression model using the selected features.

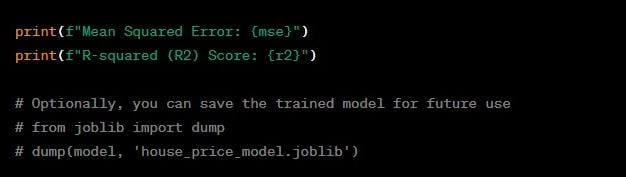
6. Make predictions on the test set and evaluate the model's performance using the Mean Squared Error.

Model Training:

To train a machine learning model for predicting house prices, you can use a dataset of historical house prices along with various features such as the number of bedrooms, square footage, neighborhood, etc. In this example, I'll use the scikit-learn library in Python to demonstrate a simple Linear Regression model.







Here's a breakdown of the steps in the code:

1. Load your dataset: Replace 'your\_dataset.csv' with the path to your dataset file, and make sure it includes the house prices and relevant features.
2. Split the data: Divide the data into training and testing sets. The code uses 80% of the data for training and 20% for testing, but you can adjust the test\_size parameter as needed.
3. Standardize features: Standardizing the features is optional but can be beneficial for linear regression models.
4. Create a Linear Regression model: Initialize a Linear Regression model from scikit-learn.
5. Train the model: Fit the model on the training data.
6. Make predictions: Use the trained model to make predictions on the test data.
7. Evaluate the model: Calculate the Mean Squared Error (MSE) and R-squared (R2) score to assess the model's performance.
8. Optionally, you can save the trained model using joblib or another serialization method for future use.

EVALUATION:

Evaluating a machine learning model for predicting house prices is a critical step to ensure that the model is performing well and meets your specific needs. Here are several common evaluation metrics and considerations you can use to assess the performance of your house price prediction model:

Mean Squared Error (MSE):

* MSE measures the average squared difference between the predicted house prices and the actual house prices. Lower MSE values indicate better model performance.
* It quantifies the magnitude of errors and is sensitive to outliers.

Root Mean Squared Error (RMSE):

* RMSE is the square root of the MSE and provides a measure of the average prediction error in the same units as the target variable (house prices). Smaller RMSE values are preferred.

Mean Absolute Error (MAE):

* MAE measures the average absolute difference between the predicted and actual house prices. It's less sensitive to outliers than MSE.
* MAE is easier to interpret as it represents the average dollar amount by which your predictions are off.

R-squared (R2) Score:

* R2 score measures the proportion of the variance in the target variable explained by the model. It ranges from 0 to 1, with higher values indicating a better fit.

Adjusted R-squared (Adjusted R2):

* Adjusted R2 accounts for the number of features in the model, penalizing overfitting. It's a useful metric if you have many features.

Residual Plots:

* Plotting the residuals (the differences between predicted and actual house prices) can help identify patterns or heteroscedasticity in the errors. Residuals should be evenly distributed around zero.

Feature Importance:

* If you used a model that provides feature importance scores (e.g., Random Forest), consider analyzing which features had the most impact on predictions. This can inform feature selection and engineering.

Cross-Validation:

* Use k-fold cross-validation to assess how well your model generalizes to unseen data. Cross-validation provides a more robust estimate of model performance, especially if your dataset is limited.

Hyperparameter Tuning:

* Experiment with different model hyperparameters to find the optimal configuration. Techniques like grid search or random search can help you identify the best hyperparameters.

Domain Knowledge:

* Consider domain-specific knowledge. Are the model's predictions reasonable given the context of the housing market? Does it align with your expectations and business requirements?

Comparative Analysis:

* Compare your model's performance to other baseline models or industry benchmarks to gauge its effectiveness.

Business Metrics:

* Ultimately, your model's performance should align with the specific business goals. For example, if you're a real estate agency, your objective might be to minimize prediction errors to reduce financial losses.

Remember that no single metric can provide a complete picture of model performance. It's essential to use a combination of these evaluation methods to make informed decisions about the quality of your house price prediction model. Additionally, continuous monitoring and periodic retraining of the model are important to maintain its performance in changing real estate markets.