PREDICTING HOUSE PRICES USING MACHINE LEARNING

Artificial Intelligence Phase-2 Document

Problem statement: Consider exploring advanced regression techniques like Gradient Boosting or XGBoost for improved prediction accuracy.

One effective machine learning method that is frequently employed in problems involving home price prediction is gradient boosting. By capturing complex correlations between multiple parameters (such as square footage, the number of bedrooms, location, etc.) and the objective variable (home prices), it can aid in enhancing forecast accuracy. Gradient boosting can be used to predict property prices in the following ways:

✞ **Data Preparation**:

- Start by collecting and preparing your dataset. Clean the data, handle missing values, and preprocess features (e.g., encoding categorical variables, scaling numeric features).

✞ **Train-Test Split**:

- Split the dataset into training and testing sets to evaluate the model's performance effectively. Common splits include 70% for training and 30% for testing.

✞ **Feature Engineering**:

- Create relevant features or transform existing ones. For house price prediction, this might involve computing features like the total square footage of the house, the age of the property, or the presence of specific amenities (e.g., pool, garage).

✞ **Gradient Boosting Model Selection**:

- Choose a Gradient Boosting algorithm suitable for regression tasks. Common options include:
- **Gradient Boosting Regressor**: A standard Gradient Boosting algorithm for regression tasks.

- **XGBoost (Extreme Gradient Boosting)**: An optimized and efficient version of Gradient Boosting, widely used in machine learning competitions.
- **LightGBM**: A gradient boosting framework designed for high efficiency and scalability.
- **CatBoost**: Another gradient boosting library that handles categorical features well.

Gradient boosting regressor:

The Gradient Boosting Regressor algorithm is a powerful machine learning technique used for predicting continuous numerical values, making it particularly well-suited for tasks like house price predictions. It builds an ensemble of decision trees sequentially, where each tree is trained to correct the errors made by the previous ones.

Choose the Gradient Boosting Regressor algorithm as the model for your house price prediction task. we can use libraries like Scikit-Learn for Python, which provide easy access to this algorithm.

Working of Gradient Boosting Regressor:

- The single decision tree that the Gradient Boosting Regressor starts with is frequently quite shallow (has a small depth). Because it makes unreliable predictions, this tree is referred to as a "weak learner".
- The method is concentrated on the residuals of the weak learner's mistakes. It figures out the discrepancies between the real target values and the forecasts given by the inexperienced learner.
- Progressive Boosting A new weak learner (another shallow decision tree) is then fitted by the regression to the residuals. This second tree's goal is to foretell the residuals, essentially rectifying the first tree's mistakes.
- The new weak learner's predictions are scaled by a variable known as the "learning rate" and added to those made by the preceding learner. Iteratively repeating this method until a stopping requirement is satisfied, for a predetermined number of iterations.

- Every time the algorithm adjusts the predictions and lowers the mistakes, it fits new weak learners to the residuals. The weighted average of all the learners' predictions forms the final forecast.

Accuracy in gradient boosting algorithm:

To assess the accuracy of your Gradient Boosting Regressor in house price predictions, you would typically calculate one or more of these metrics on a holdout or test dataset. Here's a general process:

- Utilise a training dataset to hone your Gradient Boosting Regressor.
- Make predictions using the trained model on a different test dataset.
- By contrasting the model's predictions with the actual home prices in the test dataset, determine the chosen evaluation metrics (for example, MAE, MSE, RMSE, and R2).

A model is more accurate and better at forecasting house values when it has a lower MAE, MSE, or RMSE and a higher R2. When analysing these metrics, it is crucial to take into account the particular context of your application and any applicable business requirements. In order to comprehend the nature of prediction errors and spot any patterns or trends in the errors, you might also wish to do residual analysis.

XGboost(Extreme Gradient Boosting):

Extreme Gradient Boosting, or XGBoost, is a well-known and effective machine learning technique that is a member of the gradient boosting method family. It is well known for its exceptional prediction performance and scalability and is commonly utilised for both regression and classification tasks. In machine learning contests, XGBoost is frequently a top contender and is also used to tackle practical issues in a variety of industries.

The gradient boosting framework is used by XGBoost. It develops an ensemble of decision trees successively, starting with an initial prediction (often the mean of the target values), to correct the mistakes caused by the earlier trees.

Accuracy of XG boosting in House Price Predictions:

1. Train the XGBoost model on a training dataset containing historical data with features (e.g., square footage, number of bedrooms, location) and corresponding actual house prices.
2. Use the trained model to make predictions on a separate test dataset containing similar features but with actual house prices withheld.
3. Calculate the chosen evaluation metrics (e.g., MAE, MSE, RMSE, R^2) by comparing the model's predictions to the actual house prices in the test dataset.
4. Interpret the evaluation metrics: A lower MAE, MSE, or RMSE and a higher R^2 indicate better accuracy, meaning that the XGBoost model's predictions are closer to the actual house prices.

When analysing these KPIs, it's crucial to take into account the particular context of your application and any applicable business requirements. Additionally, performing residual analysis will help you comprehend the types of errors made in predictions and see any patterns or trends that may help you improve your model moving forward.

Light GBM:

A robust and effective gradient boosting framework for machine learning is called LightGBM (Light Gradient Boosting Machine). LightGBM, created by Microsoft, is renowned for its strong predictive performance, speed, and ability to handle big datasets. It is frequently employed for both classification and regression applications and excels in situations where computational resources are constrained.

LightGBM follows the gradient boosting framework, which builds an ensemble of decision trees sequentially to improve predictive accuracy. It starts with an initial prediction (usually the mean of the target values) and progressively adds more trees to correct errors.

One of the distinguishing features of LightGBM is its use of histogram-based learning. Instead of using the traditional method of splitting data into continuous bins, LightGBM bins feature values into histograms, which reduces memory usage and speeds up training.

Accuracy of Light GBM in House Price Predictions:

1. Train the LightGBM model on a training dataset that includes historical data with features (e.g., square footage, number of bedrooms, location) and corresponding actual house prices.
2. Use the trained model to make predictions on a separate test dataset containing similar features but with actual house prices withheld.
3. Calculate the chosen evaluation metrics (e.g., MAE, MSE, RMSE, R^2) by comparing the model's predictions to the actual house prices in the test dataset.
4. Interpret the evaluation metrics: A lower MAE, MSE, or RMSE and a higher R^2 indicate better accuracy, meaning that the LightGBM model's predictions are closer to the actual house prices.

When analysing these data, it's crucial to take your application's unique context into account as well as any applicable business requirements.
Additionally, performing residual analysis will help you understand the types of errors that are made in predictions and spot any patterns or trends that could be used to direct future model improvements.

Cat Boost:

- CatBoost, short for Categorical Boosting, is a machine learning algorithm. It is designed to handle categorical features efficiently and is well-suited for a variety of tasks, including house price predictions.
- Prepare your dataset, including the goal variable (home pricing) and features (such as square footage, the number of bedrooms, and location). CatBoost is ideally suited for datasets with mixed data types due to the way it handles categorical characteristics.
- To evaluate your model, divide your dataset into a training set and a test set.
- Set the depth, learning rate, and regularisation parameters for the CatBoost hyperparameters.
- Utilise the training dataset to train the CatBoost model.

Accuracy of Cat Boost algorithm in House Price Predictions:

1. Train the CatBoost model on a training dataset that includes historical data th features (e.g., square footage, number of bedrooms, location) and rresponding actual house prices.
2. e the trained model to make predictions on a separate test dataset ntaining similar features but with actual house prices withheld. Calculate
3. e chosen evaluation metrics (e.g., MAE, MSE, RMSE, R^2) by comparing e model's predictions to the actual house prices in the test dataset. erpret the evaluation metrics: A lower MAE, MSE, or RMSE and a higher R^2
4. dicate better accuracy, meaning that the CatBoost model's predictions are oser to the actual house prices.

Conclusion:

An adaptable and reliable method for predicting home price trends is gradient boosting. It may produce precise and reliable predictions when used appropriately and with thorough model adjustment, making it a useful tool in the real estate and related industries. To create the most accurate predictive model for your particular use case, it's crucial to take into account aspects like data quality, feature engineering, and business objectives.