**House Pricing Prediction in R - Documentation**

This documentation explains the code written in R for predicting house pricing using the gradient boosting machine algorithm. The code performs various preprocessing steps, model training, evaluation, and generates predictions.

**Code Overview**

The code can be divided into the following sections:

1. **Library Loading**: The necessary libraries **caret**, **randomForest**, and **gbm** are loaded to provide the required functionalities.
2. **Data Loading**: The train and test datasets are read from CSV files using the **read.csv** function.
3. **Data Exploration**: The **ExploreData** function is defined to explore the loaded datasets. It displays the number of columns, number of rows, count of missing values, and summary statistics of the data.
4. **Removing Null Columns**: The **RemoveNullColumns** function drops columns that have more than a specified percentage of NULL values. It calculates the NULL percentage for each column, identifies columns exceeding the threshold, and removes those columns from the train and test datasets.
5. **Encoding Categorical Data**: The **EncodeCategoricalData** function performs one-hot encoding on categorical variables in the train and test datasets. It saves the target variable from the train dataset, removes the target variable, combines train and test datasets, applies one-hot encoding, and normalizes the data using Z-scores. It splits the encoded data back into train and test datasets.
6. **Filling Null Values**: The **FillNulls** function replaces NULL values in a dataset with the median value of each respective column.
7. **Preprocessing**: The **Preprocess** function integrates the preprocessing steps by calling the **RemoveNullColumns**, **EncodeCategoricalData**, and **FillNulls** functions. It returns the preprocessed train and test datasets, along with other necessary variables.
8. **Train-Validation Split**: The **TrainValidateSplit** function splits the preprocessed train dataset into training and validation sets. It adds IDs to the datasets, randomly partitions the data based on a specified split size, and returns the split datasets.
9. **Removing ID and Low Correlation Columns**: The **process\_train\_data** function removes the 'Id' column from the train dataset, and the **remove\_low\_correlation** function removes columns with low correlation to the target variable from the train dataset based on a specified threshold.
10. **Model Training**: The code fits the gradient boosting machine (GBM) model using the **gbm** function from the **gbm** library. The model is trained on the processed train dataset.
11. **Model Evaluation**: The code evaluates the trained model's performance on the training dataset by calculating the mean squared error (MSE), root mean squared error (RMSE), and R-squared value. It visualizes the actual vs. predicted values and the distribution of residuals.
12. **Validation Predictions**: The code predicts the house prices for the validation dataset using the trained model and evaluates the predictions' accuracy. It again visualizes the actual vs. predicted values and the distribution of residuals.
13. **Test Set Predictions**: The code predicts the house prices for the test dataset using the trained model and creates a data frame with the sequential 'Id' and 'SalePrice' columns.
14. **Saving Predictions**: The code saves the predictions as a CSV file named "predictions.csv".

**Evaluation Metrics - Training Dataset**

The evaluation metrics for the training dataset are as follows:

* Mean Squared Error (MSE): 249,788,746
  + The MSE measures the average squared difference between the actual and predicted house prices in the training dataset. It quantifies the overall quality of the model's predictions, with lower values indicating better performance.
* Root Mean Squared Error (RMSE): 15,804.71
  + The RMSE is the square root of the MSE and represents the average difference between the actual and predicted house prices in the training dataset. It provides a measure of the model's prediction accuracy, with lower values indicating better performance.
* R-squared (R²): 0.9619353
  + The R-squared value represents the proportion of the variance in the target variable (house prices) that can be explained by the model. It ranges from 0 to 1, where a higher value indicates a better fit. In this case, the model explains approximately 96.19% of the variance in the training dataset's house prices.

**Evaluation Metrics - Validation Dataset**

The evaluation metrics for the validation dataset are as follows:

* Mean Squared Error (MSE): 1,912,787,428
  + The MSE measures the average squared difference between the actual and predicted house prices in the validation dataset. A lower MSE value indicates better performance and suggests that the model's predictions are closer to the true values.
* Root Mean Squared Error (RMSE): 43,735.43
  + The RMSE is the square root of the MSE and represents the average difference between the actual and predicted house prices in the validation dataset. A lower RMSE value indicates better prediction accuracy and implies that the model's predictions are closer to the true values.
* R-squared (R²): 0.710436
  + The R-squared value represents the proportion of the variance in the target variable (house prices) that can be explained by the model. It ranges from 0 to 1, where a higher value indicates a better fit. In this case, the model explains approximately 71.04% of the variance in the validation dataset's house prices.