### **Problem Statement**

### The goal of this project is to predict housing prices using linear regression. The dataset contains various features related to houses, such as the number of rooms, area, location, etc. The target variable is the sale price of the houses.

### The code you provided is a great start for cleaning data for house price prediction! Here's a breakdown of what the code does:

1. **Import libraries:** This section imports necessary libraries for data manipulation, visualization, and modeling.
2. **Read data:** The code reads the training data (train.csv) and test data (test.csv) from the specified paths.
3. **Explore data types and missing values:** It uses train.info() to understand the data types of each feature and train.describe() to get summary statistics. Then, it identifies features with missing values using train.isnull().any().
4. **Identify categorical features:** It defines two lists, categorical\_data and numerical\_data, to categorize features based on their data types (object or string for categorical and others for numerical).
5. **Handle missing values:** It fills missing values in categorical features with "n/a" and numerical features with 0 (although this approach might require further investigation depending on the data).
6. **Address dominant categories:** The code checks for categorical features where a single category represents more than 90% of the data. It then creates a new binary feature indicating whether the data point belongs to the dominant category. This helps the model avoid overfitting on that specific category.
7. **Visualize the Sale Price distribution:** It uses Plotly Express (px.violin) to visualize the distribution of the Sale Price. Additionally, it creates a box plot (px.box) to explore the relationship between Sale Price and Overall Quality. Finally, it uses Seaborn (sns.histplot) to create a histogram with a density curve to visualize the distribution of Sale Price.

Overall, this code provides a solid foundation for data cleaning in house price prediction. Here are some additional considerations:

* **Feature engineering:** You might want to explore creating new features based on existing ones (e.g., total area, year built difference from renovation year).
* **Outlier handling:** It might be beneficial to identify and address outliers in numerical features, especially for Sale Price.
* **Encoding categorical features:** Before feeding the data into the model, you'll likely need to encode categorical features (e.g., one-hot encoding or label encoding).
* **Model selection and hyperparameter tuning:** After data cleaning, you can explore different machine learning models (e.g., XGBoost as shown in the code) and tune their hyperparameters to improve the model's performance.