Your cleaning strategy is generally well thought-out, but here are my recommendations and reasoning for each column, along with additional tips to refine your approach:

**1. address (Drop column)**

* **Reason**: High cardinality (2000 distinct values). Likely does not provide meaningful grouping information for predictive modeling.

**2. agent name (Drop column)**

* **Reason**: High cardinality (1900 distinct values). Likely irrelevant to the analysis.

**3. agent url (Drop column)**

* **Reason**: Same as above; URLs are not useful for most ML tasks.

**4. available (15.5% None)**

* **Action**:
  + **If it’s categorical (e.g., availability status)**: Impute missing values with a mode or create a new category, such as Unknown.
  + **If it’s a date or time range**: Convert to datetime and consider feature engineering (e.g., days from now).

**5. balcony (8.7% None, imbalanced)**

* **Action**: Keep and handle missing values by imputing 0 (no balcony) or creating an Unknown category.
* **Reason**: This could be relevant for pricing or desirability.

**6. city (Check if cleaning is needed)**

* **Action**:
  + Normalize values (e.g., capitalization, spelling issues).
  + Check for consistency and group rare cities into "Other" if needed.

**7. construction type (7.3% None, super imbalanced)**

* **Action**:
  + Keep as-is but consider grouping rare classes into an "Other" category.
  + Impute missing values with the mode (existing building).

**8. deposit (37.6% None, drop?)**

* **Action**: Drop if domain knowledge suggests it’s not crucial.
* **Reason**: High missingness, and its impact on price may be minimal.

**9. district (Drop column?)**

* **Action**:
  + Check if it correlates strongly with price or city. If not, drop.
  + High cardinality can complicate modeling.

**10. dwelling type (Imbalanced)**

* **Action**: Keep. It may still hold predictive value despite imbalance.
* **Reason**: Dwelling type (e.g., apartment, house) is often critical for price prediction.

**11. energy\_rating (21.1% missing, string → integer)**

* **Action**:
  + Convert to integer scale (e.g., A=1, B=2, C=3).
  + Impute missing values based on similar properties (e.g., city, construction type) or create a new Unknown category.

**12. house name (Drop column)**

* **Reason**: High cardinality; unlikely to be useful.

**13. house\_url (Drop column)**

* **Reason**: URLs are irrelevant for analysis.

**14. description (Drop)**

* **Reason**: Text data requires extensive processing. If you don’t intend to analyze text (e.g., via NLP), dropping is reasonable.

**15. interior (16.8% None)**

* **Action**:
  + Map categories like "upholstered" and "furnished" to simpler labels, or treat as categorical.
  + Impute missing values as Unknown.

**16. maximum\_month (88.7% None, drop)**

* **Reason**: Excessive missingness; unlikely to provide sufficient information.

**17. minimum\_month (78% None, drop)**

* **Reason**: Similar to above.

**18. num\_of\_bathrooms (30% None, drop column)**

* **Action**: Only drop if it has low correlation with price. Bathrooms are often predictive of property value.

**19. num\_of\_bedrooms (18.5% None, drop column)**

* **Action**: Same as above. Retain if predictive of price.

**20. num\_of\_rooms (Good)**

* **Action**: Keep. It’s likely highly correlated with price.

**21. offered\_since (Convert to datetime)**

* **Action**: Convert to datetime. Consider feature engineering:
  + Days on market (current\_date - offered\_since).
  + Categorize into recent/old listings.

**22. pets\_allowed (58.9% None, imbalanced)**

* **Action**: Impute missing values as Unknown or Not Allowed.

**23. price (Target)**

* **Action**: Ensure this is clean and properly scaled (e.g., log-transform if skewed).

**24. property\_type (7.3% None)**

* **Action**: Impute missing values with the mode or Unknown.

**25. rental\_agreement (49.6% None)**

* **Action**: Drop if domain knowledge suggests it’s not critical.

**26. service\_cost (40.4% None, drop)**

* **Reason**: High missingness and likely minimal impact on target.

**27. smoking\_allowed (62.8% None)**

* **Action**: Drop unless strongly correlated with price or demand.

**28. status (93.7% "for rent")**

* **Action**: Drop. No predictive power due to imbalance.

**29. year\_of\_construction (Change to age, drop None)**

* **Action**: Convert to age = current\_year - year\_of\_construction.
* Drop rows with missing values if it’s a small fraction.

**Additional Tips:**

1. **Correlation Analysis**:
   * Before dropping columns, check their correlation with price (target). Keep features with moderate to high correlation.
2. **Feature Engineering**:
   * Combine related features (e.g., construction type and year\_of\_construction).
3. **Data Imputation**:
   * Use domain knowledge or related features to fill missing values.
4. **Imbalance Handling**:
   * Retain imbalanced features but carefully evaluate their importance.

Let me know if you'd like help automating this cleaning process in Python!