[innercity (kaggle.com)](https://www.kaggle.com/datasets/prasy46/innercity)

A house's value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value. For example, if I want to sell a house and I don’t know the price that I may expect — it can’t be too low or too high. To find a house price we usually try to find similar properties in the neighborhood and based on gathered data we will try to assess the house price.

**Right pricing is very imporatnt aspect to sell house. It is very important to understand what are the factors and how they influence the house price. Objective is to predict the right price of the house based on the attributes**

**As people don't know the features/aspects which commulate property price, we can provide them HouseBuyingSelling guiding services in the area so they can buy or sell their property with most suitable price tag and they didn't lose their hard earned money by offering low price or keep waiting for buyers by putting high prices.**

**Find out the significant features from the given features dataset which affects the house price the most.**

Objective: Take advantage of all of the feature variables available below, and use them to

analyze and predict house prices.

1. cid: a notation for a house

2. day hours: Date house was sold

3. price: Price is prediction target

4. room\_bed: Number of Bedrooms/House

5. room\_bath: Number of bathrooms/bedrooms

6. living\_measure: square footage of the home

7. lot\_measure: square footage of the lot

8. ceil: Total floors (levels) in the house

9. coast: House which has a view of a waterfront

10. sight: Has been viewed

11. condition: How good the condition is (Overall)

12. quality: grade is given to the housing unit, based on the grading system

13. ceil\_measure: square footage of house apart from the basement

14. basement\_measure: square footage of the basement

15. yr\_built: Built Year

16. yr\_renovated: The year when the house was renovated

17. zip code: zip

18. lat: Latitude coordinate

19. long: Longitude coordinate

20. living\_measure15: Living room area in 2015(implies-- some renovations) This might or

might not have affected the lot size area

21. lot\_measure15: lot Size area in 2015(implies-- some renovations)

22. furnished: Based on the quality of the room

23. total area: Measure of both living and lot

Numerical variables are those that represent measurable quantities and can be represented by numbers. They include variables such as age, height, weight, temperature, and price.

Categorical variables are those that represent characteristics or categories and can take on a limited, fixed number of values. They include variables such as gender, color, country of origin, and type of car.

num = ['price','living\_measure','lot\_measure',

    'ceil\_measure', 'basement', 'living\_measure15', 'lot\_measure15','total\_area']

cat = ['room\_bed', 'room\_bath', 'ceil', 'coast', 'sight', 'condition', 'quality',' furnished']

**Data Preprocessing:**

When I went through my data I have observed some symbols like, $ in few of the variables like condition, ceil, coast, and they are seen as object datatype when they should be either integer values or float values, so replaced the '$' symbol with null values and then converted the affected columns to floats or ints, is typically referred to as "data cleaning" or "data preprocessing."

In this context, data cleaning involves identifying and addressing issues with the data that could affect its quality or usability for analysis or modeling. This can include tasks such as handling missing values, correcting data types, removing special characters or symbols, and addressing outliers or inconsistencies.

By replacing the '$' symbol with null values and converting the affected columns to numerical data types (floats or ints), you have effectively cleaned the data and made it more suitable for analysis or modeling tasks. This process ensures that the data is in a format that can be properly interpreted by machine learning algorithms or statistical techniques.

Overall, data cleaning is an essential step in the data analysis process as it helps ensure that the data is accurate, consistent, and ready for further exploration or modeling.

Then I tried to get some insights into the diversity of values present in each column, which can be useful for understanding the distribution and characteristics of the data.  
and I have been able to separate the numerical and categorical variables,

num = ['price','living\_measure','lot\_measure', 'ceil\_measure', 'basement', 'living\_measure15', 'lot\_measure15','total\_area','long','yr\_built']

cat = ['room\_bed', 'room\_bath', 'ceil', 'coast', 'sight', 'condition', 'quality','furnished','yr\_built']

next I have tried to see if there are any missing values in my data

**MISSING VALUE TREATMENT:**

|  |  |
| --- | --- |
| cid | 0 |
| dayhours | 0 |
| price | 0 |
| room\_bed | 108 |
| room\_bath | 108 |
| living\_measure | 17 |
| lot\_measure | 42 |
| ceil | 72 |
| coast | 31 |
| sight | 57 |
| condition | 85 |
| quality | 1 |
| ceil\_measure | 1 |
| basement | 15 |
| yr\_built | 0 |
| yr\_renovated | 0 |
| zipcode | 0 |
| lat | 0 |
| long | 34 |
| living\_measure15 | 166 |
| lot\_measure15 | 29 |
| furnished | 29 |
| total\_area | 68 |

Median imputation is commonly used for numerical variables because it is robust to outliers and preserves the distribution of the data. It replaces missing values with the median of the available values in the respective column, ensuring that the central tendency of the variable remains unchanged. This method is especially useful when the data is skewed or contains outliers, as it is less sensitive to extreme values compared to mean imputation.

Mode imputation, on the other hand, is typically used for categorical variables. It replaces missing values with the mode, i.e., the most frequently occurring value in the respective column. Mode imputation is suitable for categorical variables because it preserves the categorical nature of the data and ensures that the imputed values are representative of the majority class. However, it may not be the best choice if the mode does not accurately reflect the underlying distribution of the data or if there is high variability in the categorical values.

Other types of imputation methods include mean imputation, which replaces missing values with the mean of the available values (suitable for normally distributed data without outliers), and regression imputation, which predicts missing values based on the relationship with other variables in the dataset (suitable for cases where variables are correlated).

**KNNImputer Data Transform**

*KNNImputer* is a data transform that is first configured based on the method used to estimate the missing values.

The default distance measure is a Euclidean distance measure that is NaN aware, e.g. will not include NaN values when calculating the distance between members of the training dataset. This is set via the “*metric*” argument.

The number of neighbors is set to five by default and can be configured by the “*n\_neighbors*” argument.

Finally, the distance measure can be weighed proportional to the distance between instances (rows), although this is set to a uniform weighting by default, controlled via the “*weights*” argument.

* Imputation should not be used if over 50% of data are missing (some authors use lower cutoffs, such as 20%).
* **### Multiple Imputation by Chained Equation:**
* MICE uses multiple imputation instead of single imputation which results in statistical uncertainty. MICE perform multiple regression over the sample data and take averages of them

Ultimately, the choice of imputation method depends on the nature of the data, the distribution of the variables, and the specific characteristics of the missing values. It is important to carefully consider these factors and select the most appropriate imputation strategy to ensure the integrity and accuracy of the data analysis.

**Descriptive statistics:**

Overall, these descriptive statistics provide insights into the distribution, variability, and range of each variable in your dataset, which can help guide further analysis and decision-making processes.

Price varies from $ 75,000 to $7,700,000, The median (50th percentile) price is $450,000, which indicates that half of the houses have prices below this value.

- The lot measure ranges from 520 square feet to 1,651,359 square feet, showing significant variability in the sizes of lots. living\_measurement, ceil\_measurement, basement measurement and total area.

this might mean there are outliers in the data.

1. **Price (House Price):**
   * The mean house price is $540,182.16, with a standard deviation of $367,362.23.
   * The minimum price is $75,000, and the maximum price is $7,700,000.
   * There are outliers with prices above $1,129,575. These could represent luxury properties or anomalies in the dataset.
2. **Living Measure (Square Footage of the Home):**
   * The mean living measure is 2,079.86 square feet, with a standard deviation of 918.50 square feet.
   * The minimum living measure is 290 square feet, and the maximum living measure is 13,540 square feet.
   * There are outliers with living measures above 4,230 square feet. These could represent large or unique properties.
3. **Lot Measure (Square Footage of the Lot):**
   * The mean lot measure is 15,104.58 square feet, with a standard deviation of 41,423.62 square feet.
   * The minimum lot measure is 520 square feet, and the maximum lot measure is 1,651,359 square feet.
   * There are outliers with lot measures above 19,085.5 square feet. These could represent large or irregularly shaped lots.
4. **Ceil Measure (Square Footage of the House Apart from the Basement):**
   * The mean ceil measure is 1,788.37 square feet, with a standard deviation of 828.10 square feet.
   * The minimum ceil measure is 290 square feet, and the maximum ceil measure is 9,410 square feet.
   * There are outliers with ceil measures above 3,740 square feet. These could represent large or unique properties.
5. **Basement (Square Footage of the Basement):**
   * The mean basement size is 291.52 square feet, with a standard deviation of 442.58 square feet.
   * The minimum basement size is 0 square feet, and the maximum basement size is 4,820 square feet.
   * There are outliers with basement sizes above 1,400 square feet.
6. **Living Measure15 (Living Room Area in 2015):**
   * The mean living measure in 2015 is 1,987.07 square feet, with a standard deviation of 685.52 square feet.
   * The minimum living measure in 2015 is 399 square feet, and the maximum living measure in 2015 is 6,210 square feet.
   * There are outliers with living measures in 2015 above 3,665 square feet.
7. **Lot Measure15 (Lot Size Area in 2015):**
   * The mean lot measure in 2015 is 12,766.54 square feet, with a standard deviation of 27,286.99 square feet.
   * The minimum lot measure in 2015 is 651 square feet, and the maximum lot measure in 2015 is 871,200 square feet.
   * There are outliers with lot measures in 2015 above 17,550 square feet.
8. **Total Area (Measure of Both Living and Lot):**
   * The mean total area is 17,192.04 square feet, with a standard deviation of 41,628.69 square feet.
   * The minimum total area is 1,423 square feet, and the maximum total area is 1,652,659 square feet.
   * There are outliers with total areas above 21,865 square feet.
9. **Keep Them As Is:**
   * If the outliers represent genuine data points and are not due to errors or anomalies, you may choose to keep them in the dataset. This approach preserves the original distribution of the data but may affect the performance of some models sensitive to outliers.
10. **Transform the Data:**
    * Applying mathematical transformations such as logarithmic, square root, or Box-Cox transformations can help reduce the impact of outliers and make the data more normally distributed. This approach is useful when the outliers have a disproportionate influence on the analysis.
11. **Remove Outliers:**
    * You can remove outliers from the dataset if they are considered anomalies or errors and are not representative of the underlying population. This can be done using statistical methods such as Z-score, IQR (Interquartile Range), or visual inspection.
12. **Winsorization:**
    * Winsorization involves capping or clipping extreme values at a certain percentile (e.g., 95th percentile). This method preserves the overall distribution while reducing the influence of outliers.
13. **Create a Separate Category:**
    * For categorical variables, you can create a separate category to represent outliers. This allows you to retain the information while mitigating their impact on the analysis.
14. **Model Robustness:**
    * Some machine learning algorithms are inherently robust to outliers, such as tree-based models (e.g., Random Forest, Gradient Boosting Machines). In such cases, outliers may have minimal impact on model performance.
15. **Domain Knowledge:**
    * Finally, consider consulting domain experts to determine the most appropriate approach for handling outliers in your specific context. Their insights can help guide decision-making based on the nature of the data and the goals of the analysis.
16. **Frequency Distribution**: Count plots show the frequency distribution of the number of bedrooms. We can observe how common different numbers of bedrooms are in the dataset.
17. **Mode of Bedrooms**: The mode, or the most frequently occurring number of bedrooms, can be identified from the count plot. This is helpful in understanding the typical configuration of bedrooms in the houses in the dataset.

Average Price Variation: You can observe how the average price varies across different categories of each categorical variable. This helps in understanding the relationship between the categorical variable and the target variable (price).

Impact of Categorical Variable: The plots reveal whether the categories of the categorical variable have a significant influence on the average price. For example, if you see large variations in average price between categories, it indicates that the categorical variable is likely to be a significant predictor of house prices.

The presence of many outliers in the living\_measure variable suggests that there are some houses with unusually large living areas compared to the majority of the houses in the dataset. These outliers could represent luxury properties, estates, or unique architectural designs that have significantly larger living spaces than typical homes.

A histogram is a graphical representation of the distribution of numerical data. It consists of a series of bars, where the height of each bar represents the frequency or count of data points falling within a specific range or "bin" of values. Histograms are commonly used in data analysis and visualization to understand the underlying distribution of a dataset and to identify patterns, trends, and anomalies.

Here's what a histogram tells us and why we use histograms for numerical variables:

1. **Distribution**: A histogram provides insight into the shape of the distribution of the data. It helps us understand whether the data is symmetric or skewed, and whether it follows a normal distribution or has other characteristic shapes such as bimodal or multimodal distributions.
2. **Central Tendency and Spread**: By examining the center and spread of the distribution, we can gain information about the central tendency (mean, median) and variability (standard deviation, interquartile range) of the data.
3. **Outliers**: Histograms help identify outliers, which are data points that significantly deviate from the majority of the data. Outliers may indicate errors in data collection, measurement issues, or genuine anomalies that require further investigation.
4. **Data Quality**: Histograms can reveal patterns and irregularities in the data, such as gaps, clusters, or unusual patterns, which may indicate data quality issues or interesting phenomena in the dataset.
5. **Comparison**: Histograms allow for the comparison of distributions between different groups or datasets. By visualizing multiple histograms side by side, we can compare the distribution of numerical variables across different categories or conditions.
6. **Calculate Upper and Lower Limits**: Calculate the upper and lower limits for each numerical variable based on a chosen method, such as the Interquartile Range (IQR) method or standard deviation method. For example, using the IQR method:
   * Upper Limit (UL) = Q3 + k \* IQR
   * Lower Limit (LL) = Q1 - k \* IQR Where Q1 and Q3 are the first and third quartiles, respectively, and IQR is the interquartile range. The value of k determines how strict the outlier treatment will be.
7. **Identify Outliers**: Determine which data points fall outside the calculated upper and lower limits for each variable. These data points are considered outliers.
8. **Cap Outliers**: Replace the outliers with the corresponding upper or lower limit. This effectively "caps" the extreme values, bringing them within an acceptable range.

Considering that you have 21,613 rows in your dataset, having counts of outliers ranging from a few hundred to a couple of thousand can indeed be considered relatively small compared to the total number of rows.

However, the significance of these outliers depends on their impact on your analysis and the specific requirements of your project. While the absolute count of outliers may seem small in comparison to the total number of rows, their influence on your analysis could still be substantial, especially if they significantly affect the distribution and statistical properties of your data.

Therefore, it's essential to carefully evaluate the outliers and consider how they might impact your analysis, model performance, and the validity of your conclusions.

**Presentation Summary: House Price Prediction**

1. **Introduction**:
   * Selling or buying a house involves various factors beyond location and square footage.
   * To accurately estimate a house's value, we need to consider multiple aspects.
2. **Objective**:
   * The objective is to predict house prices using available features, aiding both sellers and buyers in making informed decisions.
3. **Data Description**:
   * The dataset comprises various features such as the number of bedrooms and bathrooms, square footage of the home and lot, house condition, presence of amenities like a waterfront view, and more.
   * Price serves as the target variable for prediction.
4. **Data Preprocessing**:
   * Missing values were addressed using median and mode imputation for numerical and categorical variables, respectively.
   * Outliers were treated using upper and lower limit capping based on the Interquartile Range (IQR) method.
5. **Exploratory Data Analysis (EDA)**:
   * Visualizations such as histograms, box plots, and scatter plots were utilized to gain insights into the distribution and relationships among variables.
   * Analysis of numerical variables revealed right-skewed distributions and the presence of outliers.
   * Categorical variables were explored through count plots and bar plots to understand their distribution and impact on house prices.
6. **Analysis Results**:
   * The EDA highlighted significant factors influencing house prices, including living space, lot size, and house condition.
   * Relationships between certain features and house prices were observed, providing valuable insights for prediction models.
7. **Next Steps**:
   * Further analysis could involve feature engineering to create new variables or feature selection to identify the most predictive features.
   * Machine learning models, such as regression or ensemble methods, can be trained on the preprocessed data to predict house prices accurately.
8. **Conclusion**:
   * By leveraging data analysis techniques and predictive modeling, stakeholders can make more informed decisions when buying or selling houses, ultimately maximizing value and satisfaction.

**Key Observations from Data Analysis**

1. **Missing Values and Outliers**:
   * Identified missing values in several variables and applied appropriate imputation techniques for data completeness.
   * Detected outliers using statistical methods such as the Interquartile Range (IQR) and addressed them through capping or removal to ensure data quality and model accuracy.
2. **Correlation Analysis**:
   * Explored correlations among numerical variables to understand their relationships and potential impact on house prices.
   * Investigated the correlation between independent variables (e.g., living space, lot size) and the response variable (house price) to identify significant predictors.
3. **Effect of Categorical Variables**:
   * Examined how different levels of categorical variables influence average house prices.
   * Utilized visualization techniques such as bar plots and point plots to illustrate the relationship between categorical variables (e.g., house condition, presence of amenities) and house prices.

**Conclusion:**

* Through thorough data analysis, we've gained insights into the factors influencing house prices, including numerical variables such as living space and lot size, as well as categorical variables like house condition and amenities.
* Understanding these relationships is crucial for accurately predicting house prices and making informed decisions in the real estate market.

Got it. We'll calculate the age of the houses as you described, using the current year (2024). We'll replace the 'yr\_built' and 'yr\_renovated' columns with the calculated age.

This code calculates the age of each house based on the current year (2024), following the logic you provided. Then, it creates a new column 'house\_age' in the DataFrame and drops the 'yr\_built' and 'yr\_renovated' columns.

So, in terms of feature engineering I have calculated the house age, and dropped the variables lot measure and living measure since they seemed to be giving same info as living measure 15 and lot measure 15, also dropped, cid, dayhours, zipcode, lat long columns

1. **Geospatial Analysis**:
   * Visualize the distribution of houses on a map using latitude and longitude coordinates.
   * Analyze spatial patterns and clusters of houses based on their geographical locations.
   * Explore relationships between house prices and geographical features such as proximity to amenities, distance to city centers, or elevation.
2. **Temporal Analysis**:
   * Analyze trends in house prices over time by grouping sales data by year, month, or season.
   * Identify seasonal patterns in house sales and prices.
   * Investigate the effect of specific events or economic factors on house prices over time.
3. **Zip Code Analysis**:
   * Analyze house prices and other features across different zip codes to identify areas of high demand or value.
   * Compare average house prices, sizes, or other features between different zip codes.
   * Explore demographic or socio-economic factors associated with different zip code areas.
4. **Day of the Week Analysis**:
   * Analyze whether there are any patterns in house sales based on the day of the week.
   * Compare average prices or sales volume between weekdays and weekends.
   * Identify any seasonality or trends in house sales based on the day of the week.

**scaling:**

In essence, scaling helps to normalize the data and ensure that each feature has an equal impact on the clustering process, leading to more robust and interpretable clustering outcomes. it helps to ensure that all features contribute equally to the computation of distances or similarities between data points. Without scaling, features with larger magnitudes or variances can dominate the distance calculations, leading to biased results and potentially inaccurate clustering.

**Cluster results:**

1. **Price**: Cluster 0 has a significantly higher average price compared to Cluster 1. This suggests that houses in Cluster 0 may belong to a higher-priced segment of the market.
2. **Room Features**: Cluster 0 has slightly more bedrooms and bathrooms on average compared to Cluster 1. This could indicate that houses in Cluster 0 are larger or more luxurious.
3. **Ceiling Features**: Cluster 0 has a higher average number of floors (ceilings) compared to Cluster 1, indicating that houses in Cluster 0 may be multi-story or have higher ceilings.
4. **Coast and Sight**: Cluster 0 has a higher proportion of houses with a view of the waterfront (coast) and a higher average number of times the house has been viewed. This suggests that houses in Cluster 0 may have better views or more desirable locations.
5. **Condition and Quality**: Cluster 0 has slightly higher average values for condition and quality ratings compared to Cluster 1, indicating that houses in Cluster 0 may be in better condition and of higher quality.
6. **Living and Lot Measures**: Cluster 0 has larger average living and lot measures compared to Cluster 1, indicating that houses in Cluster 0 may be larger in size.
7. **Furnished**: Cluster 0 has a higher proportion of furnished houses compared to Cluster 1, suggesting that houses in Cluster 0 may be more likely to be furnished.
8. **House Age**: Cluster 0 has a lower average house age compared to Cluster 1, indicating that houses in Cluster 0 may be relatively newer.
9. **Frequency**: The frequency (or number of data points) in Cluster 0 is much lower compared to Cluster 1, suggesting that Cluster 0 represents a smaller segment of the dataset but with higher-priced houses.

Here I have done linear regression on the two clusters separately, and noticed that the the clusters are only able to capture less than 50% of the variance. And the variability in the clusters might not reflect the variability of entire population. Also may be because of the categorical variables in the data.

In a decision tree model, the parameters **max\_depth**, **min\_samples\_leaf**, and **min\_samples\_split** control the structure and complexity of the tree. Here's what each parameter means:

1. **max\_depth**:
   * This parameter controls the maximum depth of the decision tree.
   * A higher value of **max\_depth** allows the tree to grow deeper, resulting in more complex decision boundaries.
   * Increasing **max\_depth** can lead to overfitting, as the model may capture noise or outliers in the training data.
   * Setting **max\_depth** too low may result in underfitting, as the model may not capture important patterns in the data.
2. **min\_samples\_leaf**:
   * This parameter specifies the minimum number of samples required to be at a leaf node (a terminal node) of the tree.
   * A higher value of **min\_samples\_leaf** prevents the tree from splitting nodes that have fewer samples than the specified threshold.
   * Increasing **min\_samples\_leaf** can help prevent overfitting by enforcing a simpler tree structure with fewer splits.
   * However, setting **min\_samples\_leaf** too high may result in underfitting, as the tree may be too shallow to capture the underlying patterns in the data.
3. **min\_samples\_split**:
   * This parameter specifies the minimum number of samples required to split an internal node.
   * A higher value of **min\_samples\_split** prevents the tree from making splits that result in fewer samples in either of the split nodes.
   * Increasing **min\_samples\_split** can also help prevent overfitting by limiting the number of splits and enforcing a simpler tree structure.
   * However, setting **min\_samples\_split** too high may result in underfitting, as the tree may not be able to capture detailed patterns in the data.

The values provided in your example (**max\_depth**: 20, **min\_samples\_leaf**: 30, **min\_samples\_split**: 15) indicate the specific settings for these parameters that were selected as the best hyperparameters during the grid search. These settings will be used to train the final decision tree model.

Hyperparameters are parameters that are set before the learning process begins and cannot be directly learned from the data. Examples of hyperparameters include the learning rate in gradient descent, the number of layers in a neural network, the depth of a decision tree, and the regularization parameter in regression models.

Hyperparameter tuning involves selecting the optimal values for these hyperparameters to maximize the performance of the model on unseen data. This is typically done through an iterative process of training the model with different combinations of hyperparameters and evaluating its performance using a validation set or cross-validation.

There are several techniques for hyperparameter tuning:

1. **Manual Search**: This involves manually selecting hyperparameter values based on domain knowledge and intuition. While simple, this approach can be time-consuming and may not always yield the best results.
2. **Grid Search**: Grid search involves defining a grid of hyperparameter values and evaluating the model's performance for each combination of values using cross-validation. The combination of hyperparameters that yields the best performance is selected as the optimal set.
3. **Random Search**: Random search randomly samples hyperparameter values from specified distributions and evaluates the model's performance for each sampled set of values. Random search is often more efficient than grid search and can sometimes yield better results, especially when some hyperparameters are more important than others.
4. **Bayesian Optimization**: Bayesian optimization is an iterative optimization technique that models the objective function (model performance) as a probabilistic surrogate function. It uses this surrogate function to decide which set of hyperparameters to evaluate next, aiming to find the optimal set with as few evaluations as possible.
5. **Automated Hyperparameter Tuning Tools**: There are also automated hyperparameter tuning tools and libraries, such as Hyperopt, Optuna, and scikit-optimize, that use advanced optimization algorithms to efficiently search the hyperparameter space and find the optimal set of values.

Hyperparameter tuning is an important step in the machine learning workflow and can significantly improve the performance of models, leading to better predictive accuracy and generalization to unseen data.

here this is a large data set with 22,000 observations and for large data sets ideally doing non-linear regressions like random forest and ANN is more preferable since data loses its linearity as the observations increase.