ML Project Report: Hotel Property Value

# Team Members

1. Jayant Sharma - IMT2023523
2. Akshat Betala - IMT2023019
3. Avancha Dedeepya - IMT2023006

Github Link : <https://github.com/Akshat089/ML-Project-1>

# Task

Training ML models to predict the market value of hotel properties is essentially a regression problem, where the target variable is the HotelValue. Several factors, as mentioned below, contribute to predicting the final worth of a property—ranging from its physical characteristics and location to its amenities and overall condition—making this task a matter of estimating a continuous numerical value representing the property's market price in dollars.

# Dataset and Features Description

Predicting property values is a cornerstone of the real estate and hospitality industries. The ability to accurately estimate the market value of a hotel is crucial for investment, development, and financing decisions. This dataset provides a comprehensive collection of attributes for hotel properties, allowing for the development of machine learning models to tackle this valuation problem.

The dataset contains 1460 rows, each representing a unique hotel property, and 79 columns (features), including the target variable. The primary objective is to predict the HotelValue based on a wide array of factors, including location, size, construction details, quality, amenities, and recent sale information.

## Data Fields

Here is a description of the features found in the dataset:

* HotelValue – The market value of the hotel property in dollars. (Target variable)
* PropertyClass – Classification of the hotel property (e.g., standard, luxury, boutique).
* ZoningCategory – Zoning designation for the property’s location.
* RoadAccessLength – Length of road access available to the property (in feet).
* LandArea – Total land area of the property (in square feet).
* RoadType – Type of road providing access to the property.
* ServiceLaneType – Type of service lane available (if any).
* PlotShape – General shape of the property plot.
* LandElevation – Elevation/flatness of the property site.
* UtilityAccess – Types of utilities available to the property.
* PlotConfiguration – Configuration of the property plot.
* LandSlope – Slope of the land on which the property is built.
* District – Location of the property within the city/district.
* NearbyTransport1 – Proximity to nearby transport (main road/railway/airport).
* NearbyTransport2 – Proximity to a second transport facility (if any).
* PropertyType – Type of hotel property (e.g., resort, motel, serviced apartment).
* HotelStyle – Style/design of the hotel (e.g., multi-story, villa-type).
* OverallQuality – Overall quality of materials and finishing.
* OverallCondition – General condition of the property.
* ConstructionYear – Year when the property was originally constructed.
* RenovationYear – Year when the property was last renovated.
* RoofDesign – Style/design of the property’s roof.
* RoofMaterial – Material used for the roof.
* ExteriorPrimary – Primary exterior material of the hotel.
* ExteriorSecondary – Secondary exterior material (if applicable).
* FacadeType – Type of façade finish.
* FacadeArea – Area covered by façade material (sq. feet).
* ExteriorQuality – Quality of the exterior materials.
* ExteriorCondition – Current condition of the exterior.
* FoundationType – Type of foundation used in the property.
* BasementHeight – Height/ceiling quality of the basement.
* BasementCondition – General condition of the basement.
* BasementExposure – Basement exposure (e.g., walkout/garden).
* BasementFacilityType1 – Primary basement facility type (e.g., storage, staff area).
* BasementFacilitySF1 – Size of primary basement facility (sq. feet).
* BasementFacilityType2 – Secondary basement facility type.
* BasementFacilitySF2 – Size of secondary basement facility (sq. feet).
* BasementUnfinishedSF – Area of unfinished basement space.
* BasementTotalSF – Total basement area (sq. feet).
* HeatingType – Type of heating system installed.
* HeatingQuality – Quality and condition of the heating system.
* CentralAC – Whether central air conditioning is available.
* ElectricalSystem – Electrical system type.
* GroundFloorArea – Ground floor area (sq. feet).
* UpperFloorArea – Upper floor area (sq. feet).
* LowQualityArea – Low-quality finished area (sq. feet).
* UsableArea – Total usable area above ground (sq. feet).
* BasementFullBaths – Number of full bathrooms in the basement.
* BasementHalfBaths – Number of half bathrooms in the basement.
* FullBaths – Number of full bathrooms above ground.
* HalfBaths – Number of half bathrooms above ground.
* GuestRooms – Number of guest rooms (excluding staff/utility rooms).
* Kitchens – Number of kitchens.
* KitchenQuality – Quality of the kitchens.
* TotalRooms – Total number of rooms above ground (excluding bathrooms).
* PropertyFunctionality – Overall functionality rating of the property.
* Lounges – Number of lounges in the property.
* LoungeQuality – Quality of the lounges.
* ParkingType – Type of parking facility.
* ParkingConstructionYear – Year when parking was built.
* ParkingFinish – Interior finishing of the parking area.
* ParkingCapacity – Parking capacity (in number of cars).
* ParkingArea – Size of the parking area (sq. feet).
* ParkingQuality – Quality of the parking facility.
* ParkingCondition – Current condition of the parking facility.
* DrivewayType – Driveway type (paved/unpaved).
* TerraceArea – Terrace/deck area (sq. feet).
* OpenVerandaArea – Open veranda area (sq. feet).
* EnclosedVerandaArea – Enclosed veranda area (sq. feet).
* SeasonalPorchArea – Seasonal porch area (sq. feet).
* ScreenPorchArea – Screen porch area (sq. feet).
* SwimmingPoolArea – Pool area (sq. feet).
* PoolQuality – Pool quality rating.
* BoundaryFence – Type/quality of fencing.
* ExtraFacility – Additional facility not covered in other categories.
* ExtraFacilityValue – Value of the additional facility (in $).
* MonthSold – Month of sale.
* YearSold – Year of sale.
* DealType – Type of sale/deal.
* DealCondition – Condition under which the sale took place.

# EDA and pre-processing

The following section outlines the necessary steps for conducting essential Exploratory Data Analysis (EDA) to gain insights, detect potential issues, and prepare data for model training.

## 1. Import Libraries and Load Dataset

The initial step in the process involves importing essential libraries such as pandas, numpy, matplotlib, seaborn, and relevant modules from scikit-learn for data preprocessing and modeling. The train.csv and test.csv datasets are loaded into pandas DataFrames, and their shapes are printed to confirm the initial dimensions.

## 2. Data Overview

The dataset's basic structure is examined by loading the training and test sets and printing their shapes. This initial check confirms the number of entries and features available and verifies the data has been loaded correctly

## 3. EDA & Target Variable Analysis

A key step in the EDA process is analyzing the target variable, HotelValue. The script calculates the skewness of this variable and identifies that it is highly right-skewed. To normalize this distribution, which helps improve the performance of many regression models, a log transformation (numpy.log1p) is applied. The skew of the log-transformed target is found to be significantly lower, making it a more suitable target for model training.

## 4. Handling Missing Values

The dataset contains a significant number of missing values, which are handled strategically based on the feature's context:

* Numerical Imputation:
  + Features where NaN (Not a Number) likely means 'zero' or 'non-existent' (e.g., FacadeArea, BasementTotalSF, SwimmingPoolArea, ParkingArea) are filled with 0.
  + RoadAccessLength is imputed using the median() value of the column.
* Categorical Imputation:
  + Features where NaN represents a distinct category (e.g., 'No ServiceLane', 'No Pool', 'No Basement') are filled with the string 'None'.
  + A few other categorical columns (e.g., ZoningCategory, ElectricalSystem) are imputed using their statistical mode().

## 5. Handling Duplicate Values

While handling duplicates is a common preprocessing step, this specific pipeline does not include an explicit step for dropping duplicate rows. The focus is instead directed toward comprehensive feature engineering and missing value imputation.

## 6. Feature Engineering

To capture more complex relationships and provide richer information to the model, several new features are engineered from existing columns:

* PropertyAge: YearSold - ConstructionYear
* AgeSinceRemodel: YearSold - RenovationYear
* TotalSF: The sum of BasementTotalSF, GroundFloorArea, and UpperFloorArea.
* TotalBaths: A weighted sum of all full and half bathrooms.
* TotalPorchSF: The sum of all porch and veranda-related area columns.
* HasPool: A binary flag (1/0) indicating if SwimmingPoolArea is greater than 0.
* OverallQuality\_Cond: An interaction feature created by multiplying OverallQuality and OverallCondition.

## 7. Categorical Feature Encoding

To prepare categorical data for the machine learning model, a two-part encoding strategy is used:

* Ordinal Mapping: Features with an inherent order (e.g., ExteriorQuality, BasementCondition, PoolQuality) are manually mapped to numerical values based on their rank (e.g., 'None': 0, 'Po': 1, 'Fa': 2, ... 'Ex': 5).
* One-Hot Encoding: After all other preprocessing, pandas.get\_dummies() is applied to the combined dataset. This converts all remaining nominal categorical features (those without an inherent order) into binary columns, making the data suitable for the GradientBoostingRegressor.

## 8. Exploratory Data Analysis (EDA)

Various plots were generated to analyze the data distribution, detect missing values, and explore relationships among features. These visualizations provide critical insights that inform subsequent preprocessing and modeling decisions. Key visualizations and their interpretations are detailed below:

### Target Variable Analysis (Histograms)

The primary focus of the initial analysis was the target variable, HotelValue.

* Original Distribution: A histogram and kernel density estimate (KDE) plot revealed that the original distribution of HotelValue is highly right-skewed. This is confirmed by a high skewness value. This skewness can violate the assumptions of many linear models, potentially harming performance.
* Log-Transformed Distribution: To correct this, a logarithmic transformation (np.log1p) was applied to the target variable. The histogram of the transformed data shows a much more symmetrical, normal-like distribution. This normalized target is far more suitable for model training.

Conclusion: All subsequent analysis and model training will use the log-transformed HotelValue to ensure a more stable and accurate regression model.

### Missing Data Analysis

A crucial step was to identify the extent and pattern of missing data across the dataset.

* Missing Value Heatmap: A heatmap of null values was generated to visualize the entire dataset. This plot clearly indicated that a number of features have a significant percentage of missing data.
* Top Missing Features: The features with the most missing values were identified, including PoolQuality, ExtraFacility, ServiceLaneType, and BoundaryFence.
* Interpretation: This pattern strongly suggests that the missing values are not random errors but are Missing At Random (MAR), where NaN simply means the property *lacks* that feature (e.g., no pool, no fence). This insight is critical for the preprocessing step, where these NaN values will be strategically imputed (e.g., with 'None' or 0) rather than dropped.

### Correlation Matrix and Feature Analysis

A correlation matrix heatmap was generated for all numerical features to understand the relationships between variables, especially their relationship with HotelValue.

* Top 10 Correlations: To get a clearer view, the top 10 features most positively correlated with HotelValue were isolated and plotted on a bar chart.
* Key Findings:
  + OverallQuality: This feature shows the strongest positive correlation with HotelValue, confirming the intuitive idea that higher-quality materials and finishing lead to a higher market value.
  + Size-Related Features: UsableArea, GroundFloorArea, BasementTotalSF, and TotalBaths all show very strong positive correlations. This indicates that, as expected, the total size and usable space of the hotel are major drivers of its value.
  + Amenity Features: Features like ParkingArea and FullBaths also show significant positive correlations, highlighting their importance in property valuation.
* Scatterplot Confirmation: For each of these top 10 features, scatterplots were generated against HotelValue. These plots visually confirmed the strong, positive linear relationships and helped spot any potential non-linear patterns or outliers. For example, the scatterplot for UsableArea vs. HotelValue clearly shows that as area increases, the value trends upward.

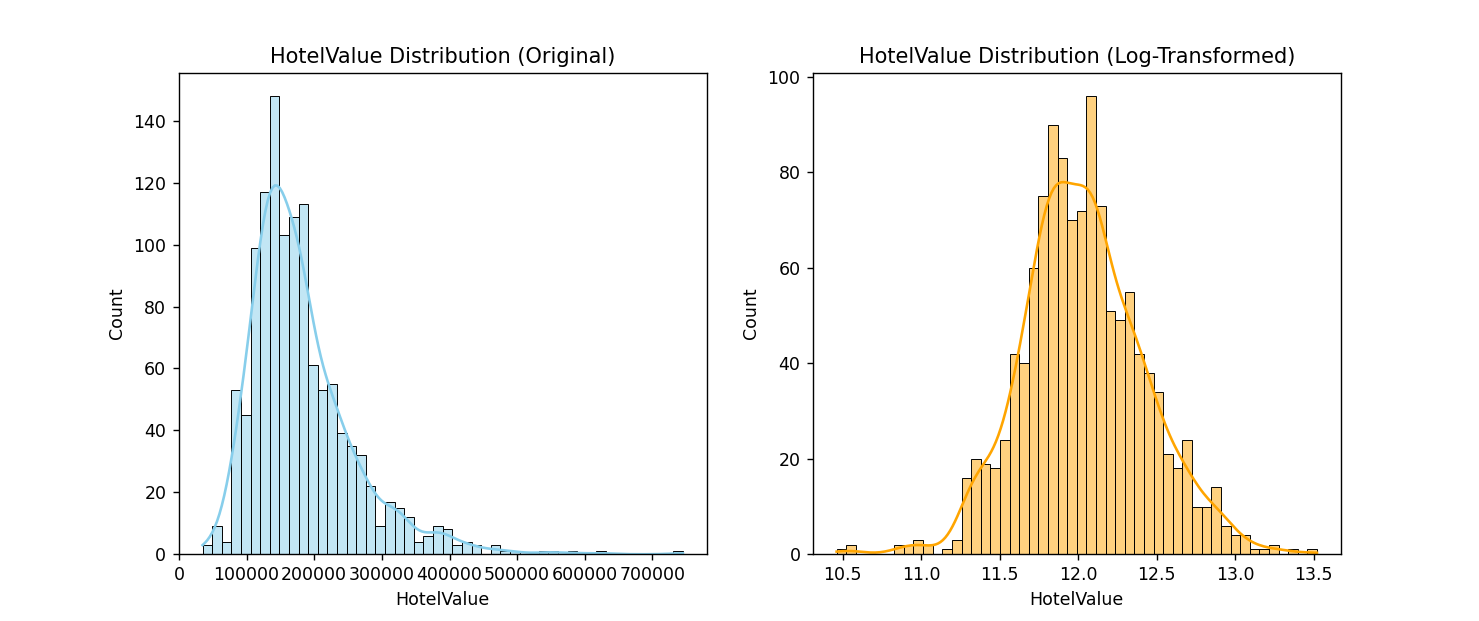
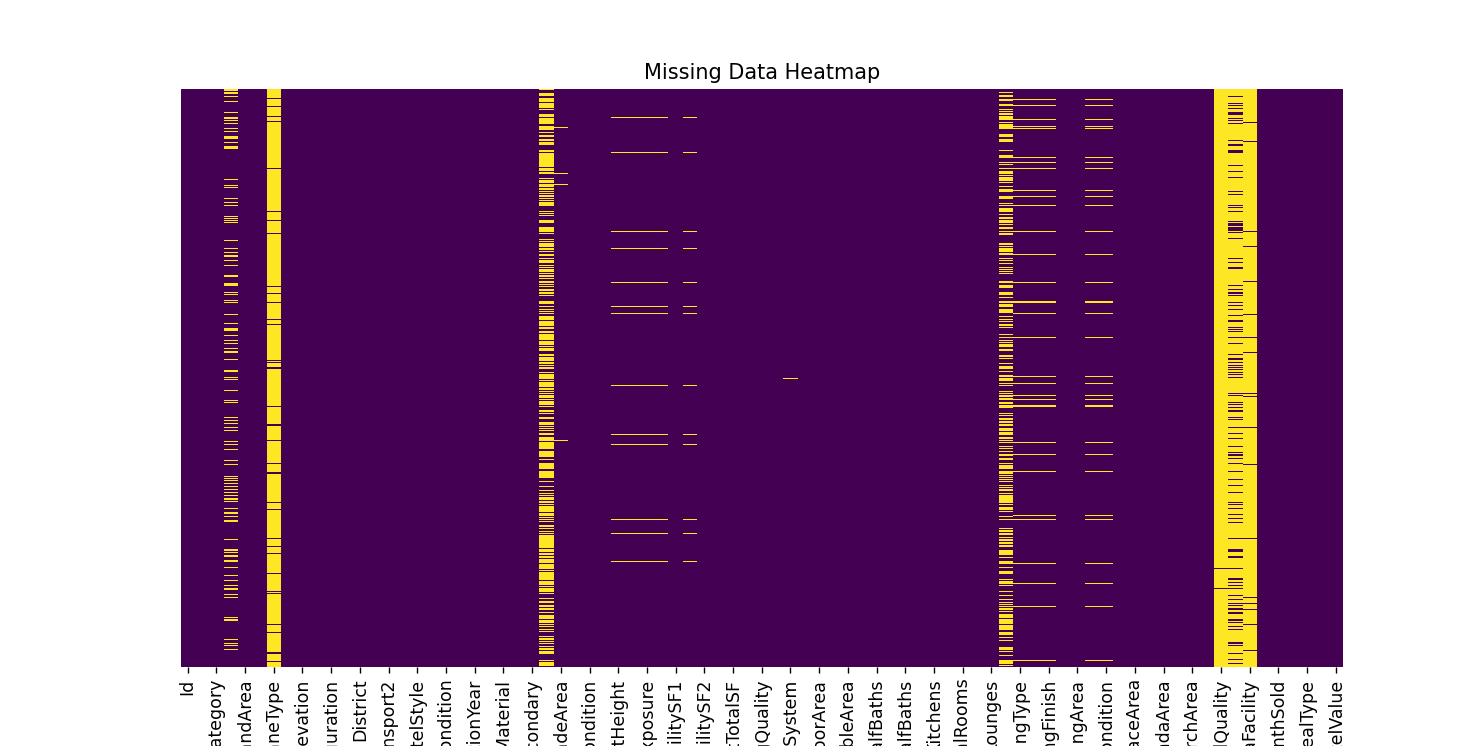
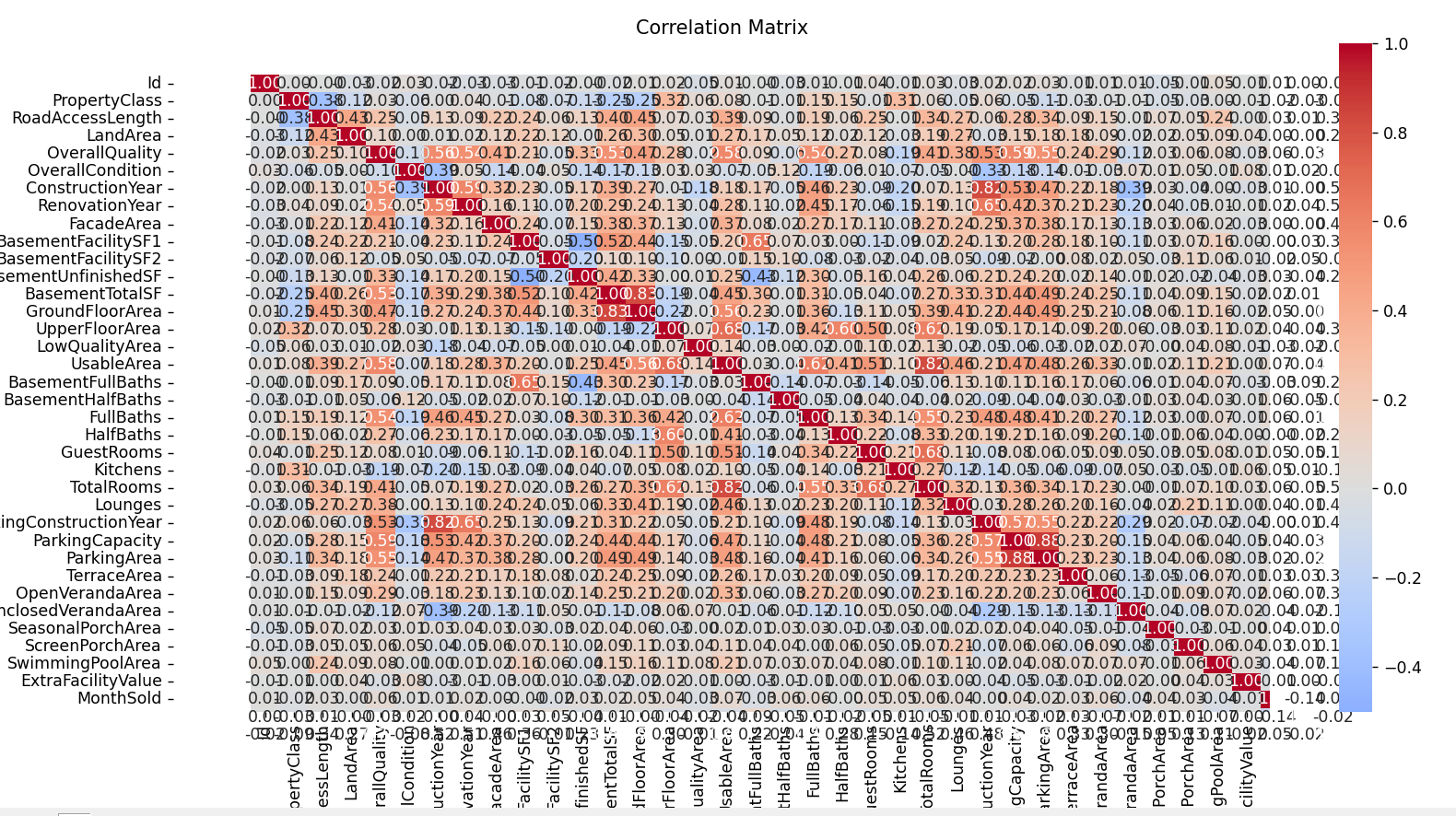
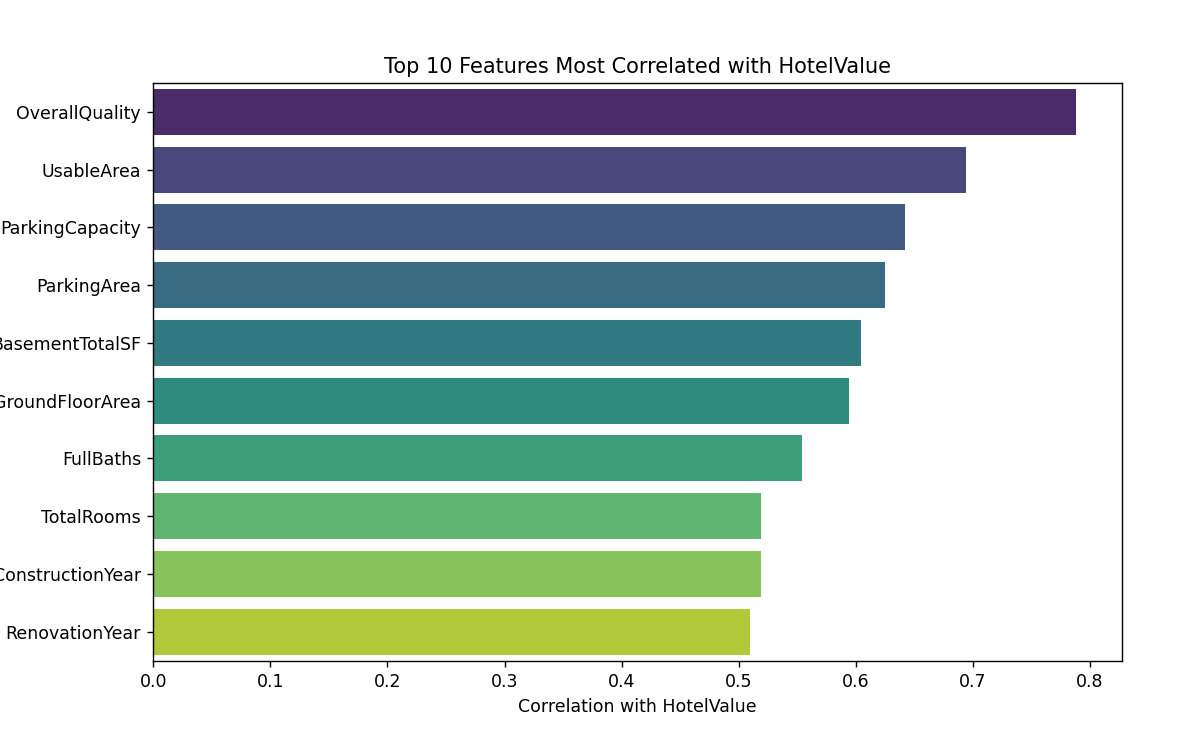
### Overall Conclusion

The EDA confirms the target variable needs transformation and that missing data is systematic. The correlation analysis identifies a set of powerful, common-sense predictors. Features like OverallQuality, UsableArea, and TotalBaths are the strongest individual predictors and will be central to the model's success.

## 9. Model Training and Validation

Instead of a single 80/20 train-test split, a more robust validation strategy was used:

* Data Splitting: The preprocessed all\_data a-frame was split back into the training set (X) and the final, unseen test set (X\_test). The log-transformed HotelValue was used as the target (y\_train).
* K-Fold Cross-Validation: K-Fold Cross-Validation (with N\_SPLITS = 5/10) was implemented for model training and validation. The training data was divided into 5 "folds." The model was trained on 4 folds and validated on the held-out fold, and this process was repeated. This approach provides a more reliable "Out-of-Fold" (OOF) RMSE score, giving a better estimate of how the model will perform on new data than a single train-test split.



Here are some of the models which we used:

* 1. Knearest neighbours
     1. For the KNN model, we followed the same preprocessing steps as outlined in the preprocessing section.
     2. Here we used 5 neighbours as n and took distances as the weights.
     3. This gave us a Kaggle score of 37229.497 which indicates that there are a lot of unnecessary features and a lot of skewed data.
  2. Random Forest Classifier
     1. Used same preprocessing as outlined in the pre processing stage.
     2. Here we just used Pipeline from sklearn
     3. Gave us a Kaggle Score of 35221.354 which is a very high score which means there is overfitting over here.
     4. Used the following Parameters  
        n\_estimators=300,
     5. max\_depth=12,
     6. min\_samples\_split=5,
     7. min\_samples\_leaf=2,
     8. random\_state=42,
     9. n\_jobs=-1
  3. Bayesian Approach
     1. This method introduces a **probabilistic framework** where the model assumes a prior distribution over the coefficients and updates it based on observed data to obtain a posterior distribution
     2. The model was trained through a pipeline including **median imputation**, **standard scaling**, and **one-hot encoding** for categorical variables. By incorporating Bayesian inference, it helps **reduce overfitting** on complex datasets and provides **uncertainty estimates** for coefficients.
     3. The Kaggle Score for this was 19417 and we used 50 iterations.
     4. This means that it performed very well just a little bit poorer compared to linear models.
  4. Linear Regression
     1. The pre-processing remains the same, which we have mentioned in the preprocessing section, involving checking for null values, removing duplicates, removing outlier points, and encoding the categorical columns.
     2. We also applied L1 and L2 regularization for Linear Regression. Since the no of columns was too much we didn’t try too much Polynomial Regression.
     3. Linear regression gave the best result of 17905. Without any L1 or L2 this was the best model which indicates simple data without extra noise.
     4. We also applied log target which makes the distribution a lot better.
     5. Using L1 we got around 20000 for different values of Alpha and for Ridge we got a score of 18156.265 for alpha as 0.001.
  5. Gradient Boosting
     1. Using Gradient Boosting with same pre processing we got 26000 roughly using the following parameters

n\_estimators=300,

learning\_rate=0.05,

max\_depth=4,

min\_samples\_split=3,

min\_samples\_leaf=2,

subsample=0.9,

random\_state=42

* + 1. Shows the dataset’s **optimal complexity level lies between fully linear and fully nonlinear**.
  1. LightGBM
     1. Gave a score of 32394.1. Shows its not anywhere ideal
     2. The **dataset size or feature structure** may not be large or complex enough for LightGBM to leverage its gradient-based tree boosting strengths.
  2. XGBoost
     1. Used these parameters with the same preprocessor as outlined before.  
        n\_estimators=1000,
     2. learning\_rate=0.05,
     3. max\_depth=6,
     4. subsample=0.8,
     5. colsample\_bytree=0.8,
     6. random\_state=RANDOM\_SEED,
     7. tree\_method='hist', # faster
     8. eval\_metric='rmse'
     9. Here the score was around 40000 which indicates the data was too complex for XGboost and not the ideal usage.