Name: Shashwat Dhayade

Project Name: Predict Housing Prices

# Introduction

The housing dataset available from Kaggle is used for data mining and machine learning, which contains information about 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa. This Kaggle project challenges you to predict the final price of each home. This report intends to answer the problem of underfitting and overfitting models, including data preparation, exploratory data analysis, modeling, and evaluation. The goal of this Kaggle project is to accurately predict the final price of residential homes in Ames, Iowa using the available housing dataset.

# Dataset

The Housing dataset is available in several formats, including CSV, GZ, and TXT, and can be downloaded from Kaggle.

train.csv - the training set

test.csv - the test set

data\_description.txt - full description of each column, originally prepared by Dean De Cock but lightly edited to match the column names used here. It contains 79 explanatory variables describing (almost) every aspect of residential homes

sample\_submission.csv - a benchmark submission from a linear regression on year and month of sale, lot square footage, and number of bedrooms

# Data Preparation

Before creating a model, it is essential to prepare the data correctly. This involves handling missing values, transforming categorical variables to numerical ones, scaling numeric features, and identifying outliers. First, I normalized the SalePrice variable by taking the natural log to remove any skewness. Then, I handled missing values by either imputing them using appropriate strategies or dropping them if they were more than a certain threshold. For example, if the data is skewed I used the median to fill the null values and for categorical variables, I used the most used value for that feature. I did this using exploratory data analysis. Outliers were handled using scatter and boxplots or some features. This reduced the skewness of those features. After handling missing values, I transformed categorical variables into numerical ones by assigning appropriate values based on their meaning or frequency in the dataset. Overall, careful data preparation is crucial for building accurate and robust models.

# Exploratory Data Analysis

Exploratory data analysis (EDA) is essential in understanding the characteristics of the dataset and identifying patterns and relationships between variables. In this project, EDA was conducted to gain insights into the distribution of variables and their correlations with the target variable, SalePrice. Various visualizations such as histograms, scatter plots, and correlation matrices were used to analyze the data. Through EDA, I found that some variables have high correlations and may cause multicollinearity issues in the model. Additionally, some variables had low correlations with the target variable and were not useful in predicting house prices. These models were also used for data preparation processes such as finding outliers, NAN values, inconsistencies, skewness, etc. Furthermore, feature extraction was done based on certain graphs because it helped understand the relationship between various features. For example, creating features like the mean of monthly SalePrice from the number of sales per month and SalePrice. Overall, EDA was executed successfully for understanding the model.

# Feature Engineering

48 RemodYears 2426 non-null int64

49 HasRemodeled 2426 non-null int64

50 HasRecentRemodel 2426 non-null int64

51 GarageBltYears 2426 non-null float64

52 Now\_YearBuilt 2426 non-null int64

53 Now\_YearRemodAdd 2426 non-null int64

54 Now\_GarageYrBlt 2426 non-null float64

55 MonthSaledMeanPrice 2426 non-null float64

56 MonthSaledCount 2426 non-null int64

57 MSSubClassMeanPrice 2426 non-null float64

58 NeighborPrice 2426 non-null float64

59 NeighborBin 2426 non-null int64

60 IsRegularLotShape 2426 non-null int64

61 IsLandContourLvl 2426 non-null int64

62 IsLotConfigInside 2426 non-null int64

63 IsLandSlopeGentle 2426 non-null int64

64 IsCondition1Norm 2426 non-null int64

65 IsCondition2Norm 2426 non-null int64

66 IsBldgType1Fam 2426 non-null int64

67 IsRoofStyleGable 2426 non-null int64

68 IsRoofMatlCompShg 2426 non-null int64

69 IsGasAHeating 2426 non-null int64

70 IsGarageFinished 2426 non-null int64

71 IsPavedDrive 2426 non-null int64

72 IsSaleTypeWD 2426 non-null int64

73 IsSaleConditionNormal 2426 non-null int64

74 IsVeryNewHouse 2426 non-null int64

75 Has2ndFloor 2426 non-null int64

76 HasMasVnr 2426 non-null int64

77 HasWoodDeck 2426 non-null int64

78 HasOpenPorch 2426 non-null int64

79 HasEnclosedPorch 2426 non-null int64

80 Has3SsnPorch 2426 non-null int64

81 HasScreenPorch 2426 non-null int64

82 SimplOverallQual 2426 non-null int64

83 SimplOverallCond 2426 non-null int64

84 OverallGrade 2426 non-null int64

85 KitchenScore 2426 non-null int64

86 FireplaceScore 2426 non-null int64

87 GarageScore 2426 non-null float64

88 TotalBath 2426 non-null float64

89 TotalPorchSF 2426 non-null int64

90 AllSF 2426 non-null float64

91 BoughtOffPlan 2426 non-null int64

92 Isgarage 2426 non-null int64

93 Isfireplace 2426 non-null int64

94 Ispool 2426 non-null int64

95 Issecondfloor 2426 non-null int64

96 IsOpenPorch 2426 non-null int64

97 IsWoodDeck 2426 non-null int64

98 TotalSqrtFeet 2426 non-null float64

99 TotalBaths 2426 non-null float64

Above mentioned are 52 features that were extracted from the dataset. These features were extracted so that the categorical features could be dropped from the dataset for better modeling. These features were extracted based on the categorical features, in order to get a numerical understanding of the dataset. Thus, I dropped 32 categorical variables from the dataset and added these features. Thus, I have 100 features in total and 2426 rows.

# Model Selection and Evaluation

The dataset was split into training (956), test (1231), and validation sets (239). I used the following models –

Linear Regression –

The goal of the competition is to predict the sale price of a house, which is a continuous variable, based on a set of features that include both continuous and categorical variables. Linear regression models can capture linear relationships between the predictors and the response variable, which makes them a good choice for this task. Additionally, the coefficients in a linear regression model have a clear interpretation, which can help in understanding the impact of each predictor on the response variable.

Gradient Boosting Regressor –

There are many variables that may have non-linear relationships with the sale price of a house. For example, the age of a house may have a non-linear impact on its sale price, with older houses being valued differently than newer houses. Additionally, there may be interactions between variables that cannot be captured by a simple linear regression model. GBR can capture these complex relationships by building a series of decision trees, where each tree learns to predict the residual error of the previous tree.

XGBRegressor - One of the main advantages of the XGBRegressor algorithm is its speed and scalability. The algorithm is designed to handle large datasets with high dimensionality, which makes it well-suited for the dataset.

The metric used is [Root-Mean-Squared-Error (RMSE)](https://en.wikipedia.org/wiki/Root-mean-square_deviation) between the predicted value and the observed sales price to determine how well the model performs. The RMSE value for the models used are –

Linear Regression – 0.124

Gradient Boosting Regressor – 0.121

XGBRegressor – 0.125

Evaluation –

Linear Regression –

The model and performance plots show that the predicted values and the true values are close to the RMSE of 0.124. The residual plot shows randomness around 0, and it has some values which are not highly concentrated around the origin. The learning curve tells us that the model is overfitting because of its simple understanding of the data. It cannot capture the complexity.

Gradient Boosting Regressor –

The learning curve and RMSE show that the training score is higher than the cross-validation score. This shows that the model might be too complex and fitting noise in training data which makes it overfitting.

XGBRegressor –

The residual plot and learning curve shows that the model might be overfitting. Because the training error is lower than the cross-validation error. But the difference between them is extremely low. This might show that the model might be better than the gradient-boosting regressor. But overall considering the RMSE value and the plots XGBRegressor model is the best model.

# Results and Implications

Overall, the project was successful because the RMSE value shows that the XGB model is a good predictor. The challenges I faced were how to incorporate the categorical variables in the dataset as numerical features. But, in the future Data Understanding will be something I will pay more attention to because the rest of the project depends on it and it helps in saving time rather than fixing upcoming unprecedented errors.

The overfitting and underfitting models were handled using an appropriate number of features. The complexity of the model was kept optimum by trying different hyperparameters. The split between the train and validation set was optimized to get better predictions.

This project can be used in the future for projects involving construction management, and various applications of industrial engineering.