Housing Price: Advanced Regression Techniques:

Syed Mustafa and Yousaf Khan

**Abstract:**

This project is intended to use machine learning algorithms to detect housing prices based on various characteristics of the house. This document focuses on the pre-processing steps applied to the acquired dataset before the machine learning models are applied to it.

**Preprocessing the Data Set:**

We start by finding the correlation between SalesPrice (our dependent variable) and all our numerical input variables for the model (the model has variables with String fields too which we will come to later in the project via discretization). We drop all attributes that have a correlation between -0.2 and 0.3, identifying this range to be weak enough so as not to be a good predictor for SalesPrice. In particular, we drop the﻿ LotArea, OverallCond, BsmtUnfSF, BsmtFullBath, BsmtHalfBath, EnclosedPorch, YrSold attributes. This can be confirmed in the results below:

A picture containing text, plaque

Description automatically generated

Next, we further drop all attributes with mostly NA data values, including ﻿PoolQC, Fence, Alley, MiscVal. We also drop the ﻿RoofMatl attribute which contains mostly the value ‘CompShg’. This is because these attributes are not useful in predicting the SalesPrice.

We also drop attributes which are dominated by 0 data values as they are not helpful for us to make good predictions. These attributes are ﻿EnclosedPorch, 3SsnPorch, ScreenPorch, and PoolArea. Notice that EnclosedPorch had already been dropped in a previous step.

Once the above attributes are dropped, we proceed to standardize our remaining numerical attributes that have no NA values - ﻿OverallQual, YearBuilt, YearRemodAdd, 2ndFlrSF, 1stFlrSF, GrLivArea, Fireplaces, WoodDeckSF, OpenPorchSF.

**Results**

We use a linear regression model and a gradient boosting algorithm to analyze the accuracy of our hypothesis values on both the training and testing set. The linear regression model gives us an accuracy of 0.76 while the later gives us a score of 0.86, both for the testing set. When applied to the testing set (on Kaggle), our algorithms give us an accuracy of 0.45(to be improved). Without standardization, we obtain a prediction accuracy of 0.59 on the training set. This demonstrates that our model has improvement after standardizing specific attributes as a score closer to 0 is considered better. The goal is to reach 85-90% accuracy of predictions through our models.