House Prices: Advanced Regression Techniques

Domain Background

The real estate domain encompasses buying, selling and renting of property and houses. Whenever we buy or sell a house, we often need a real estate agent to quote a price for the same since they know which factors are important and how much they affect the house's price, in other words domain knowledge. This project aims to build a model that can help the estate agent explain and the consumer understand the factors or features that affect house prices the most, and also the price itself.

The paper Modeling Home Prices Using Realtor Data is an example of predicting house prices using 12 features extracted from realtor data. In this project we try to solve a similar ongoing kaggle competition, but with a much larger feature set. The motivation of this project is feature engineering.

Problem Statement

In this project our goal is to build a regression model to predict sale prices of houses. For each Id in the test set we must predict the value of the SalePrice variable.

Datasets and Inputs

The Ames Housing dataset is a modernized and expanded version of the often cited Boston Housing dataset. This dataset contains **79** explanatory variables describing almost every aspect of residential homes in Ames, Iowa, including but not limited to, type of dwelling, total area of basement in square feet, type of road access, neighborhood etc. The dataset contains 1460 training points and 1459 testing points.

Upon slight analysis we can see that there are **33 numerical** features and **46 categorical** features. Among the numerical features we can see both **discrete** and **continuous** features. Among the categorical features we can see both **nominal** and **ordinal** features. The target variable SalePrice is the one that needs to be predicted.

Solution Statement

The solution takes a form of a mathematical function or regression model that trains over all the features or a subset of features across all the training points including the target variable. Once all the training points are fit, after minimizing the loss function, the model should be able to predict the target variable when a test point is fed into it.

Benchmark Model

The Data Sources on the competition page contains a file sample_submission.csv which contains values of the target variable for all the points in the testing set. This model is learned by a linear regression on year and month of sale, lot square footage and number of bedrooms.

Evaluation Metrics

Models are evaluated on Root-Mean-Squared-Error (RMSE) between the logarithm of the predicted value and the logarithm of the observed sales price as specified here.

Project Design

The problem, as is, suffers from curse of dimensionality. We have 79 features and 1460 training points. Since our training data is fixed we should employ feature engineering to reduce the number of features. The overall workflow can thus be divided into four major parts.

- **Preprocessing**

  Our objective here is to make the dataset fit for use in a regression model. We should employ, but not limited to, the following:

  - Handle missing data. Missing data can be filled with mean/median/mode or we can use domain knowledge to infer those values.
  - Remove skewness. A logarithmic transformation does the job.
  - Outlier analysis and removal of the same. IQR or interquartile range should be able to detect houses with unusually high or low price.

- **Feature reduction**

  Our objective here is to minimize the number of features while incurring minimal loss in information. Few ways to do that, but not limited to, are as follows:

  - Remove redundant features, example GarageCars, GarageArea. Both are a measure of size of garage. We can choose to keep GarageArea.
  - Remove unimportant features, example GarageYrBlt. Domain knowledge is necessary for this.
  - Remove highly correlated features. If some features are highly correlated we can keep one and remove the rest.
  - Remove low variance features.

  Empirical analysis of multiple regression models

  Our objective here is to analyze multiple regression models based on the evaluation metric. Based on the type of data choose a maximum of three regression models from below and evaluate them.

  - A LinearRegression model is a tempting choice because of Occam's razor and with better features should be able to surpass the benchmark model.
  - As suggested in the competition page a GradientBoostingRegressor model should be applied for it's improved accuracy.
  - XGBoost and LightGBM can also be applied for it's improved performance.

- **Choosing the best model and improving the results**

  Our objective here is to tune the hyper-parameters of the chosen model and improve the results by having an optimal cross validation score.

  - GridSearchCV allows a model to be tested with multiple hyper-parameter's values and returns the best parameter's value.
  - Common hyper-parameters for boosting algorithms are n_estimator and learning_rate.
  - Parameter tuning for XGBoost can be found here where as for LightGBM can be found here.