Housing Price - Advanced Regression Techniques Project

Aryan Gurubacharya, Aaryan Jha

509 South Locust Street, DePauw University
Greencastle, IN 46135, U.S.A.
agurubacharya_2024@depauw.edu
aaryanjha_2023@depauw.edu

Abstract

This paper highlights the steps and process carried out to predict the housing price in Ames, Iowa. We are given a dataset that contains 79 input attributes which describe various features of a house like its overall area, garage area, condition of the house, utilities, and other factors that affect the price of the house. Our goal is to come up with techniques that polish the data and transform it to make it more suitable for a particular type of model. The polishing of data is carried out during preprocessing. Preprocessing is a pivotal step in predicting the housing price as it deals with most of the data transformations like filling the missing values, removing outliers, and standardizing the data. For our project, we used a gradient boosting regressor algorithm as our model as it predicted the housing prices with the highest accuracy compared to other models.

Keywords: Gradient Boosting Regression, Data Pre-Processing, Model Selection

1 Introduction

The Housing Price Prediction Project is a challenge for machine learning enthusiasts to predict the sale prices of homes based on various features such as location, square footage, and number of bedrooms. In this project, we are asked to build a model that can accurately predict the sale prices of homes in the test set, based on the features provided in the training set. This is a regression problem, where the goal is to predict a continuous numeric value(the sale price of a home).

All over the world, predicting house prices in order to minimize any bad purchase is one of the most challenging problems we face everyday. There are several different approaches that can be used to determine the price of the house. We start by completely reading the datadescription.txt file to use all the attributes present in the data and how we can possibly use it.

The Data Analysis has been done using different python libraries such as the pandas, sklearn, matplot etc. The functions of these libraries are used to predict the attributes needed to make the absolute prediction for a house. The accuracy of our prediction is known by the accuracy mean calculated by each model we used.

2 Data Description

After reading the “datadescription.txt” file, we know the given data set and the message the attributes are trying to inform us. We are given 80 different attributes to predict the best price as each of them were correlated with the SalePrice. There were 37 numerical attributes and 43 categorical values. Reading through the data set will give us a brief idea about the correlation between each attribute present and how it might help us predict the price. This will lead to a crucial stage in our project.

3 Pre-Processing of Data

One of the most important processes before applying any kind of model to our dataset, is the data pre-processing. Since we have to consider our dataset to be raw and 'dirty', this process involves cleaning and preparing the data for analysis, and can have a significant impact on the results of the price prediction we make. The pre-processing method was done with the concept of trial and error and finding multiple combinations to produce best results. It helped us ensure the quality, relevance and suitability of the data we used.

First we needed to visualize our data. The initial print statements are being passed as arguments which helps read and understand the features our data presents. Upon that, we realized that our data had missing values (numerical and non-numerical), inconsistent data and outliers present in our dataset. Having these problems while building the model and making a prediction will cause our prediction price to be very inaccurate. So the problems were handled in these different ways:

3.1 Missing Values

To handle this problem, first we needed to see what exact attributes had missing values, their data type and how many fields were missing inside those attributes. We called the 'getAttrsWithMissingValues(df)' method to see the attributes with the missing values and how many missing values were present in the dataset. Knowing that, we first handled the non-numeric values present by calling the 'handlen_on_numerical_data(df)' method. This method iterates over every attribute in our dataframe and first checks if it is numeric or not. Each non-numeric attribute is then handled by converting each unique value inside the attribute with 0,1… and so on. This handles all our non-numeric attributes and gets it ready for use. After that, we handle the numeric values. To do this, we used the concept of K-nearest neighbor (KNN). It first calculates the distance between the data point with missing values with other data points in the dataset. We picked 5 numbers of data points that are closest to the original data point and computed the average into our missing values. After this, our dataset was completely filled out.

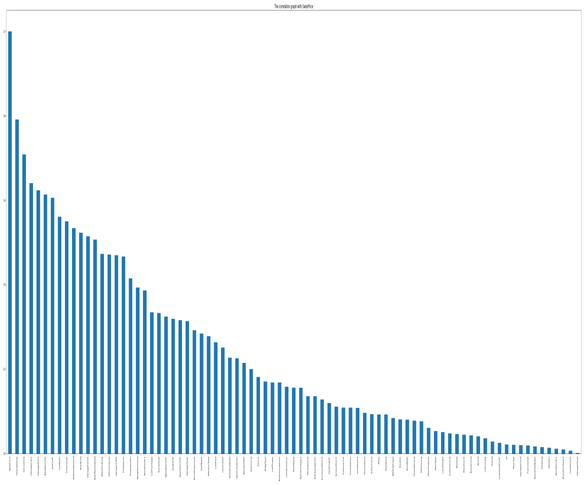

Figure 1: The correlation between SalePrice VS All the Attribute

3.2 Correlation

After our dataset is complete with values in all of its field, we find the correlation between the dataset. Correlation analysis is a method we use to investigate the relationship between any two variables. We have used correlation analysis to see the relationship between our target variable 'SalePrice' with other attributes. It gave us this result seen in Figure 1.

The Figure 1 graph is the visual representation of the strength and direction of the relationship between 'SalePrice' with other attributes in our dataframe. The height of the line of best fit indicates the strength of the correlation: a taller line indicates a stronger correlation, while a shorter line indicates a weaker correlation. The graph shows us all the attributes. But to get precise attributes to use and remove, we use the 'nlargest()' and 'nsmallest()' methods. These methods give us the 10 most correlated attributes and 10 weakest correlated attributes. This is a big tool to decide our predictors while designing our model.

3.3 Outliers

Since we now have our strongest related attributes, we need to make sure that there are no outliers in our data

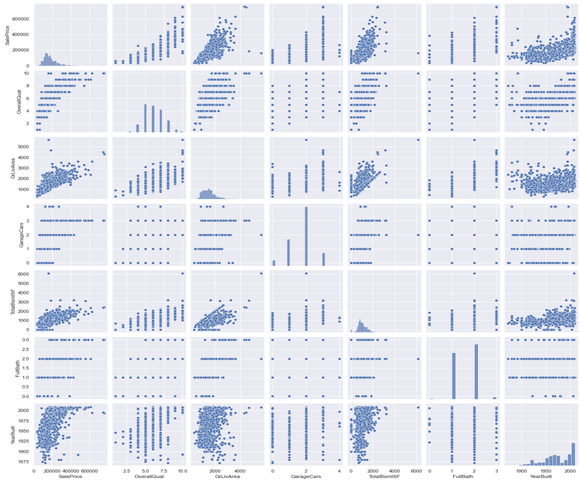

Figure 2: The correlation between SalePrice VS Closest Coorelation Attributes

that could potentially harm our results. To check for that, we created a graph Figure 2.

From the graph Figure 2, we can see that 'GrLivArea' and 'TotalBsmtSF' have outliers present. We handle the outliers by dropping them from our dataset. This makes our data ready for testing.

3.4 Standardization

Before our data goes for testing, we have to standardize our data. Standardizing our training and testing data sets is required to ensure that the data sets can be comparable. This makes the set draws from the same population and has a similar distribution of variables. Standardizing our data brings stability in our data and as it scales our data, thus our data is prepared for testing.

4 Algorithms

4.1 Gradient Boosting Regression

Gradient boosting regression is a machine learning technique that is used to predict continuous values. It involves training a sequence of models, where each model corrects the mistakes of the previous model. The final prediction is obtained by summing the predictions of all the models in the sequence. This method is an ensemble method, which means that it combines the predictions of multiple models to produce a more accurate and stable prediction. It is useful for handling complex datasets and improving the performance of weak models.

4.2 Linear Regression

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. The equation for a simple linear regression model with one independent variable is:

$$y = b_0 + b_1 * x$$

where y is the dependent variable, x is the independent variable, b_0 is the intercept, and b_1 is the slope of the line. The goal of linear regression is to find the values of b_0 and b_1 that best fit the data, so that the line can be used to make predictions about the dependent variable.

4.3 Linear Regression

Ridge regression is a method used in linear regression to address multicollinearity. Multicollinearity occurs when predictor variables in a linear regression model are highly correlated, which can cause unstable and unpredictable results. Ridge regression addresses this problem by adding a penalty term to the objective function of linear regression. This penalty term helps prevent overfitting and improves the stability and interpretability of the model. The strength of the penalty is determined by the regularization parameter, which can be adjusted to produce a more or less regularized model. The formula is:

$$\min_{w} ||y - Xw||^2 + \alpha||w||^2$$

5 Model Selection

The final part of our project is model selection. For this part, we started to experiment with various models like the linear model, Lasso model, and ridge model. These models follow a regression algorithm that is used to predict continuous values by fitting a linear equation to the data.

Table 1: Housing Price Prediction CV Mean Score

Models	CV Average Score
Linear Regression	0.8525387421529794
Ridge Regression	0.8527093026865247
Gradient Boosting Regression	0.8947292123101789

We found out that our accuracy did not vary much between these models, staying at around the range of 82%.

After extensive research, we found out about the ensemble models like gradient boosting, random forests, and bagging. Ensemble models combine the prediction of multiple models to make its final prediction. Among the models we tested, we found out that gradient boosting predicted the housing prices with the highest accuracy of 89%; therefore, we decided to use gradient boosting as our model.

6 Conclusion

This paper overall shows the model based on the average prediction of Linear, Ridge and Gradient Boosting Regression. The pre-processing method prepared our dataset and removed any negative attributes that would affect our prediction. The graph aids our visualization to select our attributes and at the end of our project, we have the CV score of 0.8925