Machine Learning and House Pricing - A Regression Problem

Nurudeen Adeshina Akintola (nurudeen.a.akintola@aims-senegal.org)

African Institute for Mathematical Sciences (AIMS)

Supervised by: Doctor Pierre-Yves Lablanche,
African Institute for Mathematical Sciences, South Africa.

Co-Supervised by: Doctor Amadou Lamine Toure,
African Institute for Mathematical Sciences, Senegal.

26 May 2017

Submitted in partial fulfillment of a structured masters degree at AIMS SENEGAL

Abstract

The purpose of this essay is to analyse a housing dataset using advanced machine learning algorithms. The dataset contains information from the AMES Assessor's Office used in computing assessed values for individual residential properties sold in Ames, Iowa, USA from 2006 to 2010.

During the essay, different regression techniques are applied to 2930 observations with 82 variables, to determine the performances of each techniques and present how powerful machine learning can be in the field of predictive analysis of house pricings.

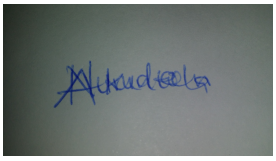
We used different algorithms namely: Random Forest, Gaussian Naive Bayesian, k-Nearest Neighbours and Ordinary Least Squares. It was found that RandomForest outperformed all other algorithms, followed by Ordinary Least Square, then k-Nearest Neighbours and finally Gaussian Naive Bayesian.

We also found out that five variables accounted to 90% of the housing prices.

Keywords: Predictive Modelling, Machine Learning, Regression techniques, Selection of variables, House price model

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.



Nurudeen Adeshina Akintola, 26 May 2017

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank the following people for their support in this project:

1. I owe the deepest gratitude to Doctor Pierre-Yves Lablanche, my supervisor, who from the very beginning of the idea to its realization has given me his substantive guidance and feedbacks. His encouragement, passion, tolerance, unlimited support and giving valuable feedbacks are really appreciated.
2. Dr Amadou Lamine Toure for advising me on the computational requirements of the project, and
3. My family and friends for their unwavering love and support

1. Introduction

Prices of real estate properties is critically linked with our economy [Shi07] . Regression is a non-trivial problem especially when you have a lot of features that are categorical or continuous. There is actually no standard model for this problem because of all features in consideration differs from one geographical location to another. AMES dataset has 2930 houses with 81 features, such as geographical location, living area, and number of rooms, and their Sale Prices. This rich dataset should be sufficient to establish a regression model to accurately predict the price of real estate properties in Ames, Iowa.

An accurate prediction on the house price is important for prospective homeowners, developers, investors, appraisers, tax assessors and other real estate market participants, such as, mortgage lenders and insurers [FJ03]. Traditional house price prediction is based on cost and sale price comparison, lacking of an accepted standard and a certification process. Therefore, the availability of a house price prediction model helps fill up an important information gap and improve the efficiency of the real estate market [Cal03].

The objective of this project is to predict the house prices so as to minimize the problems faced by the aforementioned professionals. The present method is that the customer approaches a real estate agent to manage his/her investments and suggest suitable estates for his investments. But this method is risky as the agent might predict wrong estates and thus leading to loss of the customer's investments. The manual method which is currently used in the market is out dated and has high risk. So as to overcome this fault, there is a need for an updated and automated system. Specific algorithms can be used to help investors choose an appropriate estate according to their mentioned requirements. Nonetheless, mathematical predictive models are not necessarily flexible enough to account for local specificities resulting in sometimes poor predictions.

This is where machine learning comes into play. Instead of using hard - coded parameters and static program instructions, the prediction system can learn from the dataset to teach itself and refine its parameters and make data-driven predictions. With a prediction model, it aims to assist property buyers in making predictions on future property prices by harnessing the power of the large dataset already available.

Our purpose here is to apply and optimise machine learning regression techniques to the house pricing problem, this include analysing and cleaning the dataset (pre-processing), selecting and optimising regression techniques and evaluate their performances

This work is organized in the following chapters. Chapter 1 discuss the introduction. Chapter 2 outlines previous study on regression and the house pricing problem. Chapter 3 introduces how we dealt with missing values and prepared the dataset for the machine learning algorithms. Chapter 4 discuss the implementation of the different algorithms and provides results and Chapter 5 derives conclusion from the analysis

2. House Pricing and Regression Problem

In this section, we introduce some literature reviews on works done on house pricing, the basic concepts of regression techniques along with machine learning algorithms used for regression.

2.1 House Pricing

A property's appraised value is important in many real estate transactions such as sales, loans, and its marketability, but estimations of house price can be quite tedious. Traditionally, estimates of property prices are often determined by professional appraisers. The disadvantage of this method is that the appraiser is likely to be biased due to vested interest from the lender, mortgage broker, buyer, or seller and hence house prices may differ from one appraiser to another. Therefore, an automated prediction system has the advantage to offer a standard estimation. For the buyers of real estate properties, an automated price prediction system can be useful to find under/overpriced properties currently on the market. This can be useful for first time buyers with relatively little experience, and suggest purchasing offer strategies for buying properties.

Home values are influenced by many factors. Basically, there are two major aspects:

- The environmental information, including location, local economy, school district, air quality, etc.(external factors)
- The intrinsic information of the property, such as lot size, house size , number of rooms, heating / AC systems, garage, and so on. (internal factors)

Studies on home prices have been going on for many years using various models. The traditional and standard model is the hedonic pricing model that infers that the price of dwellings is determined by the internal factors (characteristics of the property) as well as external attributes [LGW98]. The model considers various combinations of internal and external predictors for predictions [EF95]

Pashardes and Savva, [PPS+09], investigated the impact of external and internal variables on house prices in Cyprus from 1988 - 2008. They collected data from newspaper advertisements on various housing types such as semi-detached, number of bedrooms and size of building and geographical location. They selected population, cost of materials and labour in construction, the Euro exchange rate to English pound and unemployment rate as their external variables. They found that house prices are sensitive to population, cost of building materials and labour, GDP growth and the sterling-euro exchange rate.

Paterson and Boyle [PB02] used a hedonic pricing model to estimate the impact of different types of views on residential property values in Connecticut. Views were categorised into Development, Agriculture, Forested area or Water and percentage of area in each type of view visible within a kilometre were measured to differentiate views by quality. They found that the degree of visible Forested land and Development caused significantly lower property values, whereas visibility of Agriculture land did not appear to have a significant impact on property values.

Apergis and Rezitis, [AR03] used macroeconomic variables of money supply, employment, inflation, and housing loan rates. They found that housing loan rate caused the highest impact on housing price, followed by inflation and employment, while money supply does not seem to show any substantial impact.

Kestens, Theriult and Rosiers [KTDR06], used two sets of hedonic models with 761 single-property transactions carried out in Quebec City between 1993 and 2001. The variables used included living area, income of households, age profiles of buyers, house quality, in-ground pool and local tax rate. Each model can explain at least 84 % of the house price variation from the combinations of the variables without collinearity effects. Their main finding is that buyer's household income, the previous tenure status and age have a direct impact on property prices.

As for micro variables in hedonic regression models, many researches used property characteristics, building structure, tenure, neighbourhood characteristics, location and environment. The commonly used variables are number of bedrooms and bathrooms [FGM00]; detached or high rise buildings [Xu08]; age of building [CH00].

Recent studies have focused on price prediction performance comparison between hedonic-based methods and machine learning algorithms.

Fan et al. [FOK06], suggested various tree-based approaches that provide an important statistical pattern recognition tool in examining the relationship between house prices and housing characteristics.

Liu et al. [LZW06], proposed a fuzzy neural network prediction model based on hedonic price theory to estimate the appropriate price level for new real estate. The experimental results indicated that the fuzzy neural network prediction model had strong function approximation ability and was suitable for real estate price prediction.

Selim [Sel09], compared the prediction performance between the hedonic regression and artificial neural network models. This study demonstrated that artificial neural network models can be an improved alternative for prediction of house prices in Turkey.

2.2 Regression Model

Regression analysis is a statistical process for estimating the relationship among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. More specifically, regression analysis helps one understand how the typical value of the dependent variable changes when any one of the independent variables is changed, while the other independent variables are held fixed.

Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships.

Many techniques for carrying out regression analysis have been developed. Familiar methods such as linear regression and ordinary least squares regression are parametric, in that the regression function is defined in terms of a finite number of unknown parameters that are estimated from the data. Nonparametric regression refers to techniques that allow the regression function to lie in a specified set of functions, which may be infinite-dimensional.

2.2.1 The Multiple Linear Regression. The simplest regression model is linear regression, which involves using a linear combination of independent variables to estimate a continuous dependent variable [Bis06]. While the model is too simple to accurately model the complexity of Ames housing market, there are many fundamental concepts in linear regression that many other regression techniques build upon.

This thesis will use the multiple regression model, which is valid when five basic assumptions are met. When these assumptions are met the ordinary least squares (OLS) estimator is guaranteed to be the optimal estimator [Ken08].

The specification for the linear regression model is

$$y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ik}\beta_k + e_i \quad i = 1, 2, \cdots, n \tag{2.2.1}$$

In the expression, y_i is regarded as the dependent variable whose value depends on the covariates $x_{\bullet j}$. The parameters, β_j are unknown, as is typically the variance, and are to be estimated from the data. The error terms are normally distributed and denoted as e_i [Lan13].

It is often more convenient to employ matrix notation:

$$Y = X\beta + e \tag{2.2.2}$$

where,

Y is a $n \times 1$ vector given as $Y = \begin{pmatrix} y_i \\ \vdots \\ y_n \end{pmatrix}$, X is a $n \times (k \times 1)$ matrix given as $X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{pmatrix}$

β is $(k+1) \times 1$ vector given as $\beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_k \end{pmatrix}$ and e is a $n \times 1$ vector given as $e = \begin{pmatrix} e_0 \\ \vdots \\ e_n \end{pmatrix}$

2.2.2 Important Assumptions. The multiple linear regression model consists of five basic assumptions concerning the way in which the data are generated. In some cases it is permissible to violate the assumptions; in other cases they must be rigorously checked. However it should be noted, that as the data is divided into a *training* and *testing* set, the effects of violating any regression assumptions are minimised, as checking of errors from a test set will highlight the presence of any bad regression model. All listed assumptions are taken from [Ken08].

1. The first assumption is that the dependent variable can be calculated as a linear function of the explanatory variables, plus an error term. Thus, it should have the form of equation 2.2.1.
2. The second assumption is that the expected value of the error term is zero, which can be expressed mathematically as $E[e] = 0$. An estimator with the expected value of zero is called unbiased.
3. The third assumption is that the error terms all have the same variance and are not correlated with one another.
4. The fourth assumption is that the explanatory variables can be considered fixed in repeated samples, which means it is possible to redraw the sample with the same values for the covariates.
5. The fifth assumption is that there are no exact linear releationship between the expanatory variables.

Violation of this assumption:

- **Multicollinearity** - two or more covariates are approximately linearly correlated in the sample data. Further explained in section 2.2.4.

2.2.3 Ordinary Least Squares Estimation. The ordinary least square (OLS) estimator is considered the optimal estimator of the unknown parameters β when the assumptions of the multiple linear regression model are met [Ken08]. The estimates of the OLS is denoted with a hat; e.g., the OLS of β is expressed as $\hat{\beta}$ [Lan13]

The estimated $\hat{\beta}$ achieved by this method minimizes the sum of the squared errors. This is done by putting the derivative of the sum of the squared errors with respect to $\hat{\beta}$ equal to zero [Lan13].

The sum of the squared errors:

$$\sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.2.3}$$

$$= (y - X\hat{\beta})^T (y - X\hat{\beta}) \tag{2.2.4}$$

$$= y^T y - y^T X\hat{\beta} - \hat{\beta} X^T y + \hat{\beta}^T X^T X\hat{\beta} \tag{2.2.5}$$

The derivative with respect to $\hat{\beta}$:

$$\frac{\partial(y^T y - y^T X\hat{\beta} - \hat{\beta} X^T y + \hat{\beta}^T X^T X\hat{\beta})}{\partial\hat{\beta}} = 0 \tag{2.2.6}$$

$$-2y^T X + 2X^T X\hat{\beta}^T = 0 \tag{2.2.7}$$

$$y^T X = X^T X\hat{\beta}^T \tag{2.2.8}$$

$$Taking \quad transpose \quad on \quad both \quad sides \tag{2.2.9}$$

$$X^T y = X^T X\hat{\beta} \tag{2.2.10}$$

Hence, it follows that:

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{2.2.11}$$

Under the multiple linear regression model's assumptions the OLS method is unbiasedand thus $E(\hat{\beta}) = \beta$.

2.2.4 Multicollinearity. Multicollinearity is a phenomenon where two or more of the covariates are related to each other in such a way that the quantitative measure of the variables are linearly dependent to a large extent. If some covariates are collinear, the ordinary least square (OLS) estimates of these parameters will have a large variance.

2.2.5 Detecting Multicollinearity. Detecting collinearity of two covariates can be done in different ways. Below are two common ways to examine the phenomenon.

- **Scatter Plot:**

  Detecting multicollinearity can be done by putting all the measurements of each covariate in two separate, ordered, vectors and plotting them against each other. This way a scatter plot is constructed and if multicollinearity exists the measurements should be scattered around a straight line

- **Correlation Matrix:**

  A second way to detect multicollinearity is to construct the correlation matrix

$$R(X_1, X_2) = \frac{Cov(X_1, X_2)}{\sqrt{Cov(X_1, X_1)Cov(X_2, X_2)}} \tag{2.2.12}$$

  The off-diagonal elements in R represents the correlation coefficients for the data in question. A correlation coefficient above 0.8 indicates a high correlation between the variables.

2.3 Machine Learning Algorithm

Since our experiment takes the provided explanatories variables into account in making predictions, thus is it supervised, we will focus this section on supervised machine learning techniques for regression. Our focused is going to be on k-NN Algorithm, Gaussian Naive Bayesian and Random Forest and its preliminaries literatures.

Machine learning is the study of algorithms that can learn from data and make predictions, by building a model from example inputs rather than following static instructions [Bis06]. These algorithms are typically classified into three categories: *supervised learning, unsupervised learning and reinforcement learning*.

In *supervised learning*, the system is presented with example inputs and outputs, with the aim of producing a function that maps the inputs to outputs. Regression and classification problems are the two main classes of supervised learning [Mur12].

Unsupervised learning is concerned with leaving the system to find a structure based on the inputs, hopefully finding hidden patterns. Examples include density estimation [S+04], dimensionality reduction [B+10] and clustering [JMF99].

Lastly, *reinforcement learning* is the study of how a system can learn and optimise its actions in an environment to maximise its rewards [DW06], such as training a robot to navigate a maze.

The problem of predicting future housing prices can be considered a regression problem, since we are concerned with predicting values that can fall within a continuous range of outputs. Through regression, we will be able to explore the relationship between the independent variables we have selected and the property price.

2.3.1 Naive Bayesian Algorithms. Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. Given a class variable y and a dependent feature vector x_1 through x_n, Bayes' theorem states the following relationship:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{2.3.1}$$

Using the naive independence assumption that:

$$P(x_i|y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i|y) \tag{2.3.2}$$

for all i, this relationship 2.3.1 is simplified to:

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y))}{P(x_1, ..., x_n)} \tag{2.3.3}$$

Hence, instead of estimating the $P(y|X)$, the individual $P(y|x_i)$ can now be estimated separately. This dimensionality reduction makes the learning problem much easier. Because the amount of data needed to obtain an accurate estimate increases with the dimensionality of the problem, $P(y|x_i)$ can be estimated more reliably

2.3.2 k-NN Algorithms. Nearest neighbor methods belong to a class of non-parametric algorithms known as prototype methods [THF05]. They distinguish from other learning algorithms in the sense that they are memory-based and require no model to be fit.

The principle idea behind nearest neighbor methods is to find a number of training samples closest in distance to the new sample, and then infer from these the value of the output variable.

Hence, the choice of the distance used for measuring is crucial.

Distance Measure:

Since k-NN Algorithm is the grouping of similar objects, some sort of measure that can determine whether two objects are similar or dissimilar is required [RM05]. There are two main type of measures to estimate this relation: distance measures and similarity measures. Many k-NN algorithm methods use distance measures to determine the similarity or dissimilarity between any pair of objects. A valid distance measure should be symmetric and obtains its minimum value (usually zero) in case of identical vectors. The distance measure is called a metric distance measure if it also satisfies the following properties:

1. $d(x_i, x_j) = 0 \rightarrow x_i = x_j, \forall x_i, x_i \in S \in \mathbb{R}$
2. $d(x_i, x_j) = d(x_j, x_i), \forall x_i, x_j \in S \in \mathbb{R}$
3. Triangle inequality $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k), \forall x_i, x_j, x_k \in S \in \mathbb{R}$

- **Minkowski: Distance Measures for Numeric Attributes:**

  Given two p-dimensional objects, $x = (x_{i1}, x_{i2}, \cdots, x_{ip})$ and $(x_{j1}, x_{j2}, \cdots, x_{jp})$, the distance between the two data objects can be calculated using the Minkowski metric [HPK11]

  $$d(x_i, x_j) = (|x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + \cdots + |x_{ip} - x_{jp}|^g)^{\frac{1}{g}}$$

  The commonly used Euclidean distance between two objects is achieved when $g = 2$. Given $g = 1$, the sum of absolute par-axial distances (Manhattan metric) is obtained, and with $g = \infty$ one gets the greatest of the par-axial distances (Chebychev metric).

- **Distance Measure for Binary Attributes:**

  The distance measure described in the last section may be easily computed for continuous-valued attributes. The distance measures play a critical role in regression and classification [CCT10]. In the case of instances described by categorical, binary, ordinal or mixed type of attributes, the distance measure used is the Hamming distance.

  The Hamming distance, $d(x, y)$ between two strings x and y of the same length over a finite alphabet $\sum$, denoted $\Delta(x, y)$, is defined as the number of positions at which the two strings differ [Gur10], i.e.

  $$\Delta(x, y) = |\{i|x_i \neq y_i\}|$$

Once the k nearest neighbors are selected, the predicted value can either be the average of the k neighbouring outputs (uniform weighting), or a weighted sum defined by some function.

For regression, the k-nearest neighbor algorithm [FHJ51] averages the output values from the k closest training samples, that is:

$$\Phi(x) = \frac{1}{k} \sum_{(x_i, y_i) \in NN(x, \mathcal{L}, k)} y_i \tag{2.3.4}$$

where $NN(x, \mathcal{L}, k)$ denotes the k nearest neighbors of x in L. In general, the distance function used to identify the k nearest neighbors can be any metric, but the standard Euclidean distance is the most common choice.

Before building the KNN regression model, all the variables in the dataset are recommended to be centered and scaled to guarantee that contribution from all the variables is equally treated. And the optimal value of K can be decided by the resampling technique, since large K would lead to the regression under-fit, and small K would cause to the regression over-fit.

The accuracy of the predicted value can be very poor if the distribution of the dataset has no relationship with the predicted response.

The $k - NN$ algorithm is presented as follows:

Input: D, the set of k trainning object and test object, $z = (\mathbf{x}', y')$

Process:

Compute $d(\mathbf{x}', \mathbf{x})$, the distance between z and every objects, $(\mathbf{x}', y) \in D$

Select $D_z \subseteq D$, the set of closest trainning objects to z

output: $y' = \frac{1}{k} \sum_{(x_i, y_i) \in NN(x, \mathcal{L}, k)} y_i$

The cons of K-NN algorithm involve sensitive to noise, computationally expensive, large memory requirements and curse of dimensionality.

2.3.3 Regression Trees and Bootstrap Aggregating.

- **Regression Tree:**

  Regression Tree models is a special kind of nonlinear regression models, which can be used to predict continuous values by partitioning the dataset into small groups like trees with leaves and branches. It allows the input predictors to be a combination of continuous, categorical, skewed, sparse, etc. variables without the requirements of data preprocessing.

  The intuitive structure of the tree is easy to interpret and compute, and is capable to be well applied for large amounts of dataset without the need to know the relationship between the predicted response and the predictors.

- **Bagging Tree:**

  Bagging Tree, also called Bootstrap Aggregating [Bre96], is an effective approach to reduce the instability and improve the accuracy of the regression model under the regression tree methods.

  At first, it generates a certain number of new training sets by bootstrap sampling from the original dataset uniformly and with replacement. Then, a set of tree models can be trained independently by the new training sets. At last, the predicted responses of the different models are aggregated by averaging to create a single bagged prediction.

  Apart from the great reduction of the instability of the regression model, another advantage is that there are certain samples left as long as a bootstrap sample is generated, and these out-of-bag

samples can be used directly to evaluate the predictive performance of the corresponding model. So that, the predictive performance of the entire regression model can be estimated by the average value of the out-of-bag error estimates

2.3.4 Random Forest. Random Forest (RF) methodology is a machine learning technique useful for prediction problems. RF was developed to address drawbacks inherent to single decision trees such as instability and overfitting by adopting an ensemble approach. The RF algorithm, developed by Leo Breiman [Bre01], applies bootstrap aggregation (bagging) [Bre96] and random feature selection [AG97] to individual classification or regression trees for prediction.

In bagging, successive trees do not depend on earlier trees - each is independently constructed using a boot-strap sample of the data set. In the end, a simple majority average measure is taken for prediction.

In random feature selection, Breiman [Bre01] proposed Random Forest, which add an additional layer of randomness to bagging. In addition to constructing each tree using a different bootstrap sample of the data, Random Forest change how the regression trees are constructed. In standard trees, each node is split using the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node. This somehow counter-intuitive strategy turns out to perform very well compared to many other classifiers, including discriminant analysis, support vector machines and neural networks, and is robust against overfitting [Bre01].

The random forests algorithm for regression is as follows:

- Draw n_{tree} bootstrap samples from the original data.
- For each of the bootstrap samples, grow an *unpruned* classification or regression tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample m_{try} of the predictors and choose the best split from among those variables.
- Predict new data by aggregating the predictions of the n_{tree} trees (i.e., average values for regression).

The main drawback of Random Forest is its overfitting when dealing with noisy dataset.

2.4 Model Validation

When using regression in order to create a predictive model it is important to examine how well the model represents the data it is derived from and to what extent it is possible to use the model for predictive purpose. This type of analysis is referred to as model validation and may be done with different types of statistical tools. In this section we present the tools that we will later use in chapter 4 of this thesis.

2.4.1 Coefficient of Determination, R^2. R^2 is a measure of goodness of fit. It measures how well the covariates in the model explains the variance in the dependent variable. R^2 is equal to the square of the sample correlation coefficient between y and $x\hat{\beta}$[Lan13]

$$R^2 = \frac{Var(x\hat{\beta})}{Var(y)} \tag{2.4.1}$$

The sample variance of y can be decomposed into two terms:

$$Var(y) = Var(x\hat{\beta}) + Var(\hat{e}) \tag{2.4.2}$$

Thus, R^2 can also be expressed as:

$$1 - \frac{Var(\hat{e})}{Var(y)} \tag{2.4.3}$$

It follows from equation 2.4.3 that the model should have as high R^2 as possible since this minimizes the error term, $\hat{e}$ and therefore implies an improved estimation of the dependent variable y [Lan13].

2.4.2 Residual Analysis. The second assumption of the multiple linear regression model states that the expected value of the error term is zero. This is however seldom the case in practical applications. It is thus of importance to study the residual in order to examine in what extent assumption two may be violated. This will make it possible to recognise patterns in the residual that could increase the understanding of the regression and eventually improve it. This is referred to as residual analysis.

We recall the regression equation:

$$y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ik}\beta_k + e_i \qquad i = 1, 2, \cdots, n \tag{2.4.4}$$

When the regression is done and estimates of β_j are determined, the residuals $\hat{e}_i$ can be achieved by the following manipulation of equation 2.4.4 [Lan13]

$$\hat{e}_i = y_i - (\hat{\beta}_0 + x_{i1}\hat{\beta}_1 + \cdots + x_{ik}\hat{\beta}_k) \qquad i = 1, 2, \cdots, n \tag{2.4.5}$$

2.4.3 Root Mean Square Error, RMSE. The Root Mean Square Error (RMSE) (also called the root mean square deviation, RMSD) is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modelled. The RMSE represents the sample standard deviation of the differences between predicted values and observed values. These individual differences are also called residuals, and the RMSE serves to aggregate them into a single measure of predictive power.

The RMSE of predicted values $\hat{y}_t$ for times t of a regression's dependent variable y_t is computed for n different predictions as the square root of the mean of the squares (MSE) of the deviations, and is given as:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum(\hat{y}_t - y_t)^2} \tag{2.4.6}$$

It follows from equation 2.4.6 that the model should have as low RMSE as possible.

3. Data Analysis and Pre-processing

3.1 Pre-processing and Resampling Techniques

Data pre-processing is always needed during the implementation of machine learning algorithm, since different models have different requirements to the predictors in the model, and different data preparation can give rise to different predictive performance. The model validation technique can be often used to evaluate the model generalizability, where a training set is used to fit a model and the testing set is used to estimate the efficacy.

3.2 Data Overview

The objective of data transformation is to improve the performance of the model by reducing the negative effect of the outliers or skew in the dataset. Changing the number of variables in a model will affect the fitness of the model.

3.2.1 Dataset Description. The dataset used for this research was constructed by Dean De Cock in 2012 for the purpose of an end of semester project for an undergraduate regression course. The original data (obtained directly from the Ames Assessors Office) is used for tax assessment purposes but lends itself directly to the prediction of home selling prices. The type of information contained in the data is similar to what a typical home buyer would want to know before making a purchase.

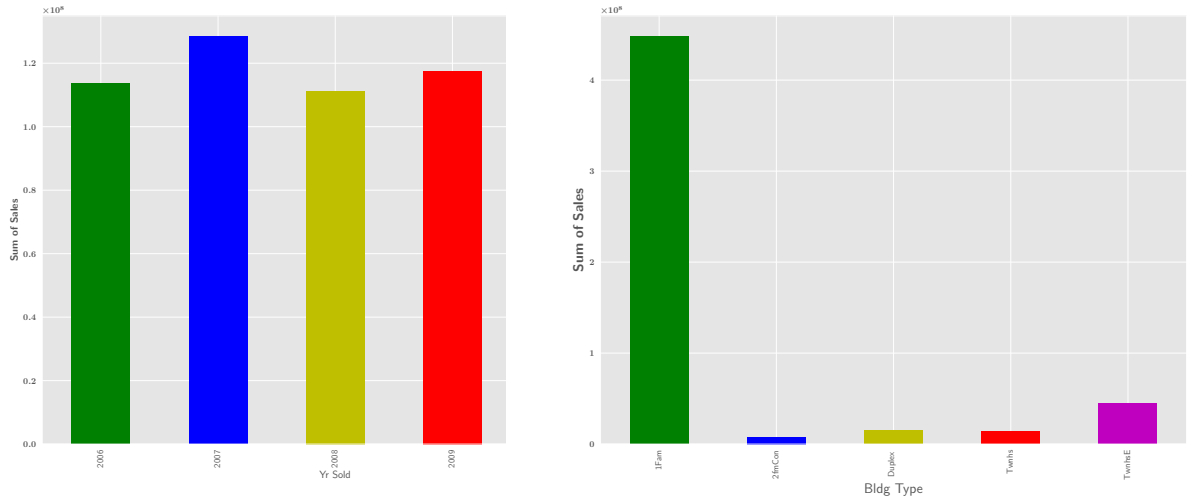
Our dataset contains details of assessed values for individual residential properties sold in Ames, Iowa from 2006 to 2010. We have 2930 entries total, each of them consisting of 82 features. Several entries have from one to many missing values and in section 3.3.4 we described how we dealt with this issue.

For each property transaction, various information are provided and can be categorised as follows:

- **ID** (Dwelling Type ID)
- **Date** (Year Built, Month Sold, Year Sold)
- **Property Classification** (Building Type, House Style, Living Area Square feet etc)
- **Address Information** (Zonal Classification, Neighbourhoods, Sales Types etc)
- **Sale Price** (the target variable)

In this dataset, we have 28 discrete numerical variables, 11 continous numerical variables and 43 categorical variables.

Further analysis of the dataset shows that the bulk of the property sales happened during 2006 to 2009 and a majority of the transactions are Single-Family House



(a) Sales by year (b) Sales by building type

Figure 3.1: A figure of Sales by year and building type

3.2.2 Data Visualization. In this section,we present the graphical representation of the target data and five explanatory variables chosen randomly.

- **Sale Price:**

  The sale price is the main variable we want to predict. From the density plot below, we notice that sale price deviates from the normal distribution, has appreciable positive skewness and shows peakness. The value of its skewness and kurtosis are 1.74 and 5.12 respectively. Each house prices has an average costs of $180,796

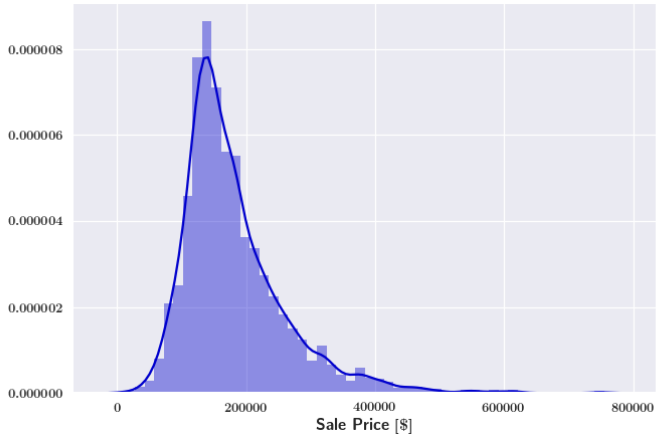


Figure 3.2: Distribution plot of houses Sale Price in USD. The solid blue line on top of the histogram represent the smoothed sale price density distribution estimation. The distribution has a skewness of 1.74 and kurtosis of 5.12.

- **Ground Living Area:**

  The ground living area describes the above ground living area per square feet. It has a high linearity with sales price. It also has a high CORRELATION with sales price (0.7068). An area per square feet of Ground Living Area costs $\$1499$

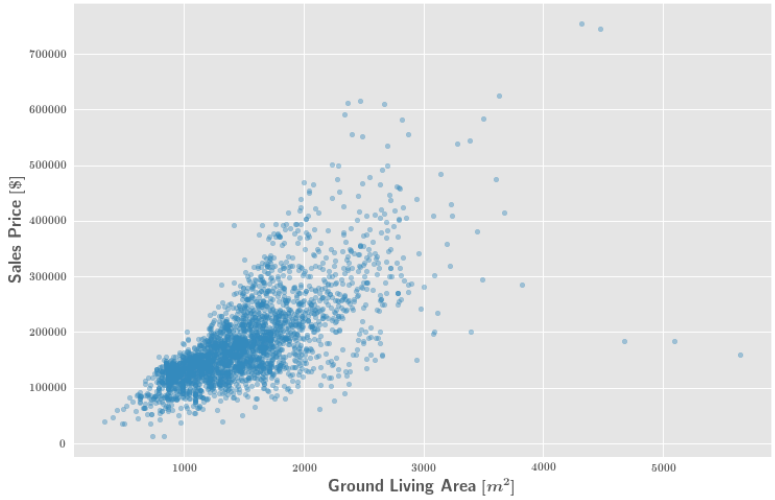


Figure 3.3: Scatter plot of Ground Living Area and Sale Price. We note a clear global trend of Sale Price increases as the Ground Living Area increases which is expected.

- **Garage Area:**

  The garage area describes the size of garage per square feet. From the graph below, we observe that as the area of garage increases so is the sale price increasing. Hence, there is a high linearity between Garage Area and Sale Price. The vertical line at zero corresponds to houses with no garage or with missing values. Each garage per square feet in Ames costs an average of $472

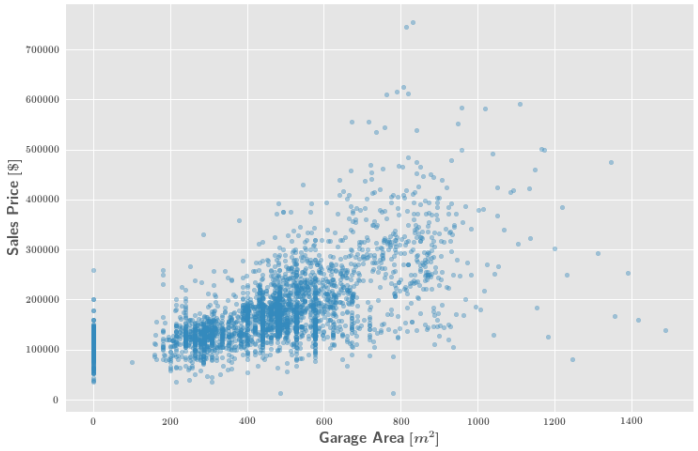


Figure 3.4: Scatter plot of Garage Area in m^2 and Sale Price in USD. The vertical line at zero represents the houses with no garage or missing values. This show the problem of missing or no values that could bias the regression

- **Overall Quality:**

  The overall quality rates describes the quality of the house material used for both construction and finishing respectively. It ranges from 1 to 10 with 1 been very poor and 10 very excellent. We obseved that the quality of the house affects the sale prices positively with high significance. 60 $\%$ of properties put up for sales has an above average ratings equivalent to rating 6.

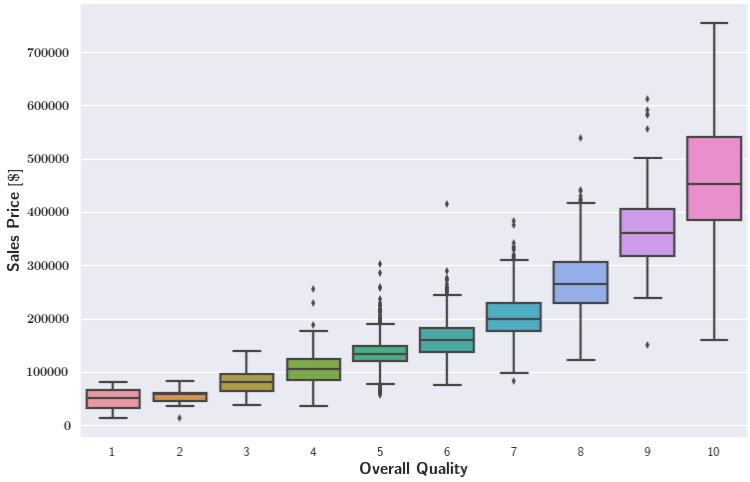


Figure 3.5: Box plot of Overall Quality and Sale Price in USD, The overall quality is a ratings of the total condition of the house, ranging from 1 to 10 accordingly. This shows a high correlation (0.7993) with the Sale Price as expected.

- **Lot Area:**

  Lot Area describes the lot size in square feet. This shows clear outliers which could also be a problem for the regression. More than 90% of lot area ranges between 0 - 50000. Each square feet of properties in Ames cost averagely $\$10147$

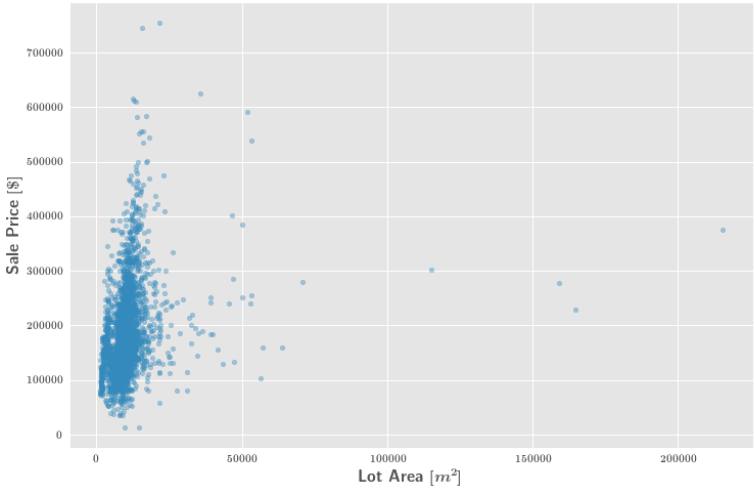


Figure 3.6: Scatter plot of Lot Area in m^2 and Sale Price in USD with low correlation (0.2665) which was not expected.

- **Total Basement:**

  Total basement describes the total square feet of basement area. There is clearly a trend but apparently two separate distribution. The vertical line at zero corresponds to houses with no basement or with missing values. The area of the total basement per square feet costs averagely $\$1051$

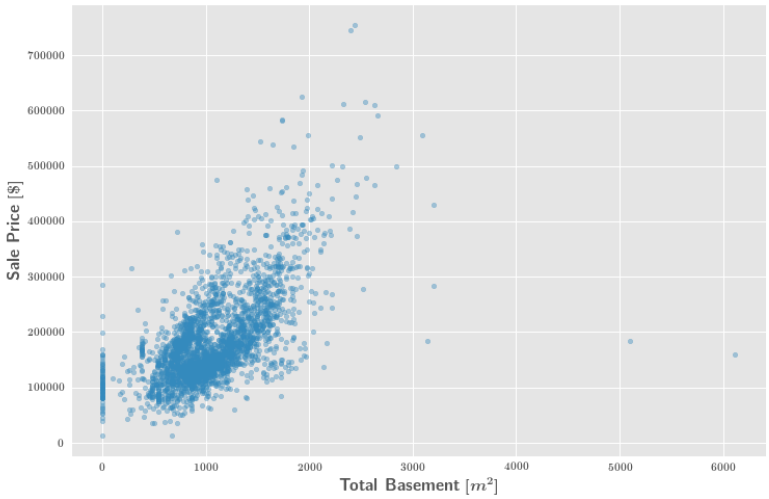


Figure 3.7: Scatter plot of Total Basement in m^2 and Sale Price in USD. The vertical line at zero represents houses with no basement or with missing values.

3.3 Missing Data Techniques

The missing data handling techniques for ignorable missingness are roughly classified into the following two categories:

1. Missing data ignoring techniques
2. Missing data imputation techniques

3.3.1 Missing Data Ignoring Techniques. These techniques omit the incomplete records which have missing values. They are very simple to use but are not efficient and produce biased results. They can be used only when the percentage of missingness is very low. It has two main techniques:

- **Litwise Deletion (LD)**

  It deletes any instances with missing values and performs statistical analysis with only available instances which are complete. Due to its simplicity and ease of use, LD is the default analysis in most of the statistical software. But its implementation has some drawbacks. Omitting the significant records may lead to loss of valuable information and relationship between attributes which is a major problem in large datasets. It subsequently reduces statistical power of the analysis.

- **Pairwise deletion (PD)**

  It is a variant of litwise deletion in which the incomplete instances are used for analysis only when the required attribute values are present in them. In this method, the available values in each observation of the attributes are considered separately i.e., different instances will be utilized for different cases [SEEM01]

3.3.2 Missing Data Imputation Techniques. Imputation techniques estimate the possible values to be substituted for the missing values using statistical analysis based on assumptions and models. It estimates values based on the related attributes and hence are found to be better than ignoring techniques particularly when the dataset is large.

- **Mean /Mode imputation (MMI):**

  In this method, mean of the available values in the dataset is used to replace the missing hole, if the attribute is continuous and mode (most frequent) value is used if the attribute is discrete [PS05]

  The pros of this method is that it is very simple and fast and it uses all the available values for the attribute and does not exclude any of the valid instances from analysis.

- **K Nearest Neighbor (KNN) Imputation**:

  In this method, nearest neighbors for the missing record are first identified by calculating the distance between missing instances and other complete instances in the dataset. Mode of the nearest neighbors is used as solution in case of discrete attributes and mean is used for continuous attributes. Efficiency of this method depends on the number of complete instances available in the dataset since only complete instances are used for analysis. An important factor to be carefully determined is the number of neighbors (k) to be used for analysis

3.3.3 Adding or Deleting Variables. During the implementation of regression models, adding or deleting variables can be kept on until a specified criterion is met. A model can be started with all the variables in the dataset, and then remove them one by one until the performance of the model would be degraded or the variables can be added to the model one by one, this processing can be stopped when adding variables would not improve the fitness the model at all.

There are several advantages to delete variables prior to modeling. First, removing variables is one of the important methods for dealing with multicollinearity, which would make it difficult to interpret the individual coefficients parameters in the regression model. Second, deleting variables with degenerate distributions can improve the stability of the system significantly. Third, fewer number of variables means fewer necessary resources, such as storage space and computational time.

3.3.4 Missing Values. As information were collected from the Ames Assessors' Office from 2006 to 2010, it appeared that the number of measurements were not the same for each datasets resulting in a sparse data set.

A first rough approach to missing values in the data set is simply to not consider elements with one or several missing measurements. This is a simple idea that cannot be applied here because it would have resulted in discarding nearly 30% of the data set.

It has also been suggested to replace missing measurements with the average of all available same measurements. Although this method has the advantage to keep the number of elements, it will bias statistical result toward the mean value.

Therefore we adopted the technique of both delection and mean/ mode imputations.

There are several advantages to delete variables prior to modeling. First, deleting variables with degenerate distributions can improve the stability of the system significantly. Second, fewer number of variables means fewer necessary resources, such as storage space and computational time.

The mean imputations deals with missing values for continous variables whilst mode imputation deals with missing values for categorical variables and median imputation for discrete variables.

Within the dataset, 27 features contained missing values made up of eleven (11) continous variables and sixteen (16) categorical variables. Table 3.3.4 summaries the number of missing values per feature and their percentages

Missing Variables	Sum of of missing data values	Percentage of missing values
PoolQC	2917	0.9956
Misc Feature	2824	0.9638
Alley	2732	0.9324
Fence	2358	0.8048
Fire placeQu	1422	0.4853
Lot Frontage (*cont.*)	490	0.1672
Garage Qual	159	0.0543
Garage Yr Blt (*cont.*)	159	0.0543
Garage Cond	159	0.0543
Garage Finish	159	0.0543
Garage Type	157	0.0534
BsmtExposure	83	0.0283
BsmtFin Type 2	81	0.0276
BsmtFin Type 1	80	0.0273
Bsmt Cond	80	0.0273
Bsmt Qual	80	0.0273
Mas Vnr Type	23	0.0079
Mas Vnr Area (*cont.*)	23	0.0079
Bsmt Full Bath (*cont.*)	2	0.0006
Bsmt Half Bath (*cont.*)	2	0.0006
Garage Area (*cont.*)	1	0.0003
Garage Cars (*cont.*)	1	0.0003
Total Bsmt SF (*cont.*)	1	0.0003
Bsmt Unf SF (*cont.*)	1	0.0003
BsmtFin SF 2 (*cont.*)	1	0.0003
BsmtFin SF 1 (*cont.*)	1	0.0003
Electrical	1	0.0003

Table 3.1: Missing Variables

3.4 Feature Importance and Selection

Feature selection, which is also named variable selection, is an approach to seek to capture a subset of the original variables or features for use in the implementation of the machine learning model in order to speed up the training time, enhance the learning interpretability and reduce the model over-fitting when there are many irrelevant features providing no more useful information than the current subset of variables. The irrelevant and redundant information in the dataset may greatly affect the performance of the regression model. The feature selection is often used when the number of features and the number of the observations (data points) are comparable in the dataset. And each variable in the new subset comes from the original set of variables.

Feature selection can be divided into three main categories: the *filter model*, the *wrapper model* and the *embedded model*. The *filter model* relies on a proxy measure (e.g. mutual information, Spearman correlation coefficient, significance test) to select some features in the original variables without any additional learning model on the training dataset. However, the wrapper model requires a specified predictive model for each new subset and uses the error rate of the model to score, and the subset with best performance is selected out. Since each subset is used to build the predictive model, it is much more computationally intensive than the filter model [YL03]. As is implied by the name, the embedded model conducts the feature selection as part of the predictive modelling process.

In this regression problem, we are going to use the filter model and the embedded model for the feature selections criteria.

3.4.1 Filtering the Variables. After importing the original files into Python, all the variables with more than 100 missing values are found out and deleted from the dataset. In the next step, we performed the mean and mode imputation, which have the following two characteristics: The mean imputation, for continous variables, calculates the mean value of each columns and then impute it to the missing values , and the mode imputation, for discrete, is that the most frequent discrete value is imputed to the missing values. Through these two steps, there are 11 delected variables. It means that there are only 71 variables left for predictive modeling, among which 9 variables with continuous values , 28 variables with discrete values and 34 categorical variables.

The third step is to filter the highly correlated variables, by calculating the correlation matrix of the 35 variables (for both continous and discrete variables) and the random forest features importances of all the 71 variables which can be seen in Figure 3.8 and 3.9 respectively.

- **Correlation matrix:**

  The color of each boxes represents the strength of correlation between the two variables, where the deep red color and the deep blue color are associated with the positive and negative relationship, respectively.

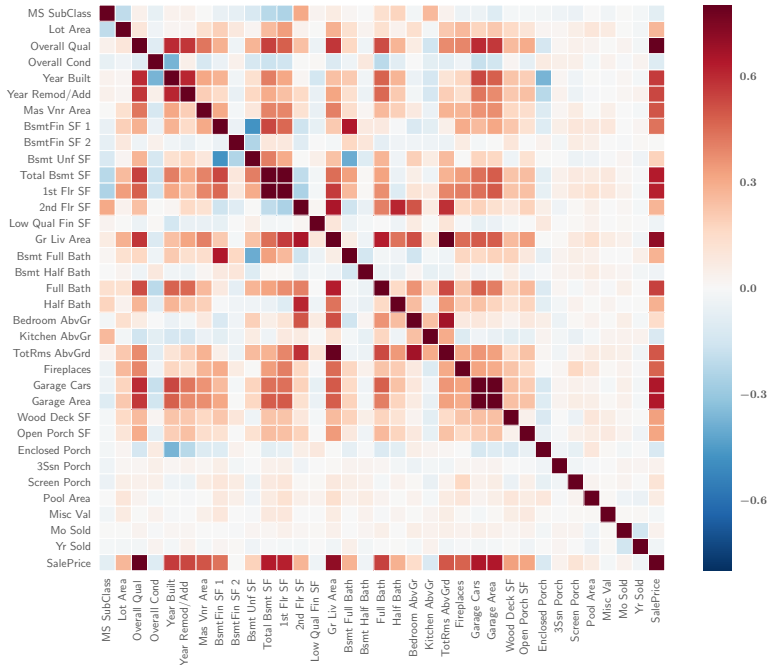


Figure 3.8: Correlation matrix of numerical variables in the dataset. We noticed that Total Bsmt SF and 1st Flr SF are highly correlated in addition to Garage Area and Garage Cars

There are at least 10 groups of highly correlated variables, such as one group of *Overall Qual*, *Gr Liv Area, Total Bsmt SF, Lot Area, Garage Cars, Year Built*, they are highly correlated with Sale Price, but almost independent with other variables. Also, we noticed that *Garage Cars* and *Garage Area* are highly correlated with each other as well as *Total Bsmt SF* and *1st Flr SF*.

- **Random Forest Feature Importances:**

  Five variables dominatly affects the value of house prices. The *Overall Qual* has a contribution of 65% to the sales prices, followed by *Gr Liv Area* (12%), *Total Bsmt SF* (4.3%), *Garage Cars* (1.5%) and *Lot Area* (1.2%)

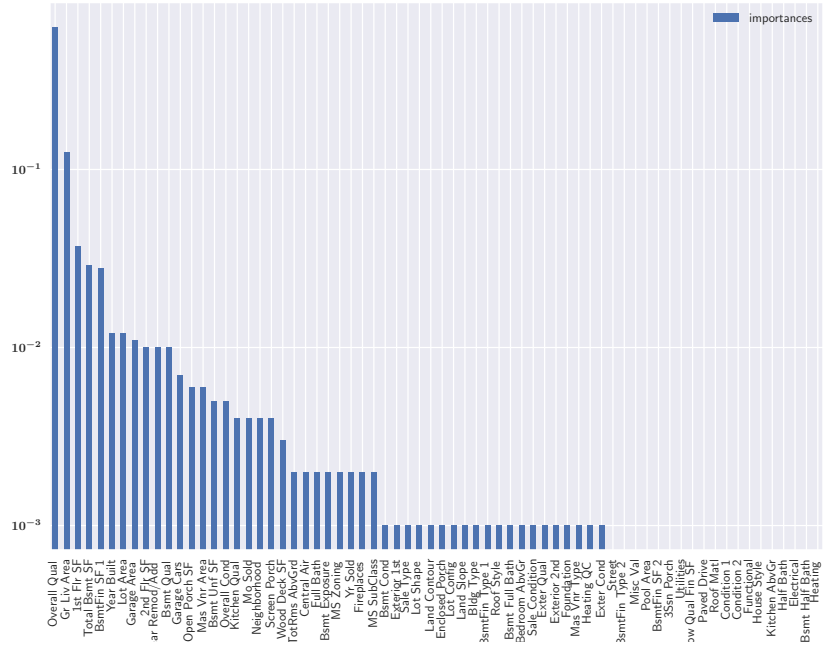


Figure 3.9: Log 10 Features importances of variables, both continous and categorical dataset. It shows that the first five variables contributes significantly to the value of the houses

The random features importances can be used to select the variables which are highly correlated with others for a given pair-wise absolute features importance value threshold. For a given cutoff 0.95, there are 66 variables returned, which can be deleted from the 71 variables. Therefore, after filtering these variables, there are only 5 useful variables with 410 observations left in the dataset, e.g. Overall Qual, Gr Liv Area, Total Bsmt SF, Lot Area and Garage Cars.

4. Implementation of Machine Learning Algorithm

After presenting the main regression algorithms and analyzing the data pre-processing and model-validating techniques in theory, four typical machine learning algorithms (Ordinary Linear Regression, Gaussian Naive Bayesian, k-Nearest Neighbours Regression and Random Forest Regression) are implemented on the housing dataset, and the corresponding performance of the built models are quantitatively and visually evaluated in details.

4.1 Algorithm Selection, Training and Testing

In this section, we present how the analysis was done, the metrics used for model performance and the split ratio of the training and testing dataset.

4.1.1 Analysis. The analysis we select is highly related with the type of prediction we want to make. As mentioned before, machine learning techniques that use regression predicts a value, learning from the relation between the input variables and the target value. In our experiment we are interested in the possibility to predict pricing indicators based on characteristics of the estate properties. Our experiment will show how well each algorithms perform on our dataset.

4.1.2 Performance Indicators. To assess the quality of our predictive system, we will use R^2 and Root Mean Squared Error (RMSE) for regression. All metrics are calculated by the SciKit-Learn and Scipy modules.

We have chosen the following three classifiers for our regressions task: *K-Nearest Neighbor Regressor* (kNN), *Gaussian Naive Bayesian* (GNB) and *Random Forest Regressor* (RFR).

4.1.3 Trainning and Testing. To select our classifiers for regression tasks, we created a development set. This set includes 82 attributes, was trained on 1,963 instances and was tested with 967 instances using a variety of regressors. This development set gave us the opportunity to compare a range of classifiers in a manageable amount of time, relative to using 67% of the total dataset.

We test our classifier with a 33-67 ratio as seen in other machine learning research ([SKKR00]; [DMS+02]). A training set of 2,930 instances would therefore have a test set of 967 instances, so a total number of 1,963 instances are used for this task.

To summarize, our analysis is conducted according to the following steps:

1. Create or load NumPy datasets
2. Split dataset into training set and test set (e.g. 67% training, 33% testing)
3. Fit the training data into an estimator using the SciKit-Learn module. Fitting is done using each regression algorithm
4. Calculate performance indicators using the SciKit-Learn module

4.2 Results

In this section, we will present the results of our experiment on different regression algorithms. We present our best performing configurations for R^2 and the corresponding RMSE. Detailed results for each regression techniques are also presented.

4.2.1 Multiple Linear Regression. First of all, the multiple ordinary linear regression is built by the LinearRegression() function with all the 1963 samples in the training set. The estimated regression coefficients in the multiple OLR model represent the value at which the dependable variable can be increased when the value of the independent variable is changed by one unit. For instance, the estimated coefficient for the variable *Overall Qual* is 12,377.8, indicating that an increase of each unit in *Overall Qual* would lead to a 12,377.8 units increase in the Sale Price, keeping the other variables fixed.

Measuring Performance in OLS Model:

Building the model is the first step on the way to do the prediction. But because the linear regression model is only fitted for the training set, we do not know how well this model will perform in the testing set.

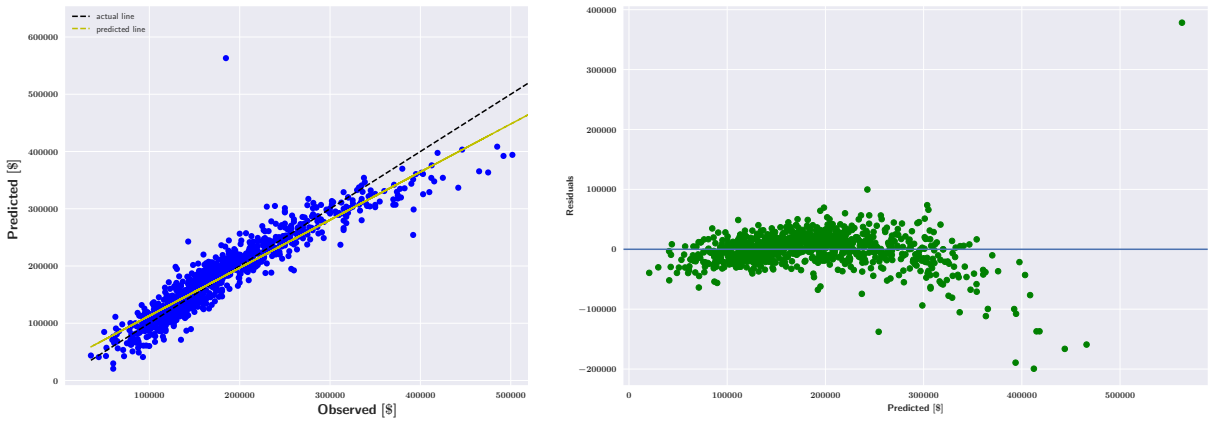
- **Quantitative Measures of Performance:**

  For regression models predicting a continuous numeric outcome, the Root Mean Squared Error (RMSE) and the coefficient of determination (R^2) are often used to evaluate the performance of the model.

  The value of the RMSE and R^2 for the testing dataset is 29,942.97 and 0.8674, respectively. Although the linear regression model with a 86% R^2 is optimistic, the average distance between the observed and the predicted values is quite large, which means that the linear regression model owns an average predictive accuracy.

- **Visualizations of the Linear Regression Model Fit:**

  Visualizations of the model fit are very useful to understand the strengths and weaknesses of the regression model, especially the observed vs predicted plots and the predicted vs residual plots. As illustrated in Figure 4.1, the left is the plot of observed values versus the predicted outcomes for the testing dataset. It only has a tendency to over-predict the low observed values, and all the other points are mainly located around the diagonal line. The right plot is of the predicted values versus the residual values, where all the points are almost randomly distributed around the horizontal line, apart from the bottom-right corner with residual mean of -607.64 and standard deviation of 29936.81. Comparing with the k-NN regression model fit in Figure 4.3, all the points in OLS model are closer to the diagonal line in observed vs. predicted plot, and all the points in OLS model are not only nearer the horizontal line in observed vs. residuals, but also the variance of those points are apparently lower than that in Figure 4.3. Therefore, the performance of the model fit is explicitly improved by the Ordinary Least Square method.



(a) Predicted plot (b) Residual plot

Figure 4.1: Visualizations of the Linear Regression Model Fit with residual mean of -607.64. The left plot indicates little variance between the actual line and predicted line

4.2.2 Gaussian Naive Bayesian (GNB). The Naïve Bayes algorithm makes use of Bayes' Theorem, which is a formula that determines a probability by estimating the frequency of values and mixture of values in the previously collected data. It determines the probability of an event happening provided that the probability of another event that has already happened.

The Naive Bayes algorithm provides a way to mix the prior probability and conditional probabilities within a single formula that can be used to determine the probability of each of the classifications in turn. After that, the class with the highest value will be chosen as the class of the new instance.

Measurement Performance of Gaussian Naive Bayesian:

After building the GNB model based on the training set, the main purpose of the model is to make the corresponding prediction by the model. For instance, the predict() function is applied to make the prediction.

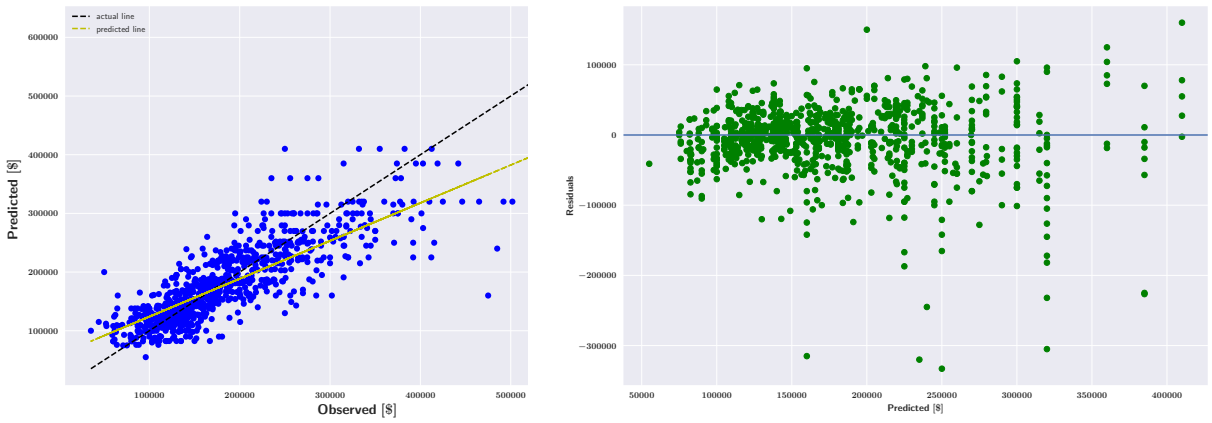
- **Quantitative Measures of Performance:**

  The value of the RMSE and R^2 for the testing dataset is 47,740.99 and 0.0072. The regression line with R^2 is very poor and the distance between the actual values and its prediction is too large.

- **Visualizations of the GNB Model Fit:**

  In Fig 4.2, the left is the plot of observed values against the predicted outcomes value for the testing dataset. It has a high tendency to under-predict the higher values. The plot on the right is the residuals plot where all the points are highly un-randonmly distributed and the variance of the residuals values is very large with mean of -5601.53 and standard deviation of 47411.23.

  From the two plots, we can conclude that Gaussian Naives Bayesian is a very bad model for predictions on this dataset.



(a) Predicted plot (b) Residual plot

Figure 4.2: Visualizations of the Gaussian NB Model Fit with residual mean -5601.53. On the left we noticed the high variance between the actual line and predicted line

4.2.3 k-Nearest Neighbours. The k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors . The core function has a number of parameters such as : k neighbours, leaf size, metrics and so on

Choosing Tunning Parameters:

We also carried out a regression fit on both the defaults parameters as well as tunning its parameters.

- **k-NN Without parametric tunning:**

  k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until regression. The k-NN algorithm is among the simplest of all machine learning algorithms.

  Both for classification and regression, it can be useful to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. The default metric used is minskowki (at p = 2) and the weight assigned is uniform, in which each points in each neighborhood are weighted equally.

- **k-NN with parametric tunning:**

  After tunning the parameters with the following options: n neighbours = 3, the number of neighbours to use for distance metrics, weight = distance, weight points by the inverse of their distance. In this case, closer neighbors of a query point will have a greater influence than neighbors which are further away and metrics = manhattan distance (at p = 1).

Measurement Performance for k-NN Model:

After building the k-NN model based on the training set, the main purpose of the model is to make the corresponding prediction by the use of the model. The predict() function is applied to make the prediction.

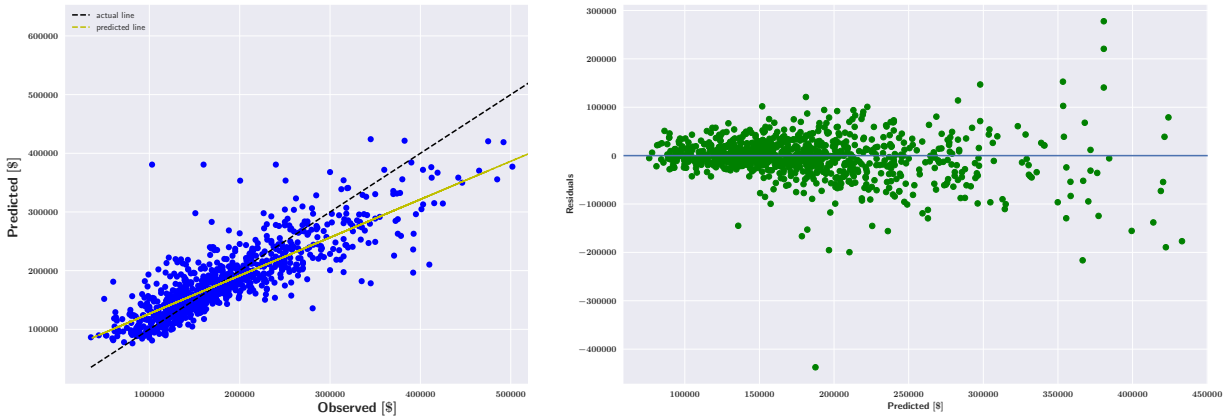
- **Quantitative Measures of Performance:**

  For the default parameters, the value of the RMSE and R^2 for the testing dataset is 44,185.35 and 0.7112. Although its regression line with R^2 is good, its distance between the actual values and its predictions is very large.

  After, the parameters tunnings, the value of the RMSE, decreased to 41,217.62 and R^2 increased to 0.7487.

- **Visualizations of the k-NN Model Fit:**

  In Figure 4.3, the left is a plot of the observed values versus the predicted outcomes where the R^2 for the testing dataset is 0.7112, but the k-NN model has a tendency to under-predict the high observed values. The right is a plot of the predicted values versus the residual values, in which all the points are apparently not randomly distributed, and the variance of the residual values is quite large with residual mean of -2762.00 and standard deviation of 44098.94. Thus, the two plots can also give us the conclusion that the k-NN model is not good enough for the prediction.



(a) Predicted plot (b) Residual plot

Figure 4.3: Visualizations of the k-NN Model Fit with residual mean of -2762.00. On the left is the plot of actual line and predicted line which are far from each other.

4.2.4 Random Forest. Random Forest is an ensemble machine learning method used for both classification and regression. In the case of regression, which is mainly implemented by building a great number of decision trees during the training time and outputting the averaging forest's prediction of the individual trees. The core function RandomForestRegressor() requires several tuning parameters: number of trees to grow tree , number of variables at each random split selection and so on.

Choosing Tuning Parameters:

For the purpose of this thesis, the results obtained is divided into two parts: analysis without parametric tunning and analysis with parametric tunning

- **RF Without parametric tunning:**

  In the first case, for comparison analysis, the parameters were not tuned, hence the number of trees for the random forest is set at default value. It is worth noting that increasing the n tree will not lead to negative influence on the model, since Breiman had proved that the random forest regression model is protected from over-fitting [Bre96]. However, the larger the random forest, the more time we will spend on training and building the model. Therefore, the default value 10 trees is used in our experiment as a starting point. Then we can train over this parameter.

  As we know, the random forest regression model is also a non-deterministic algorithm, in which randomly selected variables at each split probably give rise to totally different predictions. Thus, the RandomForestRegressor function is also run 5 times independently to select the table tuning parameter with minimum RMSE and maximum R^2

  All the five optimal results for the five instances of the function can be seen in Table 4.1. The optimal values of the number of randomly selected variables to choose is set at 10, and the minimum RMSE and maximum R^2 appears in the fouth model with the values 25,443.76 and 0.9043, respectively. According to the optimal results in Table 4.1, the tuning parameters of the Random Forest model can be chosen at n $= 10$.

  Table 4.1: Optimal Results of the RandomForestRegressor Without Parameter Tunning

Case	n_{tree}	RMSE	R^2
1	10	26,345.93	0.8973
2	10	26,072.39	0.8994
3	10	25,537.99	0.9035
4	10	25,443.76	0.9043
5	10	26,594.22	0.8954

- **RF With Parametric tunning:**

  After the tunning parameters choosing process, the option n estimators $= 150$ represents the number of trees in the forest to be increased from the default 10, and max depth $= 25$ means that the maximum depth of the tree is restrained.

  All the five optimal results for the five instances of the function can be seen in the following Table 5.2. The minimum RMSE and maximum R^2 appears in the first model with the values 23,941.88 and 0.9152, respectively.

Measuring Performance in RF Model:

After building the RF model based on the training set, the main purpose of the model is to make the corresponding prediction by the model. For instance, the predict() function is applied to make the prediction.

Table 4.2: Optimal Results of the RandomForestRegressor With Parameter Tunning

Case	n_{tree}	RMSE	R^2
1	150	23,941.88	0.9152
2	150	24,008.44	0.9147
3	150	24,348.22	0.9123
4	150	24,358.92	0.9122
5	150	24,748.34	0.9094

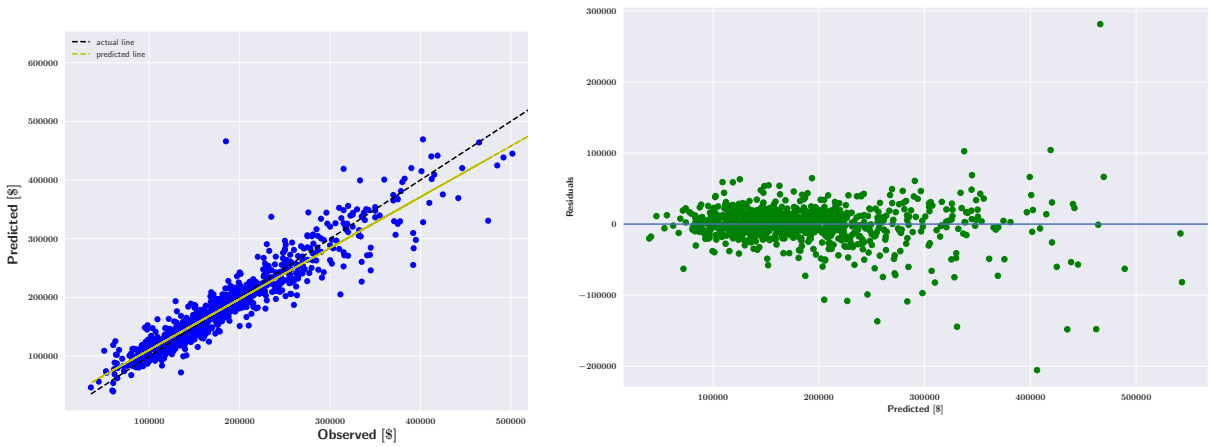
- **Quantitative Measures of Performance:**

  Owing to the non-deterministic characteristic of the random forest model, the randomForest() function has been run five times to get the best model with minimum RMSE on the testing set. The quantitative results of the five RF models can be seen that the values of RMSE and R^2 are very stable, varying from about 25,443.76 to 26,594.22 and from 0.9043 to 0.8954, respectively.

  Comparing with the corresponding results in the k Nearest Neighbours regression model (RMSE = 44,185.35 R^2 = 0.7112) and the ordinary linear regression model (RMSE = 29,942.97, R^2 = 0.8674), the quantitative performance of the RF model is apparently better than the k-NN, GNB and OLR models.

- **Visualizations of the RF Model Fit:**

  As shown in Figure 4.4, the left is the plot of observed values vs. predicted outcomes in RF Model 5 where the RMSE and R^2 for the testing dataset are 24,748.34 and 0.9094,respectively. Similar to the OLS model, it still has a tendency to over-predict the observed values, and all the other points are also mainly located around the diagonal line. The right plot is of predicted values vs. residual values, where all the points are almost randomly distributed around the horizontal line, except for the bottom-right corner due to the over-prediction for the high values with residual mean of -385.96 and standard deviation of 26493.02



(a) Residual plot

Figure 4.4: Visualizations of the RF Model Fit with residual mean of -385.96. On the left, is the plot of actual line and predicted line which are very close to each other.

5. Conclusions/Discussion

5.1 Conclusion

With the advent of the era of big data, machine learning has been widely used in many technologies and industries, which is able to get computers to learn without being explicitly programmed. As one of the fields of the supervised learning techniques, some classical models in each type are also presented, such as Ordinary Linear Regression (OLR), Gaussian Naive Bayesian (GNB), K-Nearest Neighbors (KNN) and Random Forest. The basic principal, strengths and weaknesses of each representative model are also illustrated as well.

After that, the data pre-processing and resampling techniques, including data transformation, feature selection and model validation, are explained in theory which can be used to effectively improve the performance of the training model.

During the implementation of machine learning algorithms, four typical models (Ordinary Linear Regression, Gaussian Naive Bayesian, k-Nearest Neighbours and Random Forest) have been implemented by the different packages in Python on the given big dataset.

Apart from the model training, the regression diagnostics are conducted to explain the predictive ability of the simplest ordinary linear regression model and k-Nearest Neighbours. Because of the non-deterministic characteristics of the Random Forest models, several models with in the dataset are built to get the reasonable tuning parameters, and the optimal models with minimum RMSE and maximum R^2 are chosen among several training models.

At the last step, the corresponding performance of the built models are quantitatively and visually evaluated in details.

Comparing with the ordinary linear regression model is the model performance.

Table 5.1: Results of the Models Without Parameter Tunning

Models	RMSE	R^2
Ordinary Least Aquare	29,942.97	0.8674
k-Nearest Neighbours	44,185.35	0.7112
Gaussian Naive Bayesian	47,740.99	0.0072
Random Forest	25,443.76	0.9043

Table 5.2: Results of the Models With Parameter Tunning

Models	RMSE	R^2
k-Nearest Neighbours	41,217.62	0.7487
Random Forest	23.941.88	0.9152

Clearly, Random Forest outperforms the other algorithm both before and after parametric tunnings due to the fact that it works on both categorical and continous variables and the effects of outliers in the dataset is minimized.

5.2 Discussion

5.2.1 The metrics used:. For all techniques used in scikit built k-Nearest Algorithms, we used the standard euclidean and manhattan distance. It would be interesting to investigate different metrics such as Minskowski with higher "p" values and Mahalanobis metrics.

5.2.2 The data imputation techniques used:. Apart from replacing missing values we haven't done much pre-processing. But pre-processing is often crucial. Hence it will be interesting to investigate other missing data techniques such as k-Nearest neighbours imputations and Regression techniques.

5.2.3 Algorithms used:. Apart from the four algorithms used, there are several supervised machine learning algorithms that can be used such as Neural Networks and Support Vector Machine which are robust algorithms known to perform well.

5.2.4 Parametric Tunnings:. There are more parameters to the method used in this work. A future work would be to investigate further the other parameters and find the very best combination

Appendix A. Appendix

A.1 Machine Learning Tools

There are a variety of options available for dataset analysis. We used the $SciKit - Learn^3$. The choice was made to use the SciKit-Learn module for dataset analysis because this tool seems to fit all our needs. The following are the main modules used in addition to the scikit-learn module

- **Scikit-Learn:**

  The SciKit-Learn module in Python provides the option to perform classification tasks as well as regression tasks. It is built on Numpy, Scipy and matplotlib. This module requires an n-dimensional array dataset, can handle multi-values attribute variables and is highly customizable. Due to the considerable usage among other data scientists and open source license, a substantial body of knowledge is also available.

  During the analysis of our dataset we first transform our discrete attribute from a single string into labels. We perform this step using the SciKit-Learn LabelEncoder tool that creates a label for each categorical variables in the dataset

- **Scipy:**

  SciPy is a collection of mathematical algorithms and convenience functions built on the Numpy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

  All the model validation analysis were done on the scipy modules.

- **Numpy:**

  Another powerful python package for scientific commputing that can create datasets is Numpy. NumPy is a Python extension that can modify data into large, multi-dimensional arrays and matrices on which high-level mathematical functions can operate.It is useful linear algebra, Fourier transform, and random number capabilities

- **Matplotlib:**

  Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook and web application servers.

  All the plots generated in this thesis uses matplotlib modules

- **Pandas**

  pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It is the fundamental high-level building block for doing practical, real world data analysis in Python.

  The two primary data structures of pandas, Series (1-dimensional) and DataFrame (2-dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering.

A.2 Pairplot of Sale Price and its five significant explanatory variables

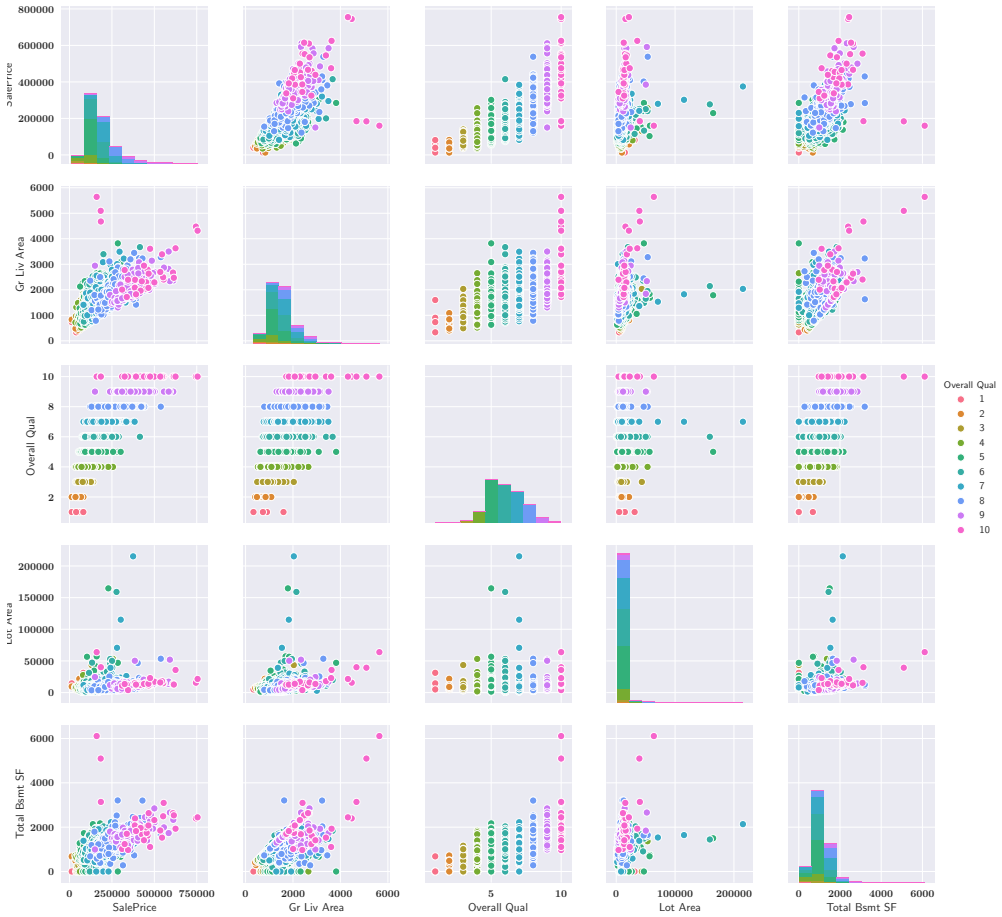
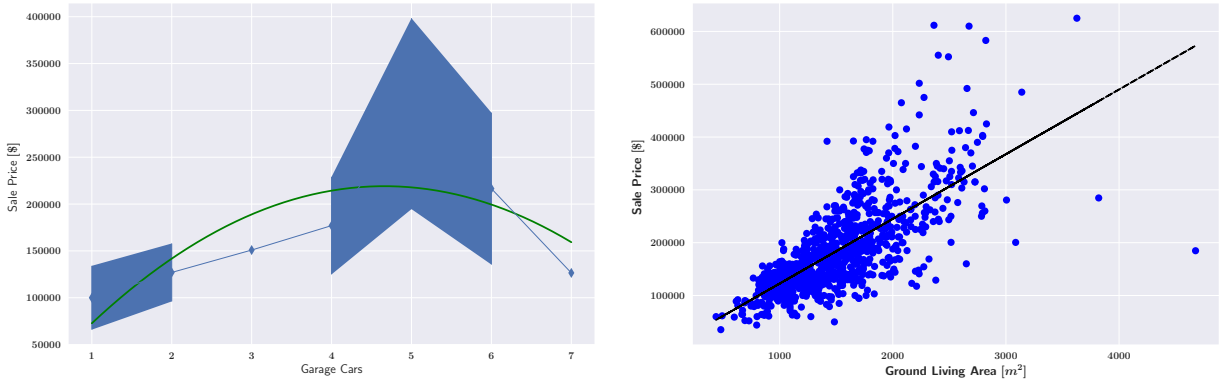


Figure A.1: Pairplot of Sale Price

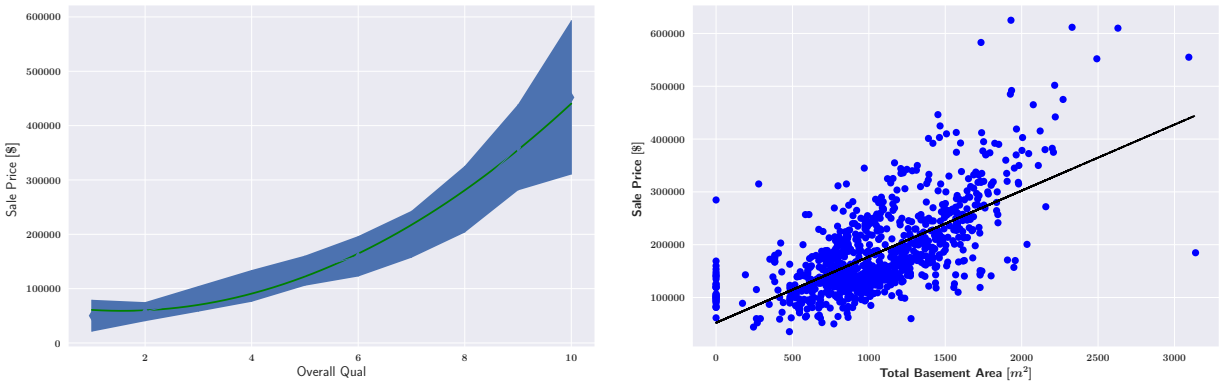
A.3 Regression fit of four significant variables with Sale Price



(a) Regression line of Garage Cars

(b) Regression line of Ground Living Area

Figure A.2: Regression line of Garage Cars and Ground Living Area



(a) Regression line of Overall Quality

(b) Regression line of total Basement

Figure A.3: Regression line of Overall Qual and Total Basement

References

[AG97] Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural computation*, 9(7):1545–1588, 1997.

[AR03] Nicholas Apergis and Anthony Rezitis. Housing prices and macroeconomic factors in greece: prospects within the emu. *Applied economics letters*, 10(12):799–804, 2003.

[B+10] Christopher JC Burges et al. Dimension reduction: A guided tour. *Foundations and Trends® in Machine Learning*, 2(4):275–365, 2010.

[Bis06] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[Bre96] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Cal03] C.A. Calhoun. Property valuation models and house price indexes for the provinces of thailand: 1992 – 2000. *Housing Finance International*, page 17(3), 2003.

[CCT10] Seung Seok Choi, Sung Hyuk Cha, and Charles C Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.

[CH00] David E Clark and William E Herrin. The impact of public school attributes on home sale prices in california. *Growth and change*, 31(3):385–407, 2000.

[DMS+02] Jenny Donovan, Nicola Mills, Monica Smith, Lucy Brindle, Ann Jacoby, Tim Peters, Stephen Frankel, David Neal, and Freddie Hamdy. Quality improvement report: Improving design and conduct of randomised trials by embedding them in qualitative research: Protect (prostate testing for cancer and treatment) study. *BMJ: British Medical Journal*, pages 766–769, 2002.

[DW06] Peter Dayan and Christopher JCH Watkins. Reinforcement learning: A computational perspective. *Encyclopedia of Cognitive Science*, 2006.

[EF95] Ahmet M Eskicioglu and Paul S Fisher. Image quality measures and their performance. *IEEE Transactions on communications*, 43(12):2959–2965, 1995.

[FGM00] Mike Fletcher, Paul Gallimore, and Jean Mangan. Heteroscedasticity in hedonic house price models. *Journal of Property Research*, 17(2):93–108, 2000.

[FHJ51] Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis, nonparametric discrimination: consistency properties. Technical report, DTIC Document, 1951.

[FJ03] J. Fred and G.D. Jud. Estimating the value of apartment buildings. *The Journal of Real Estate Research*, 25(1):77 – 86, 2003.

[FOK06] Gang-Zhi Fan, Seow Eng Ong, and Hian Chye Koh. Determinants of house price: A decision tree approach. *Urban Studies*, 43(12):2301–2315, 2006.

[Gur10] Venkatesan Guruswami. Lecture notes in introduction to code theory, January 2010.

[HPK11] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[HV12] Olga Hrydziuszko and Mark R Viant. Missing values in mass spectrometry based metabolomics: an undervalued step in the data processing pipeline. *Metabolomics*, 8(1):161–174, 2012.

[JMF99] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[Ken08] Peter Kennedy. A guide to modern econometrics, 2008.

[KTDR06] Yan Kestens, Marius Thériault, and François Des Rosiers. Heterogeneity in hedonic modelling of house prices: looking at buyers' household profiles. *Journal of Geographical Systems*, 8(1):61–96, 2006.

[Lan13] Harald Lang. Topics on applied mathematical statistics. *Stockholm: KTH*, 2013.

[LGW98] Lentz, G.H., and K. Wang. Residential appraisal and the lending process: A survey issue. *Journal of Real Estate Research, 15(1-2)*, page 11–40, 1998.

[LZW06] Jian-Guo Liu, Xiao-Li Zhang, and Wei-Ping Wu. Application of fuzzy neural network for real estate prediction. *Advances in Neural Networks-ISNN 2006*, pages 1187–1191, 2006.

[MPV15] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2015.

[Mur12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[PB02] Robert W Paterson and Kevin J Boyle. Out of sight, out of mind? using gis to incorporate visibility in hedonic property value models. *Land economics*, 78(3):417–425, 2002.

[PPS+09] Pashardes, Panos, Savva, Christos S., et al. Factors affecting house prices in cyprus: 1988-2008. *Cyprus Economic Policy Review*, 3(1):3–25, 2009.

[PS05] Cristian Preda and Gilbert Saporta. Clusterwise pls regression on a stochastic process. *Computational Statistics & Data Analysis*, 49(1):99–108, 2005.

[RM05] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005.

[S+04] Simon J Sheather et al. Density estimation. *Statistical Science*, 19(4):588–597, 2004.

[SEEM01] Kevin Strike, Khaled El Emam, and Nazim Madhavji. Software cost estimation with incomplete data. *IEEE Transactions on Software Engineering*, 27(10):890–908, 2001.

[Sel09] Hasan Selim. Determinants of house prices in turkey: Hedonic regression versus artificial neural network. *Expert Systems with Applications*, 36(2):2843–2852, 2009.

[Shi07] R.J. Shiller. Understanding recent trends in house prices and home ownership. *National Bureau of Economic Research*, Working Paper 13553:DOI : 10.3386/w13553, 2007.

[SKKR00] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000.

[THF05] J Friedman T Hastie, R Tibshirani and J Franklin. The elements of statistical learning: data mining, inference and prediction. *Mathematical Intelligencer*, 2005.

[Xu08] Ting Xu. Heterogeneity in housing attribute prices: A study of the interaction behaviour between property specifics, location coordinates and buyers' characteristics. *International Journal of Housing Markets and Analysis*, 1(2):166–181, 2008.

[YL03] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.