**Housing Price Prediction**

Submitted In partial fulfilment of the degree of

# **BACHELOR OF TECHNOLOGY**

**Submitted by**

Aishwarya Singh

Regd. No.: 1501341013





**DEPARTMENT OF COMPUTER SCIENCE AND ENGG.**

SILICON INSTITUTE OF TECHNOLOGY, SAMBALPUR

**CERTIFICATE**

This is to certify that the work contained in the thesis entitled “Housing Price Prediction, submitted by Aishwarya Singh (Regd. No.:1501341013) for the award of the degree of Bachelor of Technology to the Silicon Institute of Technology, Sambalpur, is a record of bonafied research works carried out by him under my direct supervision and guidance.

I considered that the thesis has reached the standards and fulfilling the requirements of the rules and regulations relating to the nature of the degree. The contents embodied in the thesis have not been submitted for the award of any other degree or diploma in this or any other university.

|  |  |  |
| --- | --- | --- |
| **Guide** | **External** | **H.O.D** |
| Ms. Madhusmita Samal |  | Mr. Saroj Kanta Misra |
| Asst. Professor, C.S.E. | . | Asst. Professor, C.S.E. |
| SIT, Sambalpur |  | SIT ,Sambalpur |

**DECLARATION**

I certify that the work contained in the thesis is original and has been done by myself under the supervision of my supervisor. The work has not been submitted to any other Institute for any degree or diploma. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Whenever I have quoted written materials from other sources and due credit is given to the sources by citing them.

**Date:**  Aishwarya Singh

**Place:** Sambalpur **Regd. No**.: 1501341013

**ACKNOWLEDGEMENT**

I would like to express my hearty indebtedness to all those who helped me during my project work, especially to my advisor **Ms. Madhusmita Samal** for her valuable suggestions, motivations and guidance right from scrap up to the completion**.** Her expertise and guidance proved to be valuable during every stage of this research. Thanks for patiently listening to all the silly questions and for the constant motivation to strive for excellence.

I wish to express my gratitude to the HOD, CSE and other faculty members of the Department of Computer Science & Engineering for teaching me and helping me whenever required. Thank you all teachers, you have been a great source of inspiration.

Lastly, I am thankful to my parents, without their support and inspiration, this thesis would not have been possible.

**Aishwarya Singh**

**Approval Sheet**

This dissertation entitled, “**Housing Price Prediction”** by Aishwarya Singh is approved for the degree of **Bachelor of Technology** in **Computer Science and Engineering.**

**Examiner**

**Supervisor**

Date:

Place: Sambalpur

**Abstract**

House prices increase every year, so there is a need for a system to predict house prices in the future. House price prediction can help the developer determine the selling price of a house based on the features and can help the customer to arrange the right time to purchase a house and helps seller in selecting the right selling price of the house. There are so many factors that influence the price of a house which includes **House style, Ground living area**, **Total rooms**, **Year built**, etc. In this dataset there are 81 features from which only important features has been taken as independent variable in feature engineering. First I fitted the linear regression model and to reduce the mean squared error I have applied log transformation in the dependent variable. This research aims to predict house prices based on the features of it with **multivariate linear regression** and **random forest** and **batch gradient descent** for optimization. The result from this research proved that random forest regression algorithm is suitable and get the minimum prediction error obtained and it has given the accuracy of **99.72%** which is better than the result of **linear regression**.

*keywords:* prediction model, target variable, linear regression, gradient descent, random forest, performance metric.

**List of figures**

|  |  |
| --- | --- |
| Chapters | Page |
| Chapter 3 |  |
| Figure 3.1: | 14 |
| Chapter 5 |  |
| Figure 5.1: Behaviour of outlier | 18 |
| Figure 5.2: Regplot between all the features | 19 |
| Figure 5.3: Regplot of GrdLivarea vs SalePrice | 20 |
| Figure 5.4: Regplot of TotalBsmtSF vs SalePrice | 20 |
| Figure 5.5: Regplot of 1stFlrSF vs SalePrice | 21 |
| Chapter 7 |  |
| Figure 7.1: Scatter plot to show regression line | 24 |
| Figure 7.2: Scatter plot to show error | 25 |
| Figure 7.3: Random forest tree | 27 |
| Figure 7.4: Feature importance | 28 |
| Chapter 9 |  |
| Figure 9.1: Residual plot of linear model | 32 |
| Figure 9.2: Residual plot of transformed linear model | 33 |
| Chapter 10 |  |
| Figure 10.1: Gradient Descent | 36 |
| Figure 10.2: Stochastic Gradient Descent | 38 |

**Contents**

[**Introduction** 8](#_Toc3593097)

Problem Statement…………………………………………………………………………………………………….8

Objective of the problem…………………………………………………………………………………………….8

Organization of the problem……………………………………………………………………………………….8

1. Introduction…………………………………………………………………………………………………………………9

2. Data Exploration and Preparation……………………………………………………………………………..10

2.1 Understanding data………………………………………………………………………………………10

2.1.1 Data fields………………………………………………………………………………………..10

2.1.2 Variable identification………………………………………………………………………11

2.2 Univariate analysis………………………………………………………………………………………..12

2.3 Bivariate analysis…………………………………………………………………………………………..12

3. Correlation matrix……………………………………………………………………………………………………..13

3.1 Application of correlation matrix…………………………………………………………………..13

4. Missing Value Treatment…………………………………………………………………………………………..14

4.1 Missing Values………………………………………………………………………………………………15

4.2 Handling of Missing Values……………………………………………………………………………15

4.3 Methods of missing data treatment………………………………………………………………15

5. Outlier Treatment……………………………………………………………………………………………………..17

5.1 What is outlier………………………………………………………………………………………………17

5.2 Impact of outlier……………………………………………………………………………………………17

5.3 Removing outlier…………………………………………………………………………………………..17

6. Feature engineering…………………………………………………………………………………………………..17

6.1 Feature engineering………………………………………………………………………………………17

6.2 Feature Selection …………………………………………………………………………………………17

6.3 Feature Scaling…………………………………………………………………………………………….17

7. Models for implementation….…………………………………………………………………………………..30

7.1 Machine Learning………………………………………………………………………………………….23

7.1.1 Supervised Learning……………………………………………………………………………..23

7.1.2 Unsupervised Learning…………………………………………………………………………24

7.2 Linear Regression………………………………………………………………………………………….24

7.2.1 Regression Line………………………………………………………………………………..25

7.3 Prediction Models…………………………………………………………………………………………26

7.3.1 Multivariate Linear Regression………………………………………………………..26

7.3.2 Radom Forest…………………………………………………………………………………..27

7.3.2.1 Feature Importance…………………………………………………………….28

7.3.2.2 Importance of all features in ascending order………………………30

8. Performance metrics…………………………………………………………………………………………………30

8.1 Mean Squared Error(MSE)…………………………………………………………………………….30

8.2 Root Mean Squared Error(RMSE)………………………………………………………………….32

8.3 Mean Absolute Percentage Error(MAPE)………………………………………………………32

9. Outcomes of the problem………………………………………………………………………………………….33

10. Optimization……………………………………………………………………………………………………………..36

10.1 Gradient descent………………………………………………………………………………………….36

10.1.1 What is gradient ?................................................................................36

10.1.2 What is Learning Rate ?.......................................................................37

10.1.3 Algorithm…………………………………………………………………………………………38

10.1.4 Cost function……………………………………………………………………………………38

10.2 Stochastic Gradient Descent…………………………………………………………………………38

**Future Scope………………………………………………………………………………………………………….40**

**Conclusion……………………………………………………………………………………………………………..41**

**Refrences……………………………………………………………………………………………………………….42**

**Introduction**

Problem Statement:

To predict the selling price of 1458 houses on the basis of 81 features of given in the dataset.

Objective:

The motivation in doing such analysis is to provide real estate firms almost accurate information about expected values of properties. 2008 Housing financial crisis was a huge jolt-back to the world economy which exposed us to various challenges that are still available and need to be dealt with. As we all know that the crisis was caused due to the bubble created in the real estate Industry due to very easy availability of the credit. It was used as an eye opener for the world and we began to think about predicting household prices which not only helps the financial institution to estimate and provide credit on a real estate but also it is huge negotiation point for the customer who are going to buy a new house.

Organization:

In this project, the dataset is given for 1458 houses having 81 features. This data is to use for training the model. After training, the model will predict the outcomes for those data whose actual outcomes are known. This helps in checking the accuracy of the model and it also helps in checking if there is any transformation needed to do on the variables. After this step if there is any possibility of getting better accuracy by doing transformation then some transformations like sqrt, log, etc has to be applied on variables.

After this cross validation is the next step in which a part of data will be reserved and won’t be used in training of the model. This part of data will be used for testing purpose to check the accuracy of the model on new data on which they have not been trained.

In Next step, different model will be trained on training data and tested on the validation data and their accuracy will be compared.

**Chapter 1**

**Introduction**

Investment is a business activity that most people are interested in this globalization era. There are several objects that are often used for investment, for example, gold, stocks and property. In particular, property investment has increased significantly since 2011, both on demand and property selling. One of the increasing of property demand is because of high population. The result of this census indicates that the younger generation will need a house or buy a house in the future. Based on preliminary research conducted, there are two standards of house price which are valid in buying and selling transaction of a house that is house price based on the developer (market selling price). In determining the price of home, the developer must calculate carefully and determine the appropriate method because property prices always increase continuously and almost never fall in the long term or short.

There are several approaches that can be used to determine the price of the house, one of them is the prediction analysis. The first approach is regression analysis, in this I am using multivariate linear regression algorithm. In which I have trained the model and then applied log transformation on dependent variable for getting better result and optimized the model using Batch Gradient Descent Algorithm. Another approach is random forest in which I have used random forest as a regression model. And then calculated the error metrics of both models for comparison.

This research aims to create a house price prediction model using regression and BGD to obtain optimal prediction results and random forest to compare the result and select the more accurate model.

**Chapter 2**

**Data Exploration and Preparation**

1. **Understanding data**

There are 81 columns in the dataset i.e., 81 data fields. It’s very necessary to interpret the variables in the dataset.

* 1. **Data fields**

Here's a brief version of what is in the dataset being used in this project.

* **SalePrice** - The property's sale price in dollars. This is the target variable.
* **MSSubClass**: The building class
* **MSZoning**: The general zoning classification
* **LotFrontage**: Linear feet of street connected to property
* **LotArea**: Lot size in square feet
* **Street**: Type of road access
* **Alley**: Type of alley access
* **LotShape**: General shape of property
* **LandContour**: Flatness of the property
* **Utilities**: Type of utilities available
* **LotConfig**: Lot configuration
* **LandSlope**: Slope of property
* **Neighborhood**: Physical locations within Ames city limits
* **Condition1**: Proximity to main road or railroad
* **Condition2**: Proximity to main road or railroad (if a second is present)
* **BldgType**: Type of dwelling
* **HouseStyle**: Style of dwelling
* **OverallQual**: Overall material and finish quality
* **OverallCond**: Overall condition rating
* **YearBuilt**: Original construction date
* **YearRemodAdd**: Remodel date
* **RoofStyle**: Type of roof
* **RoofMatl**: Roof material
* **Exterior1st**: Exterior covering on house
* **Exterior2nd**: Exterior covering on house (if more than one material)
* **MasVnrType**: Masonry veneer type
* **MasVnrArea**: Masonry veneer area in square feet
* **ExterQual**: Exterior material quality
* **ExterCond**: Present condition of the material on the exterior
* **Foundation**: Type of foundation
* **BsmtQual**: Height of the basement
* **BsmtCond**: General condition of the basement
* **BsmtExposure**: Walkout or garden level basement walls
* **BsmtFinType1**: Quality of basement finished area
* **BsmtFinSF1**: Type 1 finished square feet
* **BsmtFinType2**: Quality of second finished area (if present)
* **BsmtFinSF2**: Type 2 finished square feet
* **BsmtUnfSF**: Unfinished square feet of basement area
* **TotalBsmtSF**: Total square feet of basement area
* **Heating**: Type of heating
* **HeatingQC**: Heating quality and condition
* **CentralAir**: Central air conditioning
* **Electrical**: Electrical system
* **1stFlrSF**: First Floor square feet
* **2ndFlrSF**: Second floor square feet
* **LowQualFinSF**: Low quality finished square feet (all floors)
* **GrLivArea**: Above grade (ground) living area square feet
* **BsmtFullBath**: Basement full bathrooms
* **BsmtHalfBath**: Basement half bathrooms
* **FullBath**: Full bathrooms above grade
* **HalfBath**: Half baths above grade
* **Bedroom**: Number of bedrooms above basement level
* **Kitchen**: Number of kitchens
* **KitchenQual**: Kitchen quality
* **TotRmsAbvGrd**: Total rooms above grade (does not include bathrooms)
* **Functional**: Home functionality rating
* **Fireplaces**: Number of fireplaces
* **FireplaceQu**: Fireplace quality
* **GarageType**: Garage location
* **GarageYrBlt**: Year garage was built
* **GarageFinish**: Interior finish of the garage
* **GarageCars**: Size of garage in car capacity
* **GarageArea**: Size of garage in square feet
* **GarageQual**: Garage quality
* **GarageCond**: Garage condition
* **PavedDrive**: Paved driveway
* **WoodDeckSF**: Wood deck area in square feet
* **OpenPorchSF**: Open porch area in square feet
* **EnclosedPorch**: Enclosed porch area in square feet
* **3SsnPorch**: Three season porch area in square feet
* **ScreenPorch**: Screen porch area in square feet
* **PoolArea**: Pool area in square feet
* **PoolQC**: Pool quality
* **Fence**: Fence quality
* **MiscFeature**: Miscellaneous feature not covered in other categories
* **MiscVal**: $Value of miscellaneous feature
* **MoSold**: Month Sold
* **YrSold**: Year Sold
* **SaleType**: Type of sale
* **SaleCondition**: Condition of sale

**2.2 Variable Identification**

The first part would be to identify the Predictor and Target Variable and after this we need to identify the data type and category

Variable Category

**Continuous**

SalePrice

GrdLivArea

**Categorical**

YearBuilt

KitchenQual, etc

Type of Variable

**Predictor Variable**

YearBuilt

GrgArea

GrdLivArea

HomeStyle, etc

**Target Variable**

SalePrice

Data Type

**Character**

HomeStyle

GarageType, etc

**Numeric**

GrdLivArea

BedRooms, etc

**2.2 Univariate Analysis**

As the name suggest we first need to analyse each variables separately. Based on the Variable category we can choose different methods to do the same.

* + 1. **Categorical Variable**: - Frequency table to understand the distribution, so basically there are two measures we can do for the same, count of the variable and Percentage count of the variable.
    2. **Continuous Variable**: - In this case we need to understand the central tendency and the spread of the variable.

|  |  |  |
| --- | --- | --- |
| Central Tendency | Measure of Dispersion | Visualization Method |
| Mean | Range | Box Plot |
| Median | Quartile | Histogram |
| Mode | Inter Quartile Range | Pie Charts |
| Min | Variance | Frequency Polygons |
| Max | Standard Deviation |  |
|  | Skewness and Kurtosis |  |

**2.3 Bivariate Analysis**

As the name suggest we first need to analyse each variable’s relationship with other variables. Based on the Variable category we can choose different methods. The most common method is correlation matrix which is explained in the next chapter.

**Chapter 3**

**Correlation matrix**

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used as a way to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.

The presence of correlation in error terms drastically reduces model’s accuracy. This usually occurs in time series models where the next instant is dependent on previous instant. If the error terms are correlated, the estimated standard errors tend to underestimate the true standard error.

**3.1 Applications of a correlation matrix**

There are three broad reasons for computing a correlation matrix.

1. To summarize a large amount of data where the goal is to see patterns. In our example above, the observable pattern is that all the variables highly correlate with each other.
2. To input into other analyses. For example, people commonly use correlation matrixes as inputs for exploratory factor analysis, confirmatory factor analysis, structural equation models, and linear regression when excluding missing values pairwise.
3. As a diagnostic when checking other analyses. For example, with linear regression a high amount of correlations suggests that the linear regression’s estimates will be unreliable.



We can see in the above plot that TotRmsAbvGrd and GrLivArea, GarageYrBlt and YearBlt are highly correlated with each other

**Chapter 4**

**Missing Value Treatment**

**4.1 Missing data**

The concept of missing values is important to understand in order to successfully [manage](http://www.statisticssolutions.com/academic-solutions/resources/dissertation-resources/data-entry-and-management/multiple-imputation-for-missing-data/) data.  If the missing values are not handled properly by the researcher, then he/she may end up drawing an inaccurate inference about the data.  Due to improper handling, the result obtained by the researcher will differ from ones where the missing values are present.

Missing data in the dataset can reduce the accuracy of a model or lead to a biased model because we have not analysed the behaviour and the relationship with the other variables correctly.

**4.2 Handling Missing Values**

The researcher may leave the data or do [data imputation](http://www.statisticssolutions.com/academic-solutions/resources/dissertation-resources/data-entry-and-management/multiple-imputation-for-missing-data/) to replace them. Suppose the number of cases of missing values is extremely small; then, an expert researcher may drop or omit those values from the analysis.  In statistical language, if the number of the cases is less than 5% of the sample, then the researcher can drop them.

In the case of multivariate analysis, if there is a larger number of missing values, then it can be better to drop those cases (rather than do imputation) and replace them.  On the other hand, in univariate analysis, imputation can decrease the amount of bias in the data, if the values are missing at random.

**4.3 Methods of Missing Data Treatment**

1. Deletion
   1. List Wise Deletion: we delete the observation where any of the variable is missing. This is the simplest method of missing value treatment, but it reduces the data and may lead to issues discussed above
   2. Pair Wise Deletion: In this case we analyse all the available cases and we skip the missing values from the analysis of column. This keeps the data intact although the number of observation changes in each column.
2. Mean/Median/Mode Imputation: Imputation is a method to replace the missing value with the estimated values. The main objective is to employ known relationships that can be used to fill up the missing values.
   1. Mean/Median is used for continuous quantitative attributes of the data
   2. Mode is used for qualitative attributes of the data
   3. The imputation can be of two types:
      1. Generalized Imputation: In this case we replace the missing value with the Mean/Median of the all the values present in the column
      2. Similar case Imputation: In this case we replace the missing value with mean/median based on another categorical variable

In our dataset, we have replaced missing data for a set of features NA means "No" or "0" ,the missing value of numeric values with 0, categorical values with mode.

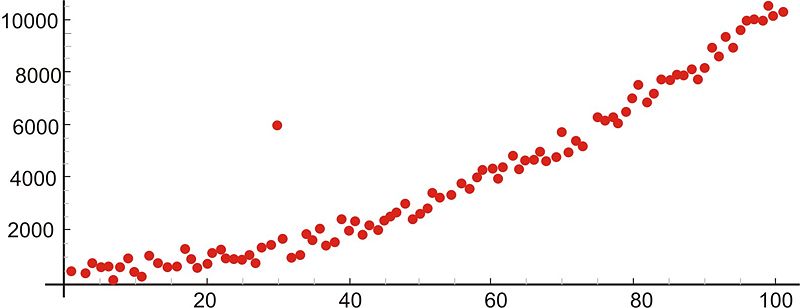
**Chapter 5**

**Outlier Treatment**

**5.1 What is Outlier?**

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Before abnormal observations can be singled out, it is necessary to characterize normal observations.

Outliers are the observation that are very far away from the general population of the attribute. There are two types of outliers: Multivariate and Univariate Outliers as the name suggest Multivariate outliers are those present in a single attribute of the dataset and Multivariate outliers are something that are present in more than one or more attributes of the outlier. Below shows the behaviour of outlier:



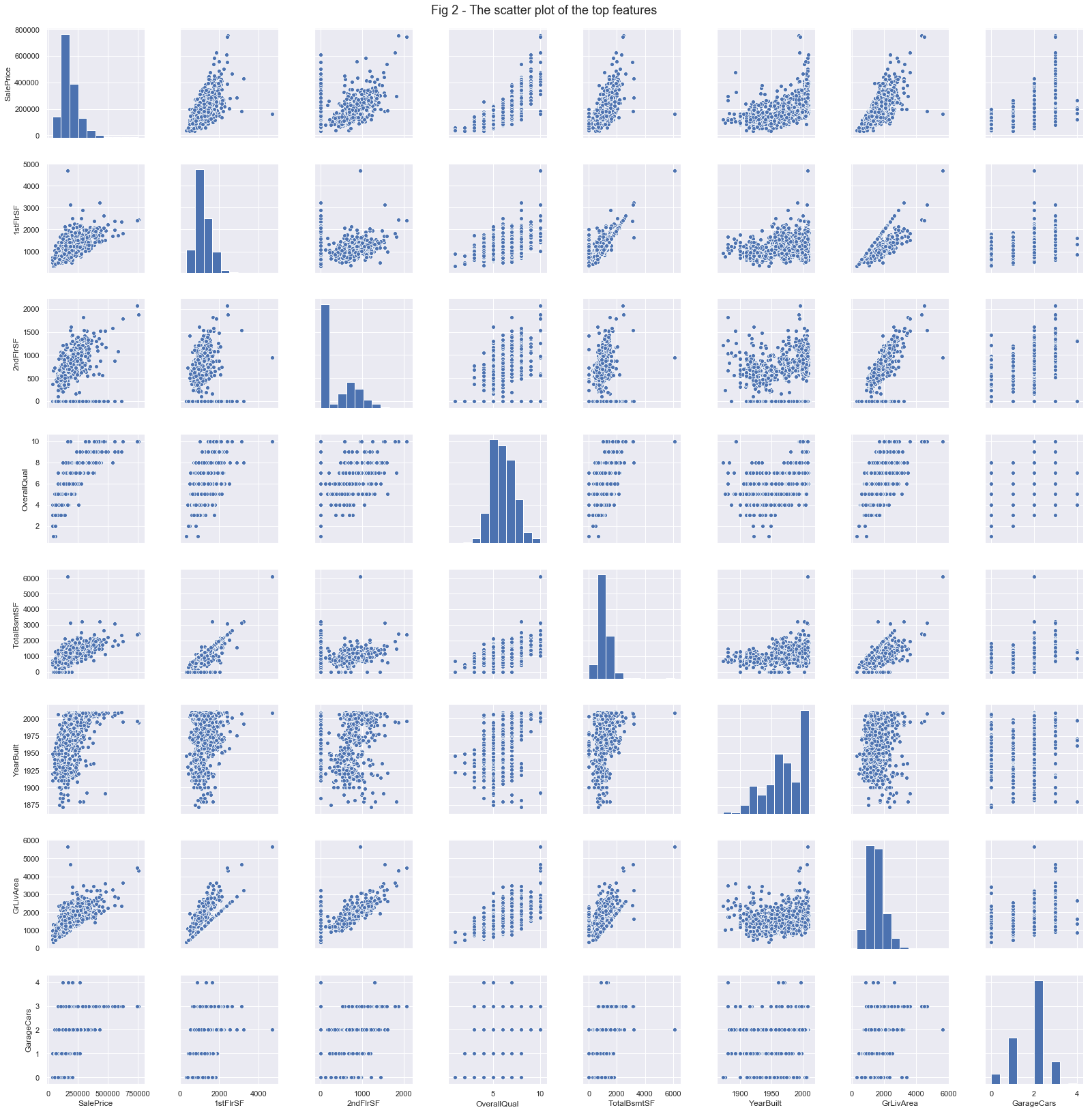
**5.2 Impact of Outliers**

* 1. It increases the error variance
  2. If they are non-randomly distributed, they can decrease normality
  3. They Bias datasets or influence the estimates for an attribute
  4. Impact the basic assumptions of the statistical models

**5.3 Removing Outliers**

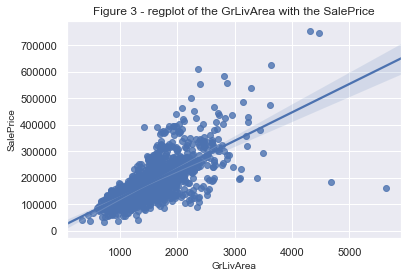
* 1. Deleting Observation: We delete the observations with outlier values, we generally trim the values from both end of the attribute
  2. Transforming and Binning: Transformation of the variable can also help to remove the outliers from the attributes. There are various methods of transforming the data .Some of the examples of transformations are
     1. Create variables based on Date and Time or difference in date and time
     2. Create new ratios and proportions
     3. Standard transformations like log, quadratic, exponential etc.
  3. Imputing Outliers: Like the missing value treatment we can impute the outliers based on the same techniques we discussed in Missing value treatment.
  4. Treat Separately: If we have more number so outliers in our dataset then we might need to treat them separately in another statistical model.

For detecting outliers in our dataset I have created scatterplot betwwen each features.



From the above regression plots we can see that there are outliers in GrdLivArea, 1stFlrSF and TotalBsmtSF. For checking that we made another scatter plot between these features and target variable.

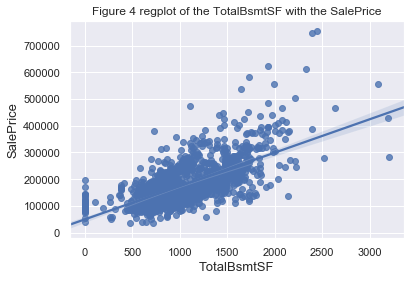
1. Outlier testing of GrdLivArea



We can see in the regression plot there are two outlier in GrdLivArea, there is a linear relation between GrdLivArea and SalePrice, the GrdLivArea should increase with increase in SalePrice. But the two largest values of GrdLivArea are showing opposite behaviour so we need to remove them.

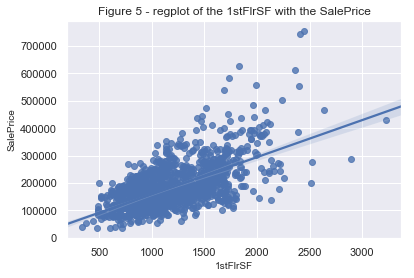
For removing this , first we need to sort the GrdLivArea in ascending order and then drop those two row from the dataset.

1. Outlier testing of TotalBsmtSF



In the above plot, there are no obvious outliers in TotalBsmtSF.

1. Outlier testing of 1stFlrSF



In the above, there is no obvious outlier in 1stFlrSF.

**Chapter 6**

**Feature engineering**

**6.1 Feature Engineering**

Feature engineering is an essential part of building any intelligent system. Each problem is domain specific and better features are often the deciding factor of the performance of your system. Feature Engineering is an art as well as a science and this is the reason Data Scientists often spend 70% of their time in the data preparation phase before modelling.

Feature engineering is the process of using **domain knowledge** of the data to create features that make machine learning algorithms work. If feature engineering is done correctly, it increases the predictive power of machine learning algorithms by creating features from raw data that help facilitate the machine learning process.

**6.2 Feature Selection**

Feature selection is the task of trying to discover the smallest set of features highly correlated with the dependent variable. It is important for the interpretability of the model but also to get a better fit, and consequently a better performance. We employed an automatic feature selection technique using a tree-based learning algorithm, and then used the tree structure produced to select the best features

**6.3 Feature Scaling**

Our last step in the pre-processing phase would be standardizing the data. This will be useful for all the models. As we are using cross-validation, the scaling has to be done independently for the training and the testing sets.

**Chapter 7**

**Models for Implementation**

This project uses two machine learning algorithms. There are various algorithms, based on their abilities to handle regression analysis and their appearances. The best performing one is Random Forest which is explained in details in the subsections.

**7.1 Machine Learning**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. **The primary aim is to allow the computers learn automatically** without human intervention or assistance and adjust actions accordingly.

* + 1. **Supervised learning**

Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

Y = f(X)

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

Supervised learning problems can be further grouped into regression and classification problems.

1. **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
2. **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.
   * 1. **Unsupervised learning**

Unsupervised learning is where you only have input data (X) and no corresponding output variables.

The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data.

Unsupervised learning problems can be further grouped into clustering and association problems.

1. **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behaviour.
2. **Association**:  An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

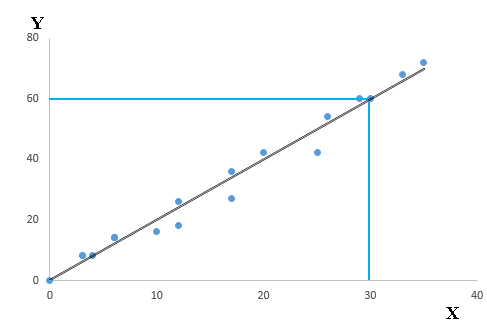
**7.2 Linear Regression**

Linear Regression is a supervised learning algorithm. It is used for predictive analysis. It is a technique which explains the degree of relationship between two or more variables (multiple regression, in that case) using a best fit line / plane. Simple Linear Regression is used when we have, one independent variable and one dependent variable.

Regression technique tries to fit a single line through a scatter plot (see below).  The simplest form of regression with one dependent and one independent variable is defined by the formula:

Y = aX + b

Let’s understand this equation using the scatter plot below:

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/10/Linear2.png)

Above, you can see that a black line passes through the data points. Now, you carefully notice that this line intersects the data points at coordinates (0,0), (4,8) and (30,60). Here’s a question. Find the equation that describe this line? Your answer should be:

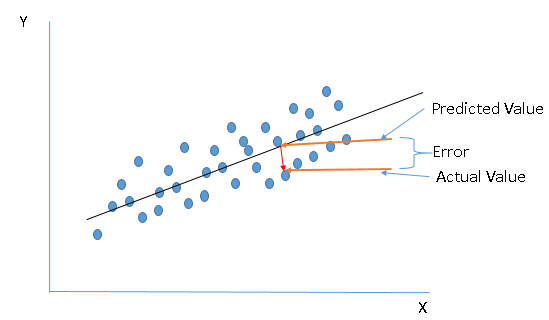
Y= a \* X + b

This equation is known as linear regression equation, where Y is target variable, X is input variable. ‘a’ is known as slope and ‘b’ as intercept. It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on input variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a \*X + b.

To choose the best fit line or the coefficients we will fit a regression line.

**7.2.1 Regression Line**

Regression line establishes a relationship between independent and dependent variable(s). A line which can explain the relationship better is said to be best fit line.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/10/Linear4.png)

**7.3 Prediction Models**

**7.3.1 Multivariate linear regression**

Let’s now examine the process to deal with **multiple independent variables**related to a dependent variable.

Once you have identified the level of significance between independent variables(IV) and dependent variables(DV), use these significant IVs to make more powerful and accurate predictions. This technique is known as “Multi-variate Regression”.

In a multiple regression model, we try to predict

[Linear15](https://www.analyticsvidhya.com/wp-content/uploads/2015/10/Linear15.png)

Here, b1, b2, b3 …bk are slopes for each independent variables X1, X2, X3….Xk and a is intercept.

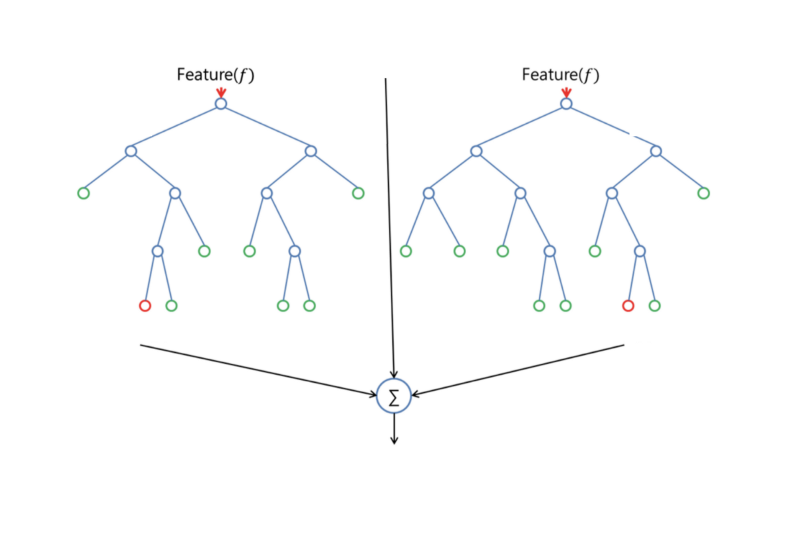
Things get much more complicated when your multiple independent variables are related to with each other.

To deal with that there are five assumptions of linear regression which need to be satisfied to use regression model. The assumptions are:

1. Linearity – There must be linear relation between dependent and independent variables.
2. Multivariate normality – Linear regression requires all variable to be multivariate normal.
3. Homoscedasticity – The presence of constant variance in the error terms results in homoscedasticity. The residuals must be equal across the regression line.
4. No or little multicollinearity – Multicollearity occurs when the independent variables are too highly correlated with each other.
5. No or little autocorrelation – Autocorrelation occurs when the residual are not independent from each other.
   * 1. **Random Forest**

Random Forest is a supervised learning algorithm. Like you can already see from it’s name, it creates a forest and makes it somehow random. The „forest“ it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

It operates by constructing a multitude of decision trees to fit the observations into groups based on their attributes’ values and outputs the mean prediction of the individual trees. As the name suggests, “decision tree” model builds a reversed tree-like structure, where the “root” is at the top, followed by multiple branches, nodes and leaves. The end of each branch is a decision leaf, which is the model’s predicted value, given the values of the attributes represented by the path from the root node to the said decision leaf.



One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node.

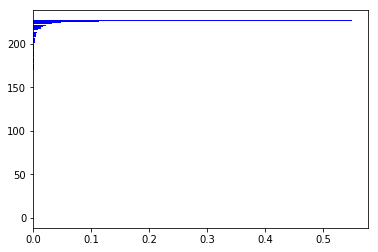
**7.3.2.1 Feature Importance:**

Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a great tool for this, that measures a features importance by looking at how much the tree nodes, which use that feature, reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results, so that the sum of all importance is equal to 1.

If you don’t know how a decision tree works and if you don’t know what a leaf or node is, here is a good description from Wikipedia: In a decision tree each internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). A node that has no children is a leaf.

Through looking at the feature importance, you can decide which features you may want to drop, because they don’t contribute enough or nothing to the prediction process. This is important, because a general rule in machine learning is that the more features you have, the more likely your model will suffer from overfitting and vice versa.

Let’s plot the bar graph of the features in ascending order of their importance.



**7.3.2.2 Importance of all features in ascending order**

1. Variable: OverallQual Importance: 0.55
2. Variable: GrLivArea Importance: 0.11
3. Variable: TotalBsmtSF Importance: 0.05
4. Variable: GarageCars Importance: 0.03
5. Variable: GarageArea Importance: 0.03
6. Variable: YearBuilt Importance: 0.02
7. Variable: BsmtFinSF1 Importance: 0.02
8. Variable: 1stFlrSF Importance: 0.02
9. Variable: LotArea Importance: 0.01
10. Variable: OverallCond Importance: 0.01
11. Variable: YearRemodAdd Importance: 0.01
12. Variable: BsmtUnfSF Importance: 0.01
13. Variable: FireplaceQu Importance: 0.01
14. Variable: GarageYrBlt Importance: 0.01
15. Variable: GarageFinish Importance: 0.01
16. Variable: CentralAir\_N Importance: 0.01
17. Variable: CentralAir\_Y Importance: 0.01
18. Variable: index Importance: 0.0
19. Variable: Id Importance: 0.0
20. Variable: LotFrontage Importance: 0.0
21. Variable: LotShape Importance: 0.0
22. Variable: Utilities Importance: 0.0
23. Variable: LandSlope Importance: 0.0
24. Variable: MasVnrArea Importance: 0.0
25. Variable: ExterQual Importance: 0.0
26. Variable: ExterCond Importance: 0.0
27. Variable: BsmtQual Importance: 0.0
28. Variable: BsmtCond Importance: 0.0
29. Variable: BsmtFinSF2 Importance: 0.0
30. Variable: HeatingQC Importance: 0.0
31. Variable: 2ndFlrSF Importance: 0.0
32. Variable: LowQualFinSF Importance: 0.0
33. Variable: BsmtFullBath Importance: 0.0
34. Variable: BsmtHalfBath Importance: 0.0
35. Variable: FullBath Importance: 0.0
36. Variable: HalfBath Importance: 0.0
37. Variable: BedroomAbvGr Importance: 0.0
38. Variable: KitchenAbvGr Importance: 0.0
39. Variable: KitchenQual Importance: 0.0
40. Variable: TotRmsAbvGrd Importance: 0.0 ……..and so on.

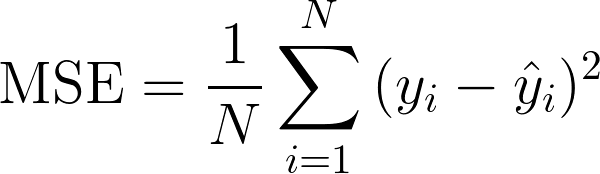
**Chapter 8**

**Performance metrics**

The model developed in this project will be tested using several methods such as Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE).

**8.1 Mean Squared Error(MSE)**

MSE basically measures average squared error of our predictions. For each point, it calculates square difference between the predictions and the target and then average those values.



The higher this value, the worse the model is. It is never negative, since we’re squaring the individual prediction-wise errors before summing them, but would be zero for a perfect model.

**Advantage:**

Useful if we have unexpected values that we should care about. Very high or low value that we should pay attention.

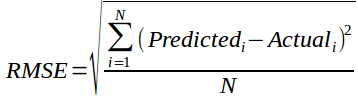
**Disadvantage:**

If we make a single very bad prediction, the squaring will make the error even worse and it may skew the metric towards overestimating the model’s badness. That is a particularly problematic behaviour if we have noisy data (that is, data that for whatever reason is not entirely reliable) — even a “perfect” model may have a high MSE in that situation, so it becomes hard to judge how well the model is performing. On the other hand, if all the errors are small, or rather, smaller than 1, than the opposite effect is felt: we may underestimate the model’s badness.

**Note that** if we want to have a constant prediction the best one will be the **mean value of the target values.**It can be found by setting the derivative of our total error with respect to that constant to zero, and find it from this equation.

**8.2 Root Mean Squared error(RMSE)**

RMSE is the most popular evaluation metric used in regression problems. It follows an assumption that error are unbiased and follow a normal distribution. RMSE metric is given by:



RMSE is just the square root of MSE. The square root is introduced to make scale of the errors to be the same as the scale of targets.

Now, it is very important to understand in what sense RMSE is similar to MSE,and what is the difference.

First, they are similar in terms of their minimizers, every minimizer of MSE is also a minimizer for RMSE and vice versa since the square root is an non-decreasing function. For example, if we have two sets of predictions, A and B, and say MSE of A is greater than MSE of B, then we can be sure that RMSE of A is greater RMSE of B. And it also works in the opposite direction.

**8.3 Mean Absolute Percentage Error**

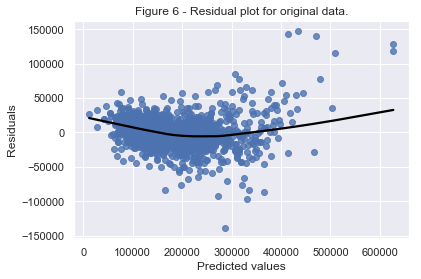
MAPE is calculated by making an average percentage of the absolute error of each predicted result. Thus, MAPE can indicate how much prediction error. The **mean absolute percentage error** (**MAPE**), also known as **mean absolute percentage deviation** (**MAPD**), is a measure of prediction accuracy of a forecasting method.

\[ \mathrm{MAPE}=\frac{100}{N}\times \sum_{i=1}^N \left | \frac{x_i - \hat x_i}{x_i} \right | \]

**Chapter 9**

**Outcomes of the problem**

1. **The result of the first case in which we have used multivariate regression model is given below:**



**For this model, the value of performance metrics are given below:**

Mean squared error: 487191610.97229356

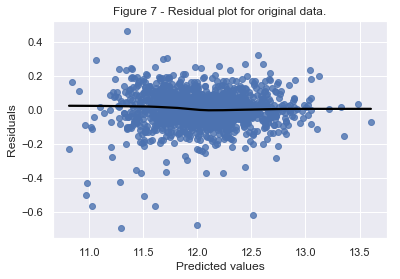
**Root mean squared error: 22072.417424747422**

**Mean Absolute Error: 15369.9 degrees.**

**Accuracy: 90.72 %.**

**In above case, the accuracy of model is very low which can be improved by applying transformations to the variables.**

1. **In the next case of this project, we have applied log transformation in the target variable. The result of this is given below:**



**For this case, the value of performance metrics are given below:**

**Mean squared error: 0.009871227911656502**

**Root mean squared error: 0.09935405332273314**

**Mean Absolute Error: 0.07 degrees.**

**Accuracy: 99.42 %.**

1. **Then in third case random forest algorithm is used for regression. Firstly, all the features has taken for feature engineering and calculated the accuracy of the model. The performance metrics of this case is given below:**

**Mean squared error: 0.002507678537244841**

**Root mean squared error: 0.05007672650288596**

**Mean Absolute Error: 0.03 degrees.**

**Accuracy: 99.71 %.**

1. **Then by sorting the important features in ascending, only the features having more importance is taken for feature engineering in random forest regression. And the result in the both cases are same.**

**Mean squared error: 0.0024355418747356735**

**Root mean squared error: 0.049351209455652384**

**Mean Absolute Error: 0.03 degrees.**

**Accuracy: 99.72 %.**

* **From above evaluation of models, Random Forest is giving the best accuracy.**

**Chapter 10**

**Optimization**

Optimization is a big part of machine learning. Almost every machine learning algorithm has an optimization algorithm at it’s core. It refers to the task of minimizing/maximizing an objective function *f(x)* parameterized by *x*. In machine learning, it’s the task of minimizing the cost/loss function *J(w)* parameterized by the model’s parameters. Optimization algorithms (in case of minimization) have one of the following goals:

1. Find the global minimum of the objective function. This is feasible if the objective function is convex, i.e. any local minimum is a global minimum.
2. Find the lowest possible value of the objective function within its neighbourhood. That’s usually the case if the objective function is not convex as the case in most deep learning problems.

**10.1 Gradient descent**

It is an optimization algorithm used in training a model. In simple words, Gradient Descent finds the parameters that minimize the cost function (error in prediction). Gradient Descent does this by iteratively moves toward a set of parameter values that minimize the function, taking steps in the opposite direction of the gradient.

**10.1.1 What is a Gradient?**

A [gradient](https://en.wikipedia.org/wiki/Gradient) is a vector-valued function that represents the slope of the tangent of the graph of the function, pointing the direction of the greatest rate of increase of the function. It is a derivative that indicates the incline or the slope of the cost function.

**In this project for optimization gradient descent algorithm is used.** The evaluation of how close a fit a machine learning model estimates the target function can be calculated a number of different ways, often specific to the machine learning algorithm.

That is exactly what Gradient Descent does. Its goal is to reach the lowest point of the mountain. The mountain is the data plotted in a space, the size of the step you move is the learning rate, feeling the incline around you and decide which is higher is calculating the gradient of a set of parameter values, which is done iteratively. The chosen direction is where the cost function reduces (the opposite direction of the gradient). The lowest point in the mountain is the value -or weights- where the cost of the function reached its minimum (the parameters where our model presents more accuracy).

**10.1.2 What is the Learning rate?**

Like we said, the gradient is a vector-valued function, and as a vector, it has both a direction and a magnitude. The Gradient descent algorithm multiplies the gradient by a number (Learning rate or Step size) to determine the next point.

For example: having a gradient with a magnitude of 4.2 and a learning rate of 0.01, then the gradient descent algorithm will pick the next point 0.042 away from the previous point.

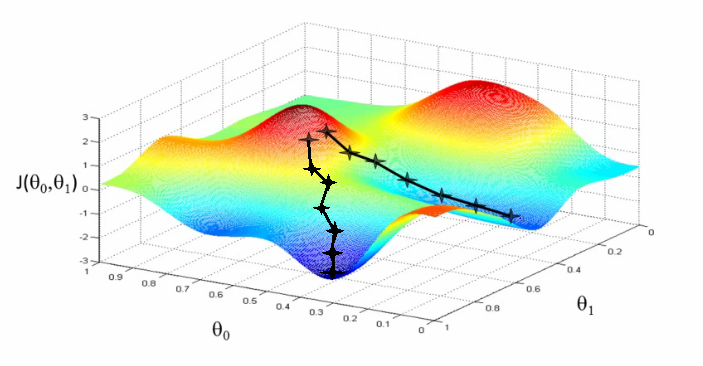


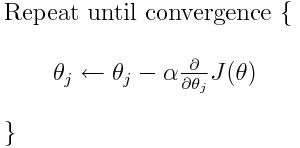
Figure: Gradient descent

**10.1.3 Algorithm**

Repeat until hit convergence:

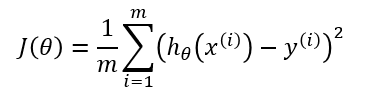
1. Given the gradient, calculate the change in the parameters with the learning rate.
2. Re-calculate the new gradient with the new value of the parameter.
3. Repeat step 1.

Here is the formula of gradient descent algorithm:

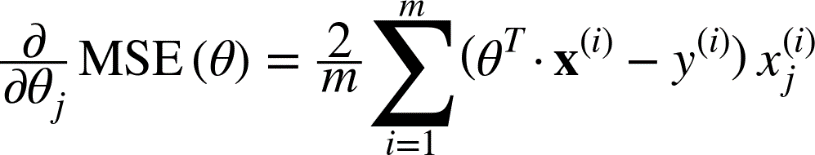


**10.1.4 Cost function**

The cost function involves evaluating the coefficients in the machine learning model by calculating a prediction for the model for each training instance in the dataset and comparing the predictions to the actual output values and calculating a sum or average error. The cost function is:



From the cost function a derivative can be calculated for each coefficient so that it can be updated using exactly the update equation given below:



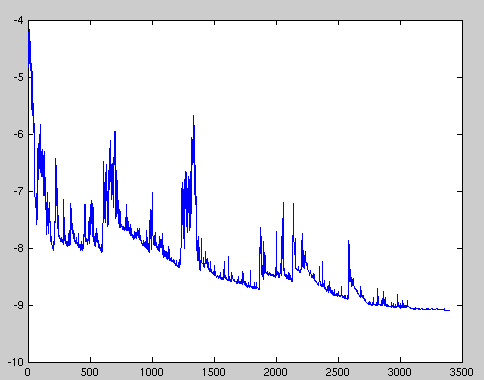
**10.2 Stochastic gradient descent**

It is the most common form of gradient descent described in machine learning. we should apply **Stochastic Gradient Descent (SGD)**, a simple modification to the standard gradient descent algorithm that computes the gradient and updates our weight matrix W on **small batches of training data**, rather than the entire training set itself.

While this leads to “noiser” weight updates, it also allows us to take more steps along the gradient (1 step for each batch versus 1 step per epoch), ultimately leading to faster convergence and no negative affects to loss and classification accuracy.

Stochastic gradient descent (SGD) performs a parameter update for each observation. So instead of looping over each observation, it just needs one to perform the parameter update. SGD is usually faster than batch gradient descent, but its frequent updates cause a higher variance in the error rate, that can sometimes jump around instead of decreasing.

To represent this graphically, notice the below graph.



**Figure: Cost function in each iteration of stochastic gradient descent**

**Future Scope**

1. **The Banking System can use this to estimate the credit line on the property.**
2. **The Insurance Industry can use this analysis to properly evaluate the real estate.**
3. **The Middle man brokers can be kept in line as the expected price range is known.**
4. **The customers will become more educated on buying a new property.**
5. **The Government can control the real estate economy if something seems off.**

**Conclusion**

In this paper, several tests have been performed using linear regression, random forest and gradient descent optimization methods to perform house price prediction. Based on the data of 1458 houses, the system is modelling house price predictions into 7 models each of them represents one area. Using different methods that match the time-series data will be used in the future research to obtain smaller error prediction values and using more data to get the better result.

**REFERENCES**

1. Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences – 12 Sep 2002, Kindle Edition by **Jacob Cohen, Patricia Cohen, Stephen G. West** and **Leona S. Aiken**.
2. Modern Multivariate Statistical Techniques - 2008 by **Izenman**, **Alan J**.
3. An Introduction to Statistics with Python: With Applications in the Life Sciences (Statistics and Computing) 1st ed. 2016 Edition, Kindle Edition by [**Thomas Haslwanter**](https://www.amazon.in/Thomas-Haslwanter/e/B07BH56Q1R/ref=dp_byline_cont_ebooks_1).
4. Machine Learning With Random Forests And Decision Trees: A Visual Guide For Beginners Kindle Edition by [**Scott Hartshorn**](https://www.amazon.in/Scott-Hartshorn/e/B00NZ7S1AG/ref=dp_byline_cont_ebooks_1)
5. Introduction to Linear Regression Analysis – Student Edition, 16 Dec 2006 by [**Douglas C Montgomery**](https://www.amazon.in/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=Douglas+C+Montgomery&search-alias=stripbooks) **,**[**Elizabeth A Peck**](https://www.amazon.in/s/ref=dp_byline_sr_book_2?ie=UTF8&field-author=Elizabeth+A+Peck&search-alias=stripbooks)**,**[**G.** and **Geoffrey Vining**](https://www.amazon.in/s/ref=dp_byline_sr_book_3?ie=UTF8&field-author=G.+Geoffrey+Vining&search-alias=stripbooks).
6. Hands–On Machine Learning with Scikit–Learn and TensorFlow Paperback – 24 Mar 2017 by [**Aurelien Geron**](https://www.amazon.in/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=Aurelien+Geron&search-alias=stripbooks)**.**
7. Kaggle Dataset from House Prices: Advanced Regression Techniques.

URL: https://www.kaggle.com/c/house-prices-advanced-regression-techniques