# Definition

## Project Overview

The real estate market is one of the most booming and richest markets in the world. With the growing demand for more space (both commercial and residential), developers are having a field time, both in terms of volume and profits. It is not only the fresh inventory that is highly sought, but re-sale value of properties, too, are higher than ever.

With the growing demand and exponentially increasing prices, it has become imperative to study in detail the various factors affecting the increase in the prices. This field of study is crucial to both the parties involved - the owner/user/renter on the demand side and the developer/renovator on the supply side. Here, we will try to predict the prices of residential properties using data from Ames, Iowa, USA.

The dataset, known as the Ames Housing dataset and compiled by Dean De Cook, can be seen as an alternative to the famous Boston Housing dataset.

Having worked on the Boston Housing dataset as part of the Supervised Learning project, this dataset would allow me to explore a similar problem in much more detail and depth.

The Ames Housing data set, too, has been used by a lot of people in their final regression project. Along with rigorous exploratory data analysis, the most common machine learning technique that has been implemented are multiple linear regression models.

Reference - Dean De Cock, Truman State University, Journal of Statistics Education Volume 19, Number 3(2011), www.amstat.org/publications/jse/v19n3/decock.pdf

## Problem Statement

To predict the sale price of residential homes on the basis of 79 variables which describe the quality and quantity of the different attributes of the individual properties.

The data is of different qualitative and quantitative attributes of various residential properties, along with their prices in the market. The idea is to leverage myriad Machine Learning techniques to build a formidable model that can predict the sale prices of the residential properties. The model needs to be able to fit the given data properly as well as predict accurately on unseen data.

The most crucial step here is to first understand the given data properly. Detailed exploratory analysis needs to be done to get to know the data better. The first thing to be done is to discover the relationships between the different variables (attributes) and try to understand their intermingling. Also to be noted are the relationship of the independent variables with the dependent Sales variable.

Once the various inter-relationships between the variables become clearer, feature engineering (if any required) needs to be carried on. Different methods may needed to be applied on the different types of variables (numeric/categorical).

After the data has been cleaned and pre-processed, Machine Learning techniques will be applied in a way deemed suitable. The models will then be tested for performance and the results will be cross-validated. The final model will then be selected, this will be the model that best predicts the Sale prices of the residential properties.

## Metrics

This being a regression problem with a continuous target variable, the following performance metric was used -

- **Root Mean Squared Error (RMSE)** = $\sqrt{((\sum(y_i - \hat{y}_i)^2)/n)}$
  where $y_i$: the actual target values
  $\hat{y}_i$: the predicted target values
  n: number of observations

The RMSE is the square root of the squared sum of the difference in the values of the actual and the predicted target variables divided by the total number of observations.
The RMSE serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSE is a good measure of accuracy to compare the forecasting errors of different models for a particular variable.

# Analysis

## Data Exploration

The Ames dataset has 1460 rows. Each row has 81 columns describing the qualitative and quantitative attributes of residential properties in Ames, Iowa, USA.

The data fields are as follows -

**Target Variable:**

- **SalePrice** - the property's sale price in dollars

**Predictor Variables:**

- **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

Out of these 79 (predictor) variables, 43 are categorical and the 37 are numeric features.

The 43 categorical variables had missing value in 16 of the features and the 37 numeric variables had missing values in 3 of the features.

Most of the numeric variables were positively skewed (skewed to the right).

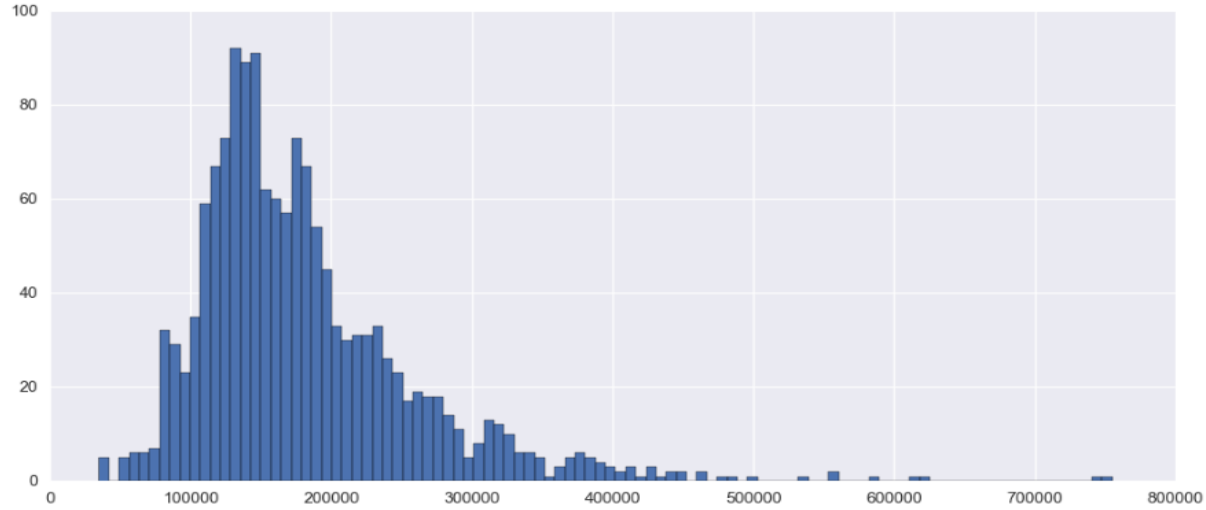
The numeric target variables, 'SalePrice' was also positively skewed.

# Exploratory Visualization

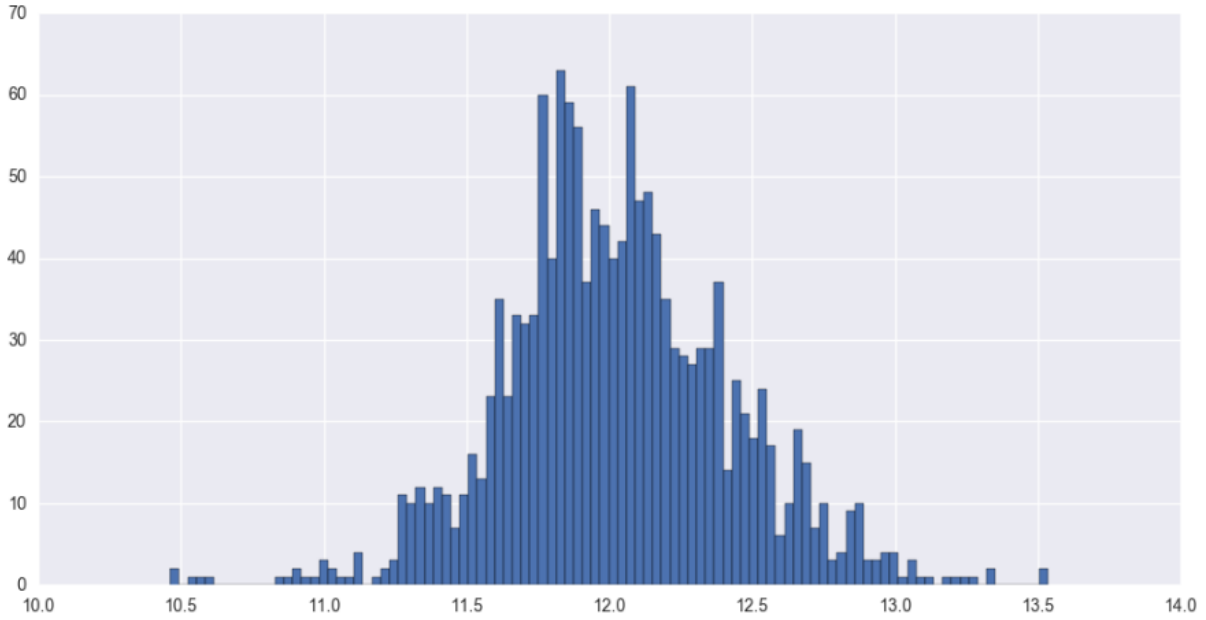
The plot (Fig. 1) below shows the distribution of the of the continuous **target variable, 'SalePrice'**.

The distribution is positively skewed, with the mean > median > mode.

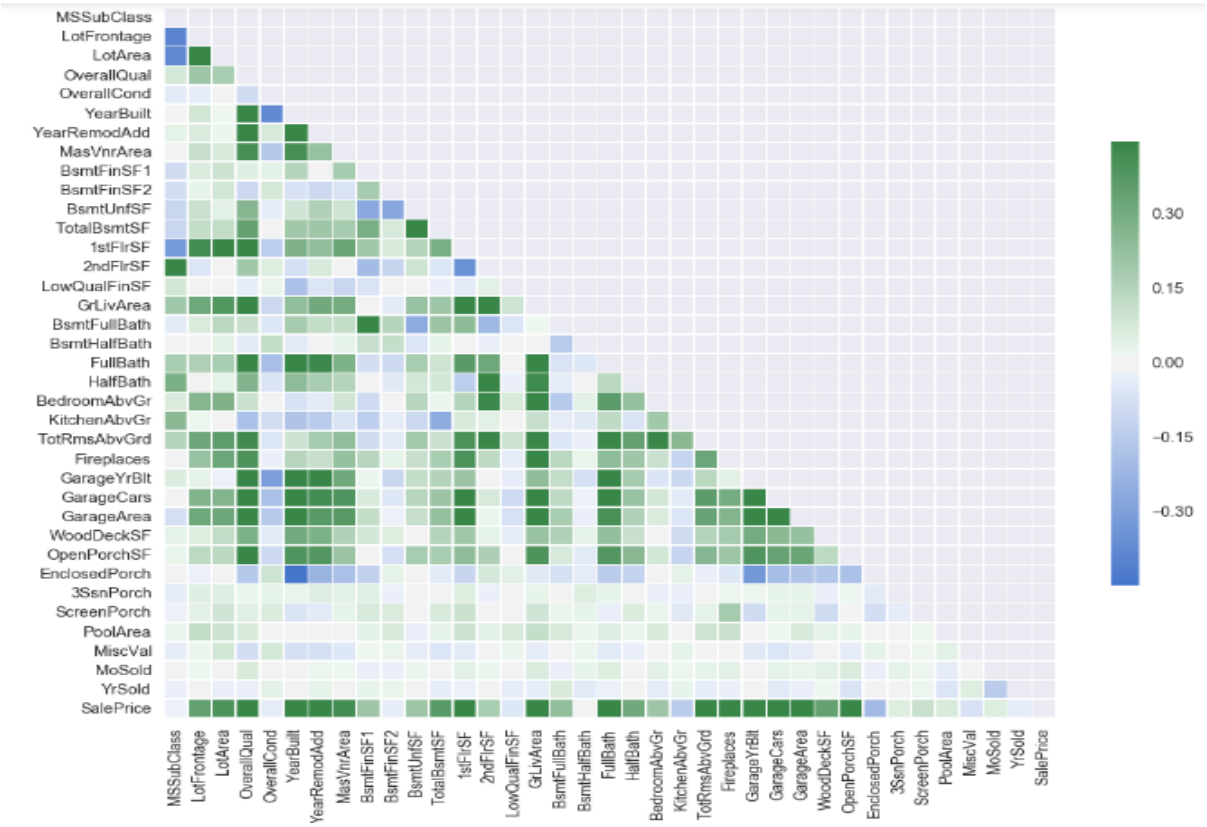
Fig. 1:




The following plot (Fig. 2) shows the distribution of the **target variable, 'SalePrice'** after applying the log transformation (log (x+1)).
The distribution of the target variable is now normally distributed.
Fig. 2:




The plot (Fig. 3) below shows **a heat map of the correlations** between the various numeric variables.
Fig. 3:




As can be seen from above, it can be concluded that the target variable has a strong positive relationship with the following variables –

- LotFrontage
- LotArea
- OverallQual
- YearBuilt
- YearRemodAdd
- MasVnrArea
- TotalBsmtSF
- 1stFlrSF
- GrLivArea
- FullBath
- TotalRmsAbvGrd
- Fireplaces
- GarageYrBlt
- GarageCars
- GarageArea
- WoodDeckSF
- OpenPorchSF

# Methodology

## Algorithms and Techniques

As the problem at hand is the prediction of sale prices (continuous variable) of different residential properties, this is a regression problem.

The algorithms to be used are:

- **Linear Regression –**
  It is an approach for modelling the relationship between a scalar dependent variable (SalePrice) and one or more explanatory variables (other predictors). Linear regression is the most basic and commonly used predictive analysis technique.
  Linear Regression fits a linear model with coefficients w = (w1,…..wp) to minimize the residual sum of squares between the observed responses in the dataset, and the responses predicted by the linear approximation. This method is known as Ordinary Least Square.
  Mathematically it solves a problem of the form:

$$\min_{w} ||Xw - y||_2{}^2$$

- **Random Forest Regression –**
  It an ensemble learning method. In the case of regression, random forests operate by constructing a multitude of decision trees at training time and outputting the class that is mean prediction (regression) of the individual trees.

- **Ridge Regression –**
  Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares,

$$\min_{w} ||Xw - y||_2{}^2 + \alpha||w||_2{}^2$$

  Here, $\alpha \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of $\alpha$, the greater the amount of shrinkage and thus the coefficients become more robust to collinearity.

- **Lasso Regression -**
  Lasso Regression is another technique that addresses the problem of multicollinearity in the predictor variables. It tends to prefer solutions with fewer parameter values, effectively reducing the number of variables upon which the given solution is dependent.
  Mathematically, it consists of a linear model trained with $\ell_1$ prior as regularizer. The objective function to minimize is:

$$\min_w \frac{1}{2n_{samples}}||Xw - y||_2^2 + \alpha||w||_1$$

The lasso estimate thus solves the minimization of the least-squares penalty with $\alpha||w||_1$ added, where $\alpha$ is a constant and $||w||_1$ is the $\ell_1$-norm of the parameter vector.

- **Elastic Net Regression –**
  Elastic Net is a linear regression model trained with L1 and L2 prior as regularizer. Elastic-net is useful when there are multiple features which are correlated with one another. Lasso is likely to pick one of these at random, while elastic-net is likely to pick both.
  The objective function to minimize is in this case

$$\min_w \frac{1}{2n_{samples}}||Xw - y||_2^2 + \alpha\rho||w||_1 + \frac{\alpha(1-\rho)}{2}||w||_2^2$$

- **Principal Component Analysis (PCA) -**
  Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

To test the accuracy of our predictions, the Root Mean Square Error is the chosen metric (details above).

**Root Mean Squared Error (RMSE)** = $\sqrt{((\sum(y_i - \hat{y}_i)^2)/n)}$

where $y_i$: the actual target values
$\hat{y}_i$: the predicted target values
n: number of observations

Very clearly, a model that gives the least value of RMSE is the best model for the data.

# Data Pre-processing

The pre-processing done on the dataset consisted of the following steps:

- Dropping the unnecessary 'Id' column –
  The column 'Id' was just like an index to the data and hence, dropped.
- Transformation of the skewed numeric features –
  The skewness for each of the numeric columns was calculated. The features for which skewness > 0.8 were log-transformed (using the log(x+1) function where x = the feature concerned).

- Missing values –
  The data was then checked for any missing values. The categorical variables (count = 43) had missing values in 16 of the features and the numeric variables (count = 37) had missing values in 3 of the features.
  For each categorical feature, the missing values were imputed with the mode (the most frequently occurring feature) value of each feature.
  For each numeric feature, the missing values were imputed with the mean value of the feature.
- Encoding of the categorical features –
  The non-numeric categorical features were then transformed into numeric features such that they contain values between 0 and (n_classes-1) (using LabelEncoder() from sci-kit learn, Python)

# Implementation

The implementation stage mainly consisted of the following steps –

- **Stage 1**: Building the models
- **Stage 2**: Checking the models' performance metrics

**Stage 1**: During this stage, the different regression techniques were trained on the prepared data.

**Stage 2**: Once the models were built in Stage 1, the models' performance metrics were checked. These include checking the actual vs. predicted plot and calculating the RMSE values.

**Details:**

**Splitting the dataset:**

The processed (pre-processing steps are described in detail above) dataset was split into two - a training dataset (70% of the data points) and a test dataset (30% of the data points). The training set had 1022 observations and the testing set had 438 observations.

**List of Regression techniques used:**

- Linear Regression
- Random Forest Regression
- Ridge Regression
- Lasso Regression
- Elastic Net Regression
- Principal Component Analysis

**Model Parameters:** This is discussed in detail in the section **Refinement.**

# Benchmark

Once the data was pre-processed (explained in detail in the above sections), the data was checked for correlations between the various predictor variables and the correlations between the predictors and the target variable.

The heat map above shows that are many variables that are correlated with the target variable (SalePrice).

To create an initial benchmark, a simple linear regression model was built using ten predictor variables that are most correlated to the target variable.

The features used are: 'OverallQual','GrLivArea', 'GarageCars', 'GarageArea', '1stFlrSF', 'FullBath', 'YearBuilt', 'YearRemodAdd', 'TotRmsAbvGrd' and 'GarageYrBlt'.

**For the benchmark model,**

**RMSE for training set = 0.1638**
**RMSE for testing set = 0.1619**

# Refinement

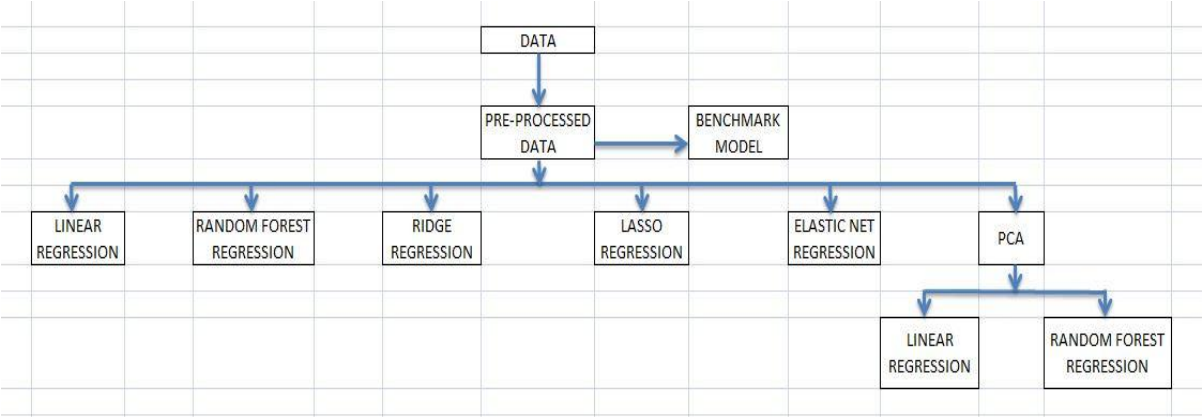
The above shows the results of an initial benchmark model.

This model has been build with the ten most correlated predictor variables with the dependent Sales variable.

Whatever model will be built further will include all other variables present in the dataset so that maximum variability in the dependent variable can be explained by the predictor variables. It needs to be ensures that the new models' evaluation metrics do not perform worse than that of the benchmark model's.

Once we have the results of the benchmark Linear Regression model, Machine Learning algorithms like Random Forest Regression, Ridge Regression, Lasso Regression, Elastic Net Regression, Principal Component Analysis has been used to build a formidable model for this particular dataset.

The methodology followed can be summarized as under -




(Note: The Python code provided is also fairly descriptive. Keeping in mind the above figure, the results shared can be reproduced quite easily.)

**Model Parameters:**

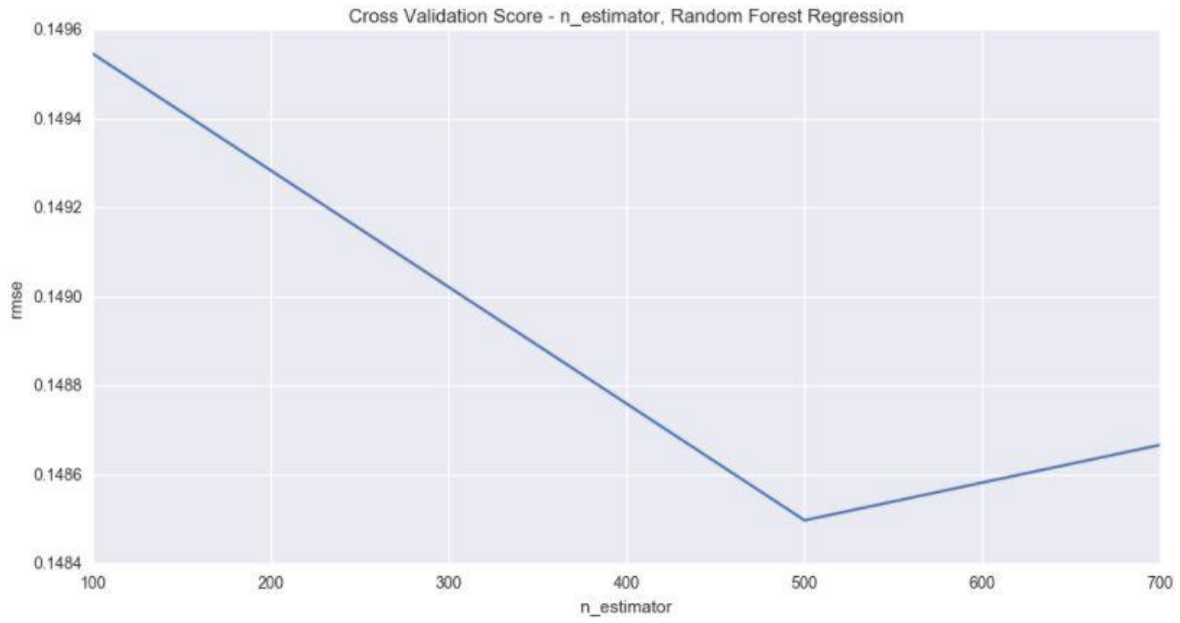
After the Benchmark model on the ten most correlated variables with the dependent Sales price variable, a various regression models were built using all the variables present in the dataset.

- The models were built on the training dataset and the model was tested on the testing dataset.
- The **Actual vs. Predicted plot** and the **Root Mean Squared Error (RMSE)** value was used for the testing purpose.

(The results are shared in the section **Results**)

**Linear Regression:** This model did not require any parameter tuning.

**Random Forest Regression:** For this model, the optimal value of the parameter **n_estimator** (the number of decision trees to build in the model) was first determined from the data. The possible value of this parameter tested were - 100, 500 and 700.



As can be seen in Fig. , the optimal value of n_estimator is 500.

Now that we have the value, the Random Forest Regression model was built following the methodology described above.

**Ridge Regression:**

For this model, a range of values were tested to get the optimal value of Alpha, the regularization parameter in a Ridge Regression model. The range of values tested were: 1e-25, 1e-10, 1e-5, 0.05, 0.1, 0.5, 1, 5, 10, 25, 50 and 100.

Cross-validation on the data, the optimal value of Alpha is 10.

Now that we have the value, the Ridge Regression model was built following the methodology described above.

**Lasso Regression:**

For this model, a range of values were tested to get the optimal value of Alpha, the regularization parameter in a Lasso Regression model. The range of values tested were: 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 and 1.

Cross-validation on the data, the optimal value of Alpha is 0.001.

Now that we have the value, the Lasso Regression model was built following the methodology described above.

**Elastic Net Regression:**

For this model, there were two parameters whose optimal values were to be estimated to get a model that best fits the data. One is the value of Alpha and the other is the L1-ratio.

For Alpha, the values tested were: 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 15, 20, 25, 50 and 100.

For L1-ratio, the values tested were: 0.1,0.5,0.7,0.9, 0.95, 0.99 and 1

After cross-validation, the optimal value of Alpha is 0.001 and the optimal value of L1-ratio is 1.0.

Now that we have the value, the Elastic Net Regression model was built following the methodology described above.

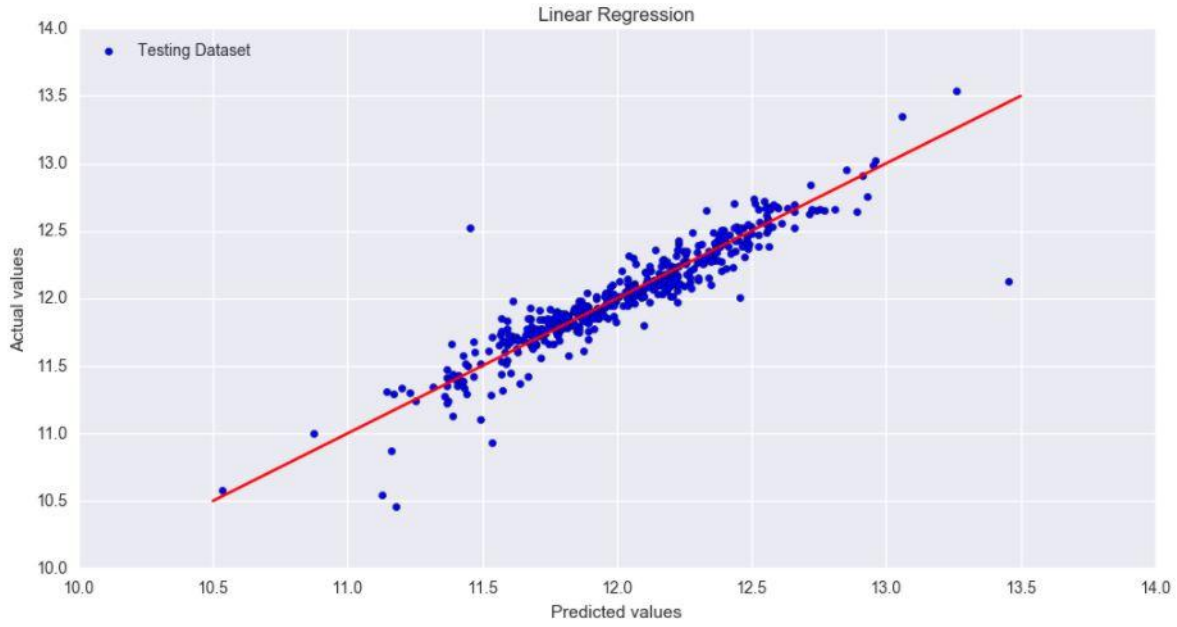
# Results

**Linear Regression -**

RMSE (training set) = 0.1476
RMSE (testing  set) = 0.1589

The plot(Fig. 4) shows the **Actual vs. Predicted plot** for the Linear Regression model.

Fig: 4

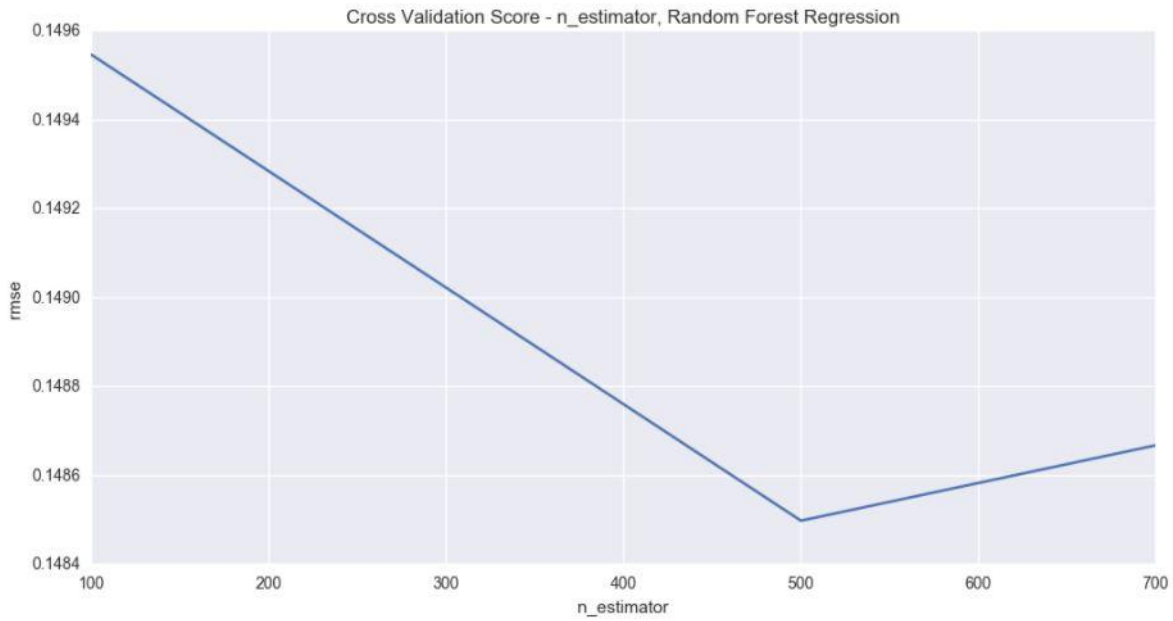



**Random Forest Regression -**

RMSE (training set) = 0.1485
RMSE (testing  set) = 0.1592

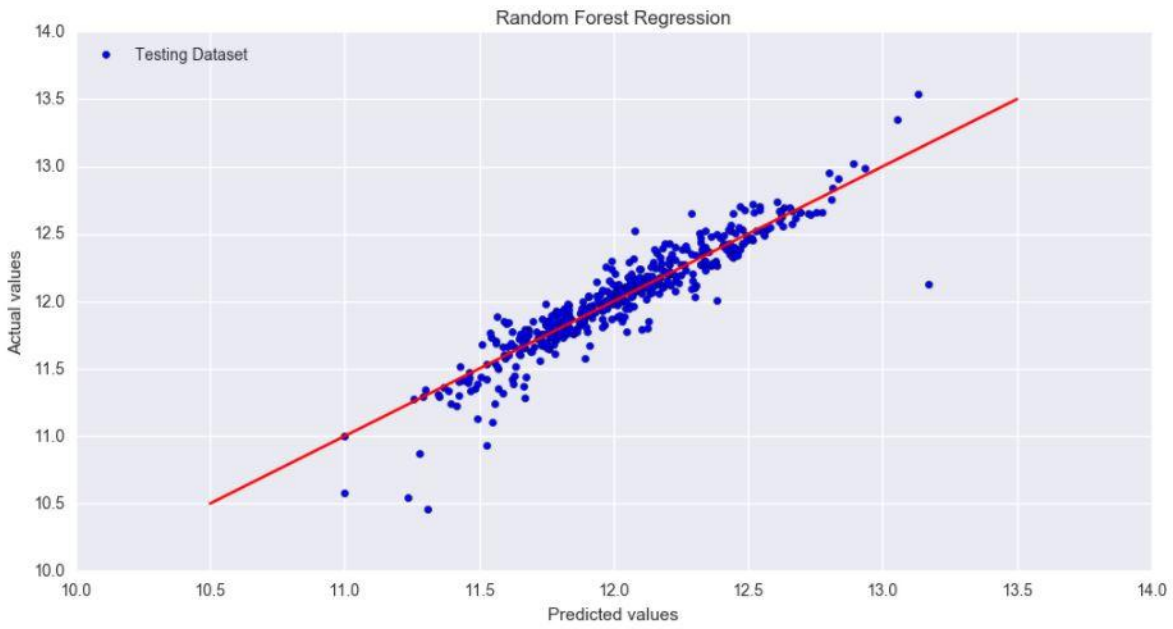
The plot(Fig. 5) shows the **Cross Validation score plot** for getting the optimal value of **'n_estimators'** for the Random Forest Regression model -

Fig. 5:




The plot(Fig. 6) shows the **Actual vs. Predicted plot** for the Random Forest Regression model.

Fig. 6:

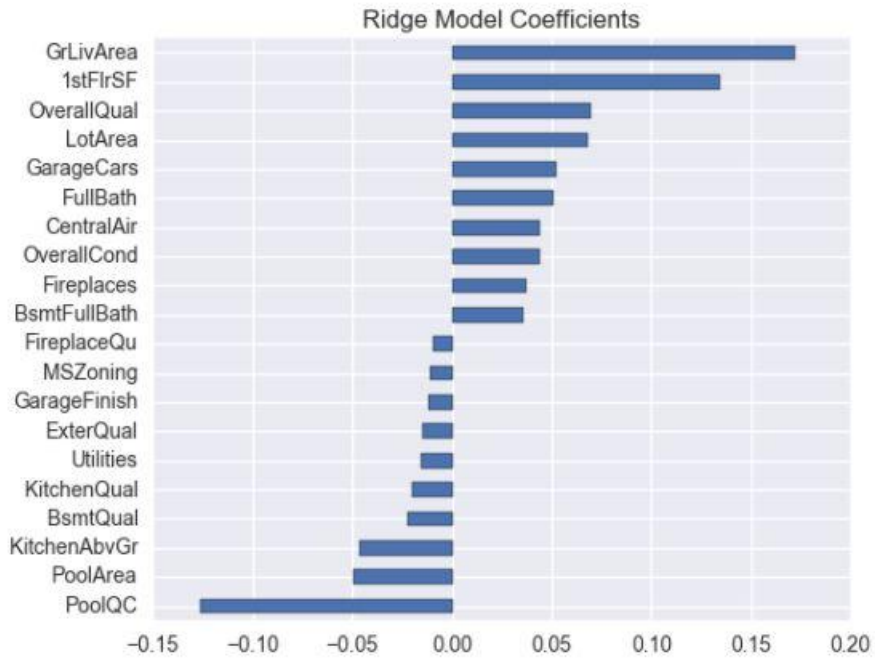



**Ridge Regression -**

Optimum Alpha Value for Ridge Regression = 10
RMSE (training set) = 0.1477
RMSE (testing set) = 0.1559

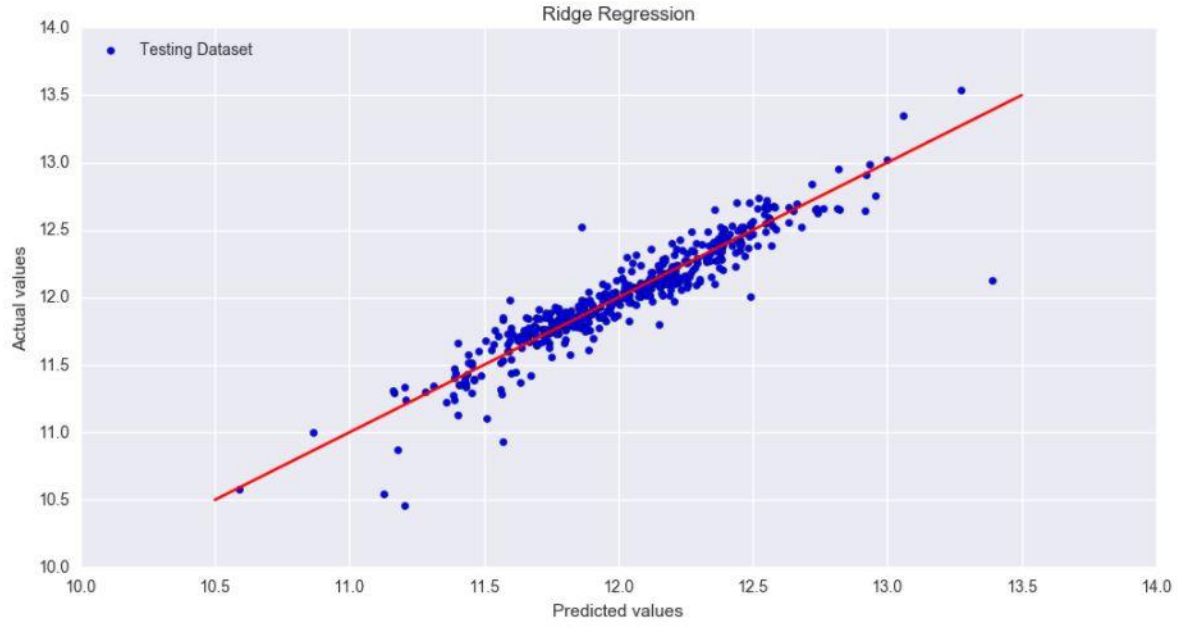
The plot(Fig. 7) shows the values of the **Ridge Model Coefficients**.

Fig. 7:




The plot(Fig. 8) shows the **Actual vs. Predicted plot** for the Ridge Regression model.

Fig. 8:

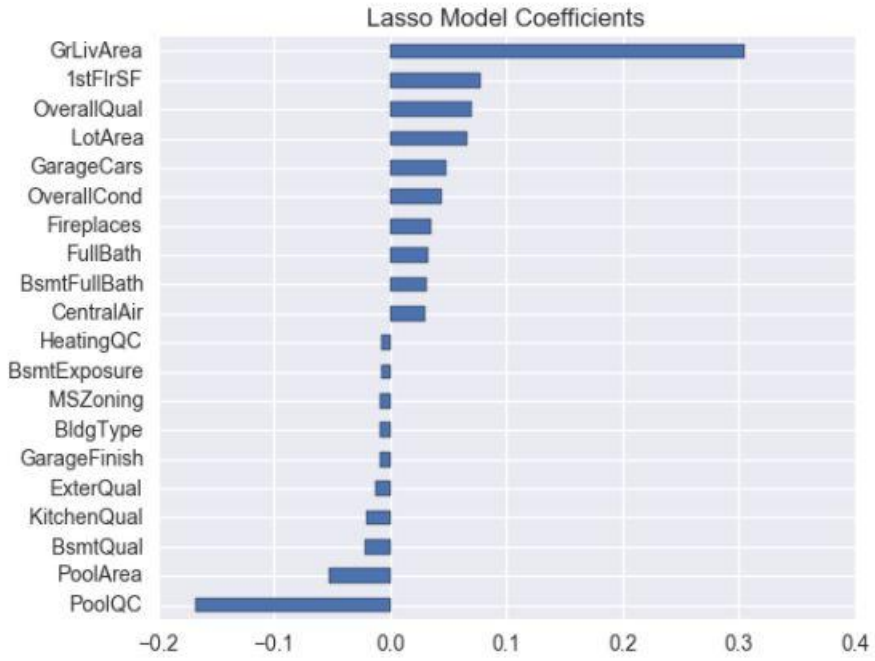



**Lasso Regression -**

Optimum Alpha Value for Lasso Regression = 0.001
RMSE (training set) = 0.1452
RMSE (testing set) = 0.1575

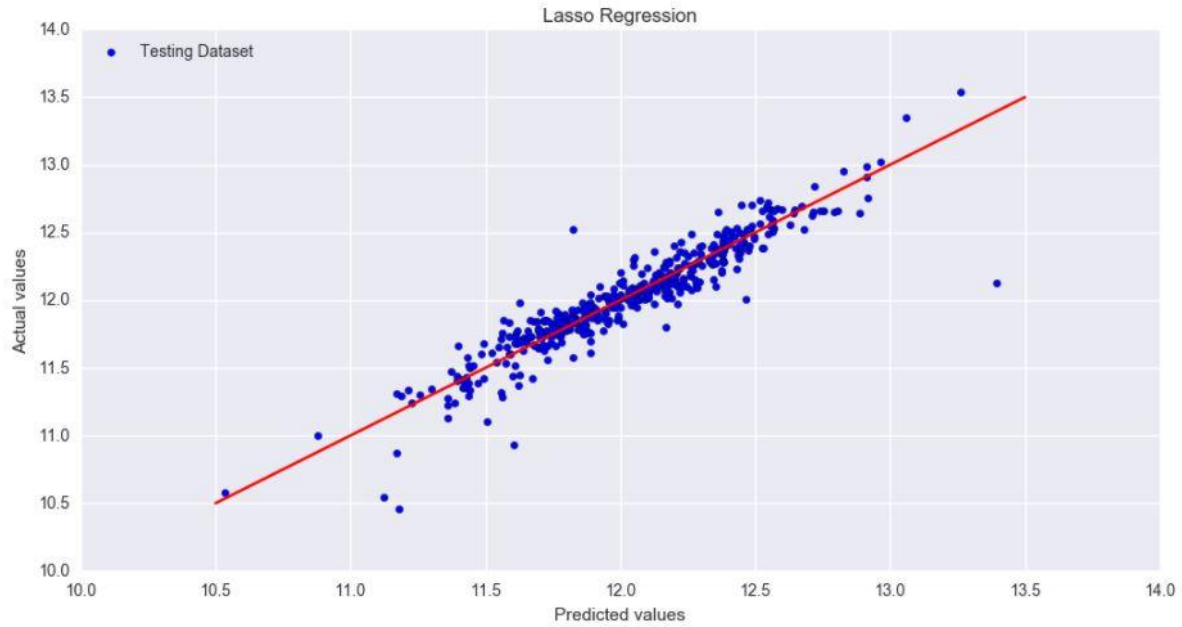
The plot(Fig. 9) shows the values of the **Lasso Model Coefficients**.

Fig. 9:




The plot(Fig. 10) shows the **Actual vs. Predicted plot** for the Lasso Regression model.

Fig. 10:

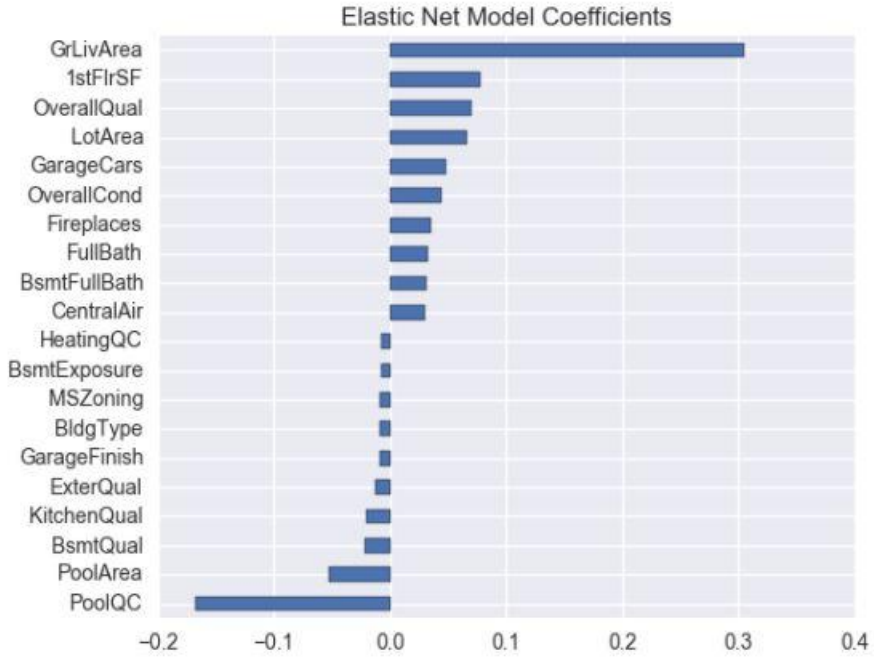



**Elastic Net Regression -**

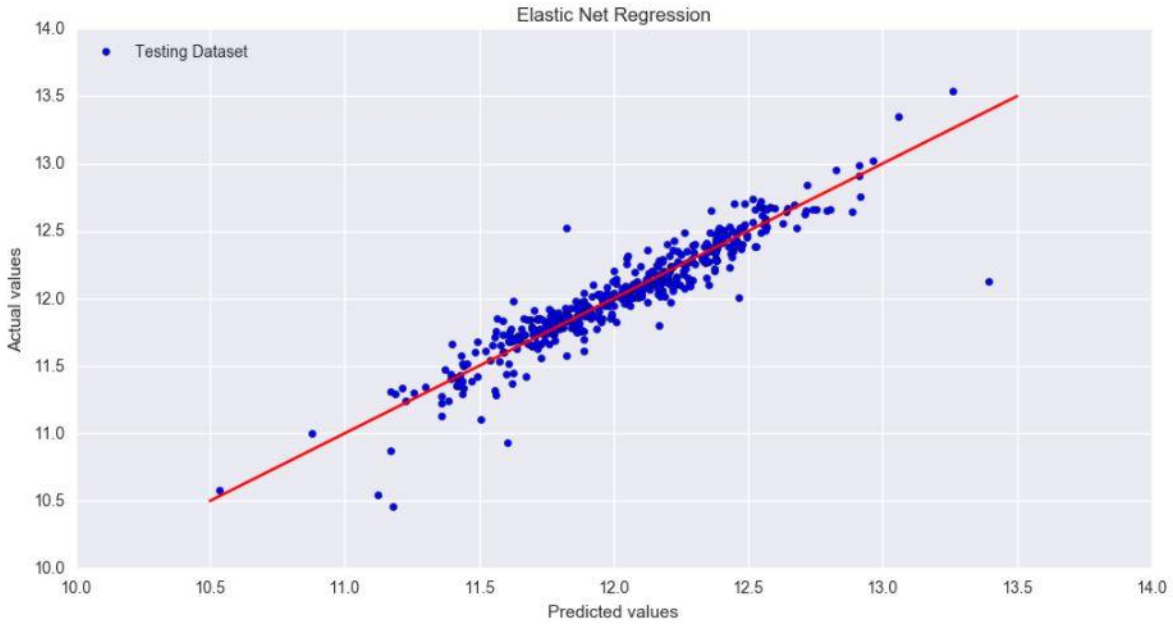
Optimum Alpha Value for ElasticNet Regression = 0.001
Optimum L1 Value for ElasticNet Regression = 1.0
RMSE (training set) = 0.1425
RMSE (testing set) = 0.1508

The plot(Fig. 11) shows the values of the **Elastic Net Model Coefficients**.

Fig.11:




The plot(Fig. 12) shows the **Actual vs. Predicted plot** for the Elastic Net Regression model.
Fig. 12:

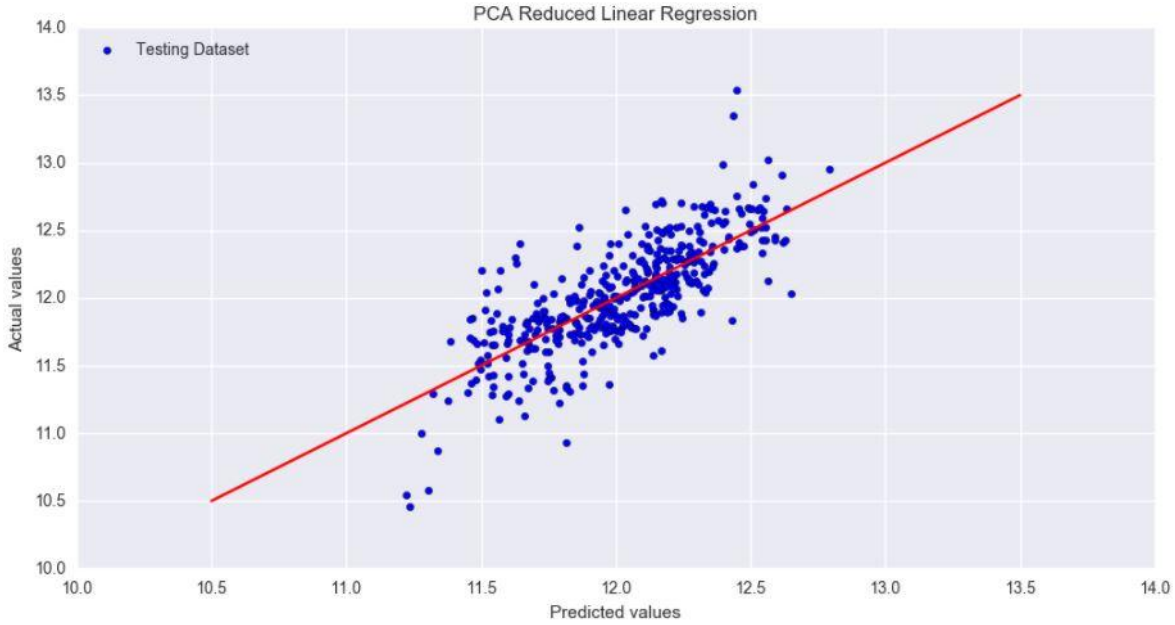



**Linear Regression on the data reduced by PCA -**

RMSE (training set) = 0.2629
RMSE (testing set) = 0.2517

The plot(Fig. 13) shows the **Actual vs. Predicted plot** for the PCA - Linear Regression model .

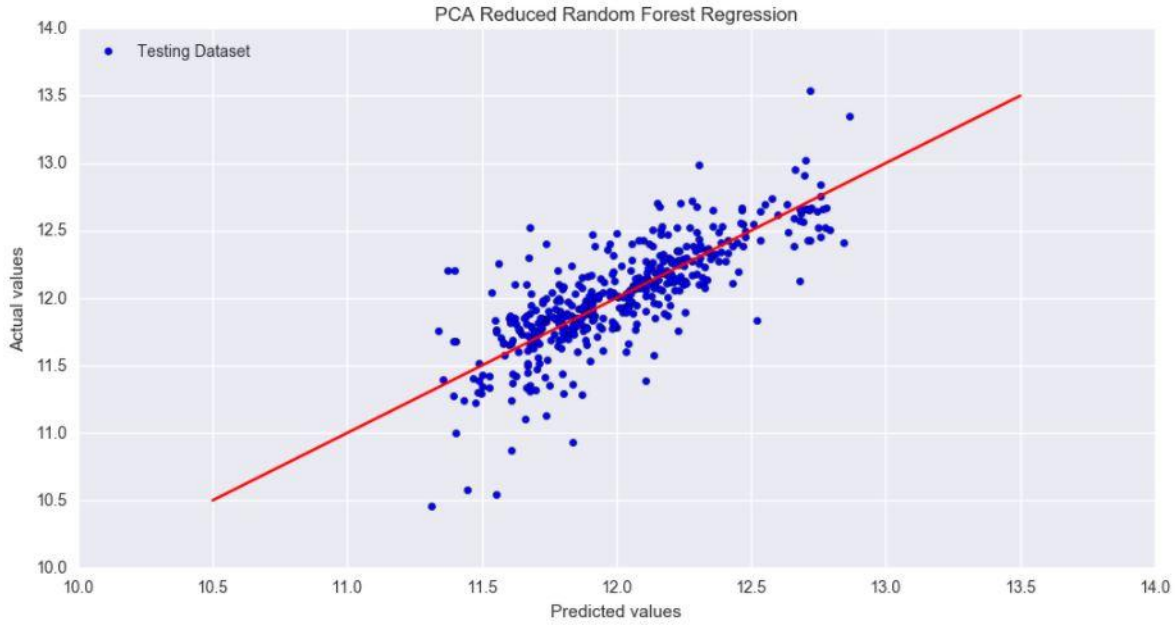
Fig. 13:




**Random Forest on the data reduced by PCA -**

RMSE (training set) = 0.2448
RMSE (testing set) = 0.2418

The plot(Fig. 14) shows the **Actual vs. Predicted plot** for the PCA - Random Forest Regression model.
Fig. 14:



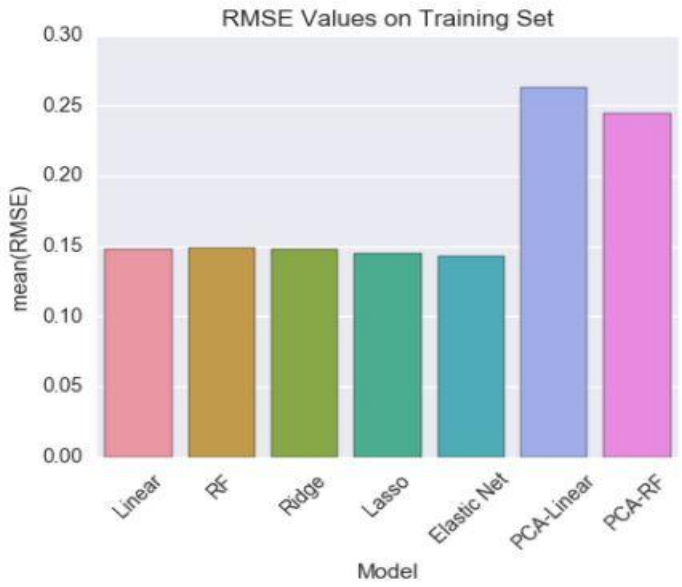
# Conclusion

## Free Form Visualization

Having used different regression techniques on the data, we have the following RMSE values (on the testing dataset) from the various models -

| Model | RMSE |
|---|---|
| Elastic Net Regression | 0.1508 |
| Ridge Regression | 0.1559 |
| Lasso Regression | 0.1575 |
| Linear Regression | 0.1589 |
| Random Forest Regression | 0.1592 |
| PCA - Random Forest Regression | 0.2418 |
| PCA - Linear Regression | 0.2517 |

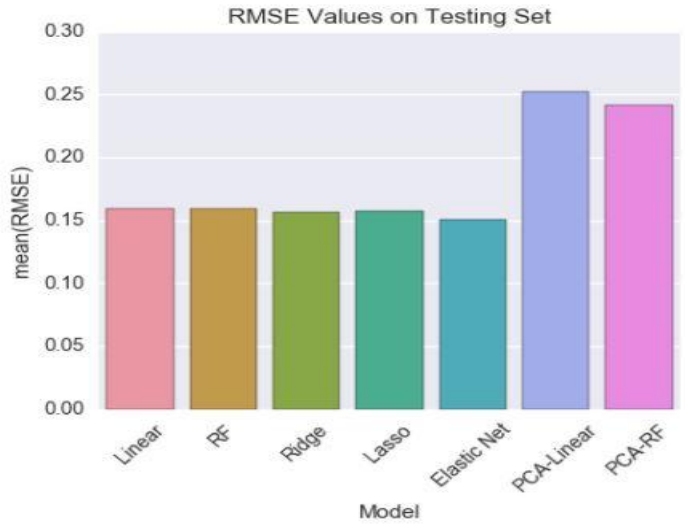
Fig. 15:




The RMSE values (on the training dataset) are as under -

| Model | RMSE |
|---|---|
| Elastic Net Regression | 0.1425 |
| Lasso Regression | 0.1452 |
| Linear Regression | 0.1476 |
| Ridge Regression | 0.1477 |
| Random Forest Regression | 0.1485 |
| PCA - Random Forest Regression | 0.2448 |
| PCA - Linear Regression | 0.2629 |

Fig. 16:




We can see from above that although Lasso Regression fits the training dataset better than Ridge Regression, it fails to perform that well on the testing dataset (falling behind Ridge Regression). This is a sign of over fitting on the training data.

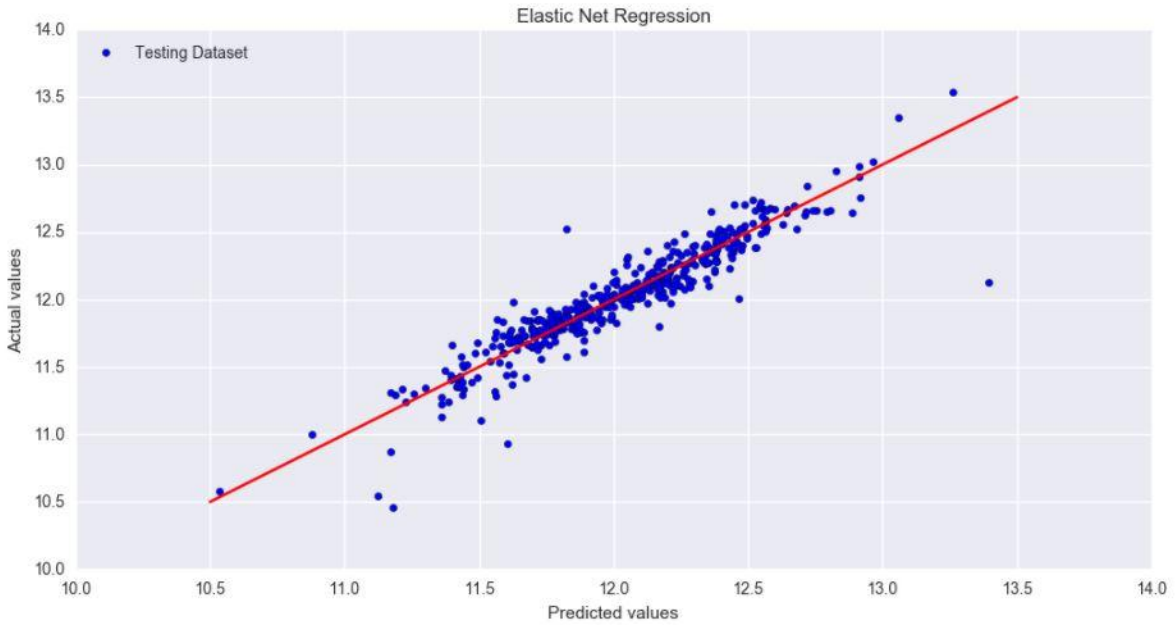
Also, although Linear Regression performs better on the training dataset than on the testing dataset (falling behind Ridge Regression). Again, this may be attributed to over fitting on the training dataset.

However, we can conclude that for the said data, the Elastic Net Regression model is the best model (performing equally well on both the training and the testing dataset).

Here, the optimum Alpha value is 0.001 and the optimum L1 value is 1.0.

We can see that the **Actual vs. Predicted** plot below -

Fig.: 15



Now that we have a model that fits on the training dataset and adequately predicts on the testing dataset, it can be used to predict the sale prices of houses given the values of the independent variables.

## Reflection

Although the problem seemed simple at first glance, it did turn out to be rather gruelling. There were many aspects that needed thorough exploration. The skewness of the dependent variable, the interesting correlations between the variables, imputation of missing values, dealing with categorical variables were some of the areas that required major concentration.

Upon researching various algorithms, it was interesting to note that Principal Component Analysis (PCA) did not perform as good as it was expected to do so. The approach followed was as follows -

1. Run PCA on the processed data.
2. Reduce the large number of independent variables to a lesser number of factors that best explained the variability in the data.
3. Use Regression techniques on this PCA reduced data.

In fact, the RMSE values (Root Mean Squared Error) in this case were far worse than that of the Benchmark Model.

The final model chosen is Elastic Net Regression model. Elastic Net models can handle fairly complicated datasets and perform quite well. Hence, it is not too surprising that this model served as the best solution to the problem at hand.

Due to the nature of Elastic Net models, they can be used in a general setting to solve these types of problems.

## Improvement

Due to the nature of today's growing avalanche of refined techniques and improved methodologies, there is always a scope of improvement to almost all models.

There is the possibility of using Ensemble techniques to refine the models used and to get better results. There is also the vast expanse of Neural Networks and use of Deep Learning techniques that can be utilized in problems like these. These models learn on their own, taking minute details from the dataset into account.

Also, to be fair, a benchmark is always a benchmark. If a better model is chosen as a benchmark, there always will be a scope for improvement!