House Prices: Advanced Regression Techniques

Deepa Kasinathan[1]*, Shridivya Sharma[1],Janani Muppalla[1]

Abstract

Every house has a price at which it will sells.If a house does not sell, there must be some logical reasons to explain why it did not sell. The price must correlate with the location, the features, the condition, the market and recent history. This paper outline the ways to predict the final sale of individual properties in Ames and IOWA from year 2006 to 2010. The dataset contains 2930 observations and 79 exploratory variables giving us insights about every individual property in estimating house properties.We had to do preliminary manipulations to the dataset prior to the further analysis. Inconsistencies and missing values were taken care during this process. We also analyzed selection of features which could help us in predicting the sale price of a house property more accurately. With appropriate features and cleaned data, We applied different regression models to predict the final sale price

Keywords

Multiple Regression– Linear Models–Assessed Value– Group Project

[1] *Computer Science, School of Informatics and Computing, Indiana University, Bloomington, IN, USA*

***Corresponding author**: deepkasi@indiana.edu,shrishar@iu.edu,jmuppall@iu.edu

All work is solely ours

Contents

Introduction

Introduction: Data analytics has taken over the limelight since the data explosion and there is no field where it hasn't been implemented! House sales prediction is one of them. Demand is healthy and house values are rising. Owning a home remains a better deal than renting one and mortgage rate are near record lows, meaning borrowing money to buy is cheap. Economists say, that after years of depressed prices, many house owners have regained much of the equity they lost in the downturn, so they may seek to cash in on that value and sell in 2016 to move up to their next home. As the economy continues to grow and more jobs are added, potential home buyers with strong credit will be more willing to jump into the market.

Predictive analytics of house sales can be helpful to both buyers and sellers. Who doesn't want systems that are more predictive? Everybody wants everything scored. Everybody wants to understand what the next best offer is, the next best opportunity and how to make things a little bit more efficient. Similarly, housing and management agencies can use predictive analytics to assess various departments of their business. On the other hand, buyers will use it to find the qualities they truly need in a home and the amount of investment they would need to buy that particular house in future. The house sale prediction project has been a significant part of the CSCI- B565 Data mining course curriculum. This project analyses house sales data based on 79 critical parameters of which buyers generally base their decisions on buying a house. This project required us to work on teams and analyze the raw data that has been provided to us and come to making meaningful insights. Initially, raw data is cleaned and missing data is predicted using pre-processing methods. We then worked on feature selections as to find out which feature gives more accurate values. Using these features we implemented different kinds of predictive models like – Linear regressions, Random Forest and Lasso regression model to reach to our conclusion.

1. Background

1.1 Linear regression

Linear regression is an approach for modeling the relationship between a scalar dependent variable y and one or more independent variables) denoted X. The case of one independent variable) is called simple linear regression.
$y_i = \alpha + \beta x$

Linear regression has some assumptions and they are as follows:

1. Linear relationship: linear regression needs the relationship between the independent and dependent variables to be linear. It is also important to check for outliers since linear regression is sensitive to outliers effects.
2. Multivariate normality: The linear regression assumes that all the the variables are multivariate normal. If the data is not normal, then normalizing the data will fix this issue.
3. No or little multicollinearity: Linear regression assumes that there is no multicollinearity. Multicollinearity occurs when the independent data are not actually independent but they have some dependence. This can be checked using Principle component analysis.
4. No auto-correlation. Autocorrelation occurs when the residuals are not independent from each other. In other words when the value of y(x+1) is not independent from the value of y(x)
5. Homoscedasticity: This means that different response variables have the same variance in their errors, regardless of the values of the predictor variables

1.2 Random Forest

Random forests or **random decision forests** are an ensemble learning method for classification, regression.
[1] In Random Forest classification is done by growing multiple classification trees. To classify a new object from the given input, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree ”votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest). Features of Random Forests

1. It gives excellent accuracy among current algorithms.
2. It runs efficiently on large data bases.
3. It can handle thousands of input variables without variable deletion.
4. It gives estimates of what variables are important in the classification.

One more advantage of random forest is it does not overfit and it is fast. You can run as many trees as you want

1.3 Feature Selection

1.3.1 Feature

A feature is a piece of information that might be useful for prediction. Any attribute could be a feature, as long as it is useful to the model.The purpose of a feature, other than being an attribute, would be much easier to understand in the context of a problem. A feature is a characteristic that might help when solving the problem.

1.3.2 Importance of features selection

Feature selection is an important part of machine learning. Feature selection refers to the process of reducing the inputs for processing and analysis, or of finding the most meaningful inputs. Feature selection is important for building a good model for several reasons. One of the advantage of feature selection is cardinality reduction, we can cut off on the input attributes that can be passed to our predictive model. Sometimes data can contain information which might not be useful for our model or wrong kind of information. Thus discarding unnecessary attributes from the data set can improve overall quality of predictive model. [2]

Not only feature selection helps in improving the quality of the model , but it also makes the process of modeling more efficient. For example, if a dataset contains 500 columns, more CPU and memory is required during the training process and more space in memory is required for the completed model.
In nutshell: Feature selection helps us in the following ways:

1. **Removing Noisy/ redundant data:** Presence of noisy data makes it more difficult to discover patterns
2. **Dimensionality Reduction :**
   If the data set is high-dimensional, most data mining algorithms require a much larger training data set.

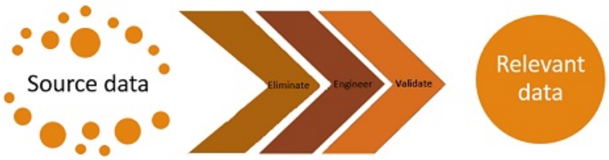


Figure 1. Data Features

In short, feature selection helps solve two problems: having too much data that is of little value, or having too little data that is of high value. Our goal in feature selection should be to identify the minimum number of columns from the data source that are significant in building a model.

1.3.3 Feature Selection Techniques

[3]

1. **Filter methods:**
   These are preprocessing methods. They attempt to assess the merits of features from the data, ignoring the effects of the selected feature subset on the performance of the learning algorithm. Examples are methods that select variables by ranking them through compression techniques (like PCA or clustering) or by computing correlation with the output.

2. **Wrapper methods:**
   These methods assess subsets of variables according to their usefulness to a given predictor. The method conducts a search for a good subset using the learning algorithm itself as part of the evaluation function. The problem boils down to a problem of stochastic state space search. E.g. the stepwise methods in linear regression.

3. **Embedded methods:**
   They perform variable selection as part of the learning procedure and are usually specific to given learning machines. Examples are classification trees, random forests, and methods based on regularization techniques.

1.3.4 Filter methods

Principal Component Analysis

Principal component analysis (PCA) is one of the most popular methods for linear dimensionality reduction. It can project the data from the original space into a lower dimensional space in an unsupervised manner. Each of the original dimensions is an axis. However, other axes can be created as linear combinations of the original ones. PCA creates a completely new set of axes (principal components) that like the original ones are orthogonal to each other.

1. The first principal component is the axis through the data along which there is the greatest variation amongst the observations. This corresponds to find the vector a=$[a_1, a_2, ..., a_n] \in R^n$ such that the variable

$$z = a_1.x_1 + a_2.x_2 + ... + a_n.x_n = a^T x \quad (1)$$

   has the largest variance. It can be shown that the optimal a is the eigenvector of Var $[x]$ corresponding to the largest eigenvalue.

2. The second principal component is the axis orthogonal to the first that has the greatest variation in the data associated with it; the third p.c. is the axis with the greatest variation along it that is orthogonal to both the first and the second axed; and so forth

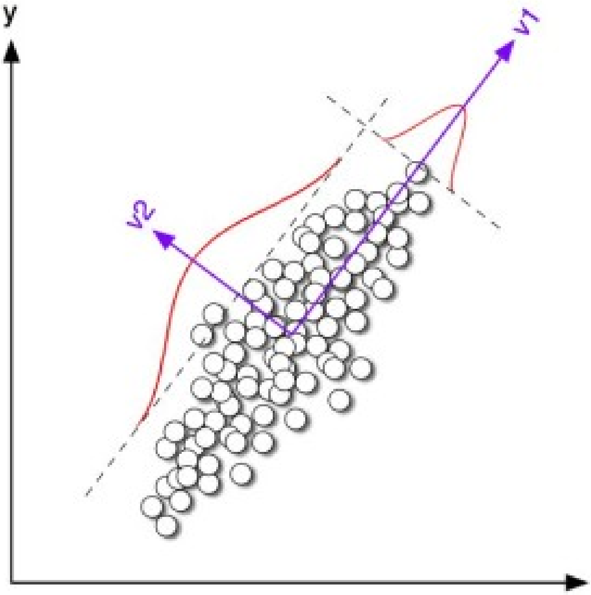


Figure 2. PCA

1.3.5 Clustering

This is also known as unsupervised learning.All these methods require the definition of a distance function between variables and the definition of a distance between clusters. **Nearest neighbor clustering**: The number of clusters is decided first, then each variable is assigned to each cluster. Examples are Self Organizing Maps (SOM) and K-means.

Agglomerative clustering: They are bottom-up methods where clusters start as empty and variables are successively added.Example is hierarchical clustering: it begins by considering all the observations as separate clusters and starts by putting together the two samples that are nearest to each other. In subsequent stages also clusters can be merged. The output is a dendogram.

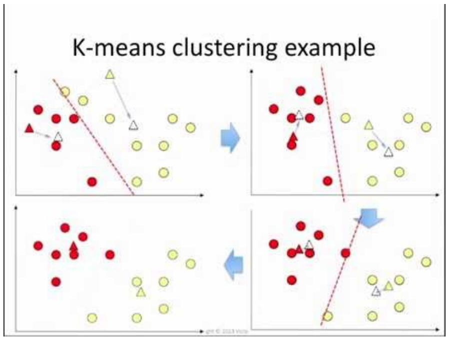


Figure 3. Clustering

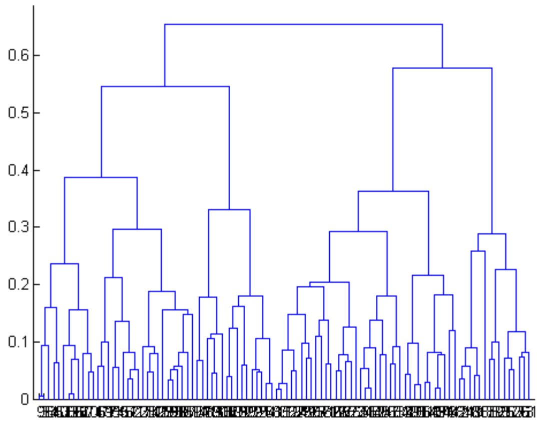


Figure 4. Dendograms

1.3.6 Ranking

They assess the importance (or relevance) of each variable with respect to the output by using a univariate measure.They are supervised techniques of complexity O(n).
Measures of relevance which are commonly used are:

1. Pearson correlation (the greater the more relevant) which assumes linear dependency

2. Significance p-value of an hypothesis test (the lower the more relevant) which aims at detect the genes that split well the dataset. Parametric (t-test) and nonparametric (Wilcoxon) tests have been proposed in literature

After the univariate assessment the method ranks the variable in a decreasing order of relevance.These methods are fast (complexity O(n)) and their output is intuitive and easy to understand. At the same time they disregard redundancies and higher order interactions between variables.

2. Data Analysis

2.1 Cleaning the data

Before applying further predictive model on the raw data,data obtained must be processed or organized for analysis. We have analyzed the data initially by going through it in excel. Our Data analysis observations were as follows.

1. **Train Data**:3 Columns have missing values. In these 1 columns are numeric columns and 2 Columns are categorical with missing values.

2. **Test Data**: 2 Columns have missing values. In these 1columns are numeric columns and 1 Columns are categorical with missing values.

For further analyzing data, we loaded the test and train data into two data frame using pandas in python. There are several types of data cleaning that depend on the type of data. Since we are dealing with numerical and categorical data, we used the following technique for commuting the missing values:

1. **Numerical values**: We used mean values for filling up the missing data for numerical values.We can also use median for doing the same.

2. **Categorical Values**: We replaced missing values by the most frequent occurring value in data frame (train and test).We also tried filling it with dummy variables, but we could not find a much difference in the accuracy in our predicted value.

2.2 Data Normalization:

We have to normalize our data, so that proper features are selected for feature selection. In data normalization we normalize or scale the data so that all data is one scale. After applying normalization, all the attributes will transform to a common range. If the data of some attributes is large and sparse, then the larger scale attribute will influence the outcome. In order to eliminate or minimize such bias, we must normalize the data. There are multiple approaches for data normalization. We applied log transformation on data to do normalization.[4]

2.3 Hot Encoding

One hot encoding transforms categorical features to a format that works better with classification and regression algorithms .Let's take the following example. We have a category - 'taste' and it has three variable - good,better and best. We could have assigned some random numerical value but it would not make any sense of machine learning algorithms.

Sample	Category-'Taste'
1	Good
2	Better
3	Best
4	Good

Instead of assigning a numerical value, we can generate one boolean column for each category.

Sample	Good	Better	Best
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0

This works very well with most machine learning algorithms. Some algorithms, like random forests, handle categorical values natively. Then, one hot encoding is not necessary. The process of one hot encoding may seem tedious, but fortunately, most modern machine learning libraries can take care of it.

2.4 Feature Selections using PCA

Correlation is a statistical measure that describes relationship between two or more independent variables. A positive correlation tells how much one variable can increase or decrease in parallel; a negative correlation indicates the extent to which one variable increases as the other decreases.
It is important to understand our underlying dataset structure and characteristics so that we can decide which attributes are highly correlated with each other. For Machine learning algorithm, correlation can help us in selecting proper attributes which can help us predict our target variable better.

Pearson Correlation Coefficient:
[5]
One of the simplest method for understanding a feature's relation to the response variable is Pearson correlation coefficient, which measures linear correlation between two variables.
The resulting value lies in [-1;1], with -1 meaning perfect negative correlation (as one variable increases, the other decreases), +1 meaning perfect positive correlation and 0 meaning no linear correlation between the two variables.
We can find correlation between sales price and each input attribute. Predictivity is nothing else but check how much can one input attribute is able to predict a target variable.
Using Pearson function in python , we calculated the correlation for each attribute and target variable:

Feature	Correlation
OverallQual	7909816005838047
GrLivArea	0.7086244776126511
GarageCars	0.640409197258349
GarageArea	0.6234314389183598
TotalBsmtSF	0.6135805515591944
1stFlrSF	0.6058521846919166
FullBath	0.5606637627484452
TotRmsAbvGrd	0.5337231555820238
YearBuilt	0.5228973328794967
YearRemodAdd	0.5071009671113867
GarageYrBlt	0.48636167748786213
MasVnrArea	0.4774930470957107
Fireplaces	0.4669288367515242
BsmtFinSF1	0.38641980624215627
LotFrontage	0.35179909657067854
WoodDeckSF	0.32441344456813076
2ndFlrSF	0.31933380283206614
OpenPorchSF	0.31585622711605577
HalfBath	0.2841076755947784
LotArea	0.2638433538714063
BsmtFullBath	0.22712223313149718
BsmtUnfSF	0.214479105546969
BedroomAbvGr	0.1682131543007415
KitchenAbvGr	-0.1359073708421417
EnclosedPorch	-0.12857795792595636
ScreenPorch	0.11144657114291048
PoolArea	0.09240354949187278
MSSubClass	-0.08428413512659523
OverallCond	-0.0778558940486776
MoSold	0.04643224522381936
3SsnPorch	0.04458366533574792
YrSold	-0.028922585168730426
LowQualFinSF	-0.02560613000068015
Id	-0.021916719443431112
MiscVal	-0.02118957964030379
BsmtHalfBath	-0.016844154297359294
BsmtFinSF2	-0.011378121450215216

The housing price correlates strongly with **OverallQual, GrLivArea(GarageCars), GargeArea, TotalBsmtSF, 1stFlrSF, FullBath, TotRmsAbvGrd, YearBuilt, YearRemodAdd, GargeYrBlt, MasVnrArea and Fireplaces**. But some of those features are highly correlated among each others.

2.4.1 Visualizing Correlations

Let's visualize correlation between the given attributes

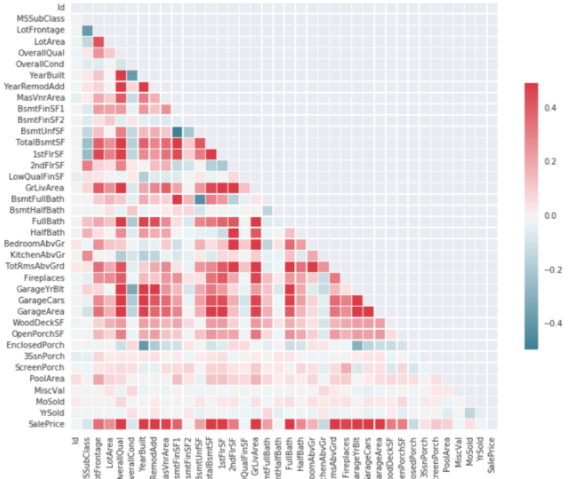


Figure 5. Correlation between attributes

Let us visualize sale price variation with highly correlated values :

Sales price Vs OverAll quality : Since overall quality attribute is highly correlated with sales price,using regplot (seaborn) package- we can draw a scatterplot of two variables, sale price and Overall quality and then fit the regression model sales price $\sim$ overall and plot the resulting regression line and a 95% confidence interval for that regression

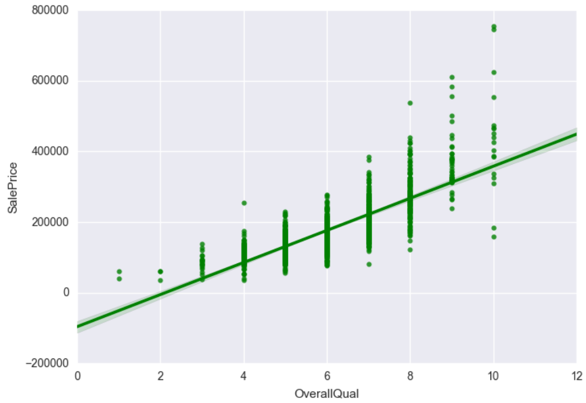


Figure 6. correlation between saleprice and over quality

For other variables - lets us see scatter plots: (Scatter plots are used to see the correlations between two variables)

3. Problem Description

The main aim of this project is to predict the final sales price of the house when 79 variables are given which explains different features of the house. The main problem is that this data has many dimensions on which the final price is dependent. So, in our initial meetings we have analyzed the data and figured out the problems involved in this scenario.

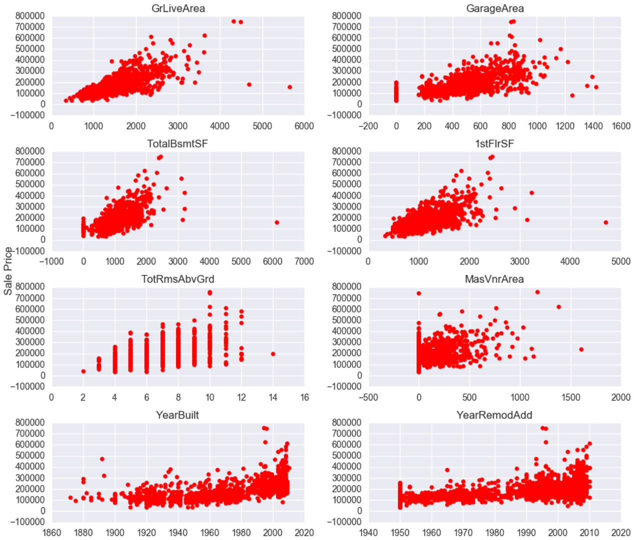


Figure 7. correlation between saleprice and other attributes

The problems are as follows:

1. First problem is dealing with high dimensional data. Each row has 79 attributes on which the sales price depends. So, we had to analyze whether any attributes are providing redundant information so that dimensionality can be reduced. Should dimensionality reduction techniques like PCA are to be implemented?. Will dimensionality reduction actually make a difference in predicting the output or all attributes should be used in predicting the sales price?.

2. Second problem is with missing values. The data provided has a lot of missing values and we know that multiple approaches like mean, median, mode can be used to replace these missing values.

3. Third problem is normalizing the data.

4. Fourth problem is dealing with categorical data. In the given data, many attributes have categorical values. The categorical data cannot be directly used in any regression algorithms and therefore these categorical data have to be converted into numerical values.

5. Fifth problem is feature selection. After data preprocessing, feature (attribute) selection is to be done because there are 79 attributes and somehow the features that are important for the prediction of sales price are to be figured out.

6. Sixth problem is selecting the regression model. There are many models like linear regression, ridge regression, random forest which can be used in prediction of the sales price.

4. Algorithm and Methodology : Experimenting With Data

4.1 Experiment 1 : Using Linear Regression Model with RFE feature selection

Using pandas and numpy packages, we loaded our train and test data into python. After carefully observing train and test data, we found that there were missing values in both the datasets. We combined both the dataset so that we can compute missing values as well as to maintain the consistency throughout the experiment. We wrote own two functions one which will separate the target value and rest of the attributes (Target : sales price) and other function to calculate the missing values for rest of the attributes. We will convert cleaned data's categorical values into numerical values using hot encoding technique . We used RFE package from sklearn feature selection which to select best features on the basis of their relevance in minimizing sum of squared errors. RFE.ranking command will give us the rank for each attribute and we selected the top 20 among those. We created one linear regression model, fitted our train data and finally predicted value using test data.

4.2 Experiment 2 : Using Random Forest Model with RFCEV feature selection

We followed the same procedure with data preprocessing as in our experiment 1. We used random forest model instead of linear regression. In this experiment we used n-estimators as 20 which means our random forest model have 20 different decision trees. The final result will be the mean of predicted values computed by these 20 different decision trees. As a part feature selection, we used RFCEV package of sklearn feature selection which automatically gives n best features where n is decided by model itself based on data.

4.3 Experiment 3: Using Random Forest Model and PCA feature selection

With not much accuracy using RFE and RFCEV methods for feature selection ,We decided to use PCA for further analysis. We found out that OverallQual, GrLivArea(GarageCars), GargeArea, TotalBsmtSF, 1stFlrSF, FullBath, TotRmsAbvGrd, YearBuilt, YearRemodAdd, GargeYrBlt, MasVnrArea and Fireplaces were highly correlated with Sales Price. In addition to this, We figured out few categorical values for example- HeatingQC which had five 'Ex','Gd',TA, 'Fa','Po' attributes.Now some of these attributes were highly correlated among themselves. So we replaced HeatingQC value something like :
$alldata$.HeatingQC.replace
$(\{'Ex' : 0,$
'Gd': 0,
'TA': 0,
'Fa': 1,
$'Po' : 1\})$

This helped us in making our model robust because earlier while hot encoding all the five were converted into five different binary columns , but now Ex,Gd and TA will have one binary column and Fa and Po will have one binary for these two. Due to this modification for categorical attributes, we could reduce the number of columns exponentially. We applied random forest model over this optimized data and results were much more accurate than the previous two experiments.

5. Experiments and Results

Experiment 1 : Linear Regression Model Running our linear regression model with RFE feature selection gave us total accuracy of 68% and a total error of 0.18743.
Our actual sale price: Our predicted values:

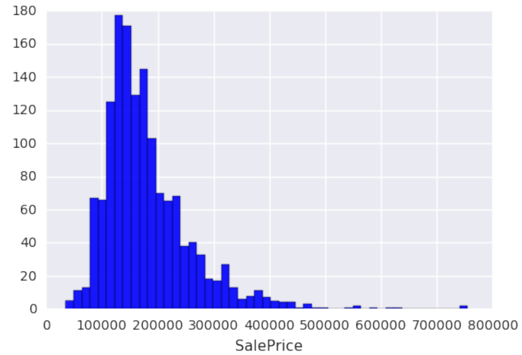
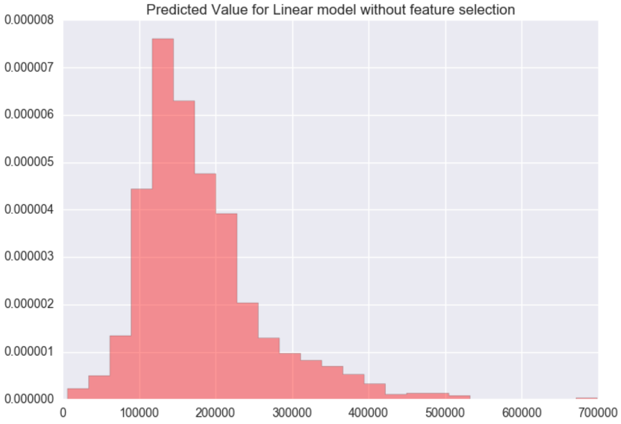


Figure 8. "Our Actual Values"

Figure 9. Predicted Values without using Feature Selection



Experiment 2: Random forest Model with RFCEV selection : Running our Random Forest Model with RFCEV feature selection gave us total accuracy of 86% and a total error of 0.1445.
Our predicted values:

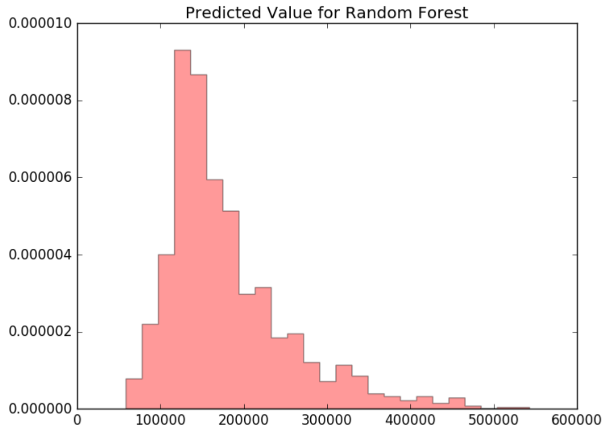


Figure 10. Random forest Model with RFCEV selection

Experiment 3: Random Forest with PCA : Running our Random Forest Model with PCA feature selection gave us total accuracy of 89% and a total error of 0.1224.
Our predicted values:

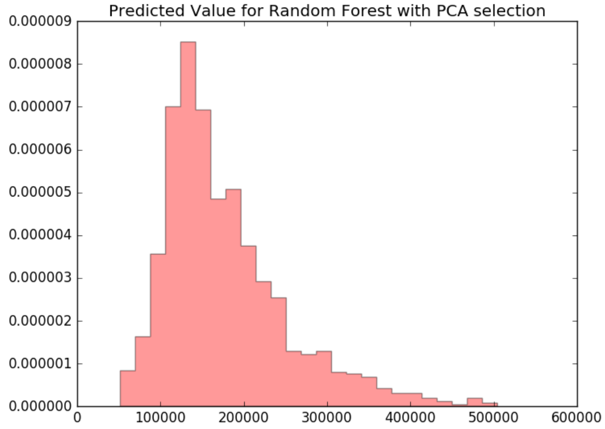


Figure 11. Random Forest with PCA

6. Summary and conclusion:

Our analysis and results say that preprocessing and feature selection is very much important than the model or the technique that we used for predicting the house sales price. We have implemented Linear regression, Random forest models to fit the data. Cleaned the data by replacing missing values with mean and median, applied data normalization and transformation and feature selection using PCA and RFECV. The result using random forest as the model and PCA as the feature selection gave us 88 % accuracy for predicting the sales price of the house. We have submitted this on kaggle and currently we are in top 25%.(Team Name: PowerPuff Girls - Rank - 457)

7. Future Work

We all know that finding a perfect model that will give us best results is not very easy and for that we have to implement advanced regression models and preprocess the data in other ways.

1. Lasso regression can be implemented and analyze how accurate it is on predicting the value.
2. Analyze the data more and use a different approach to scale the data which will not result in many columns. Because the approach which we are using now results in many columns which increases the dimensions.
3. Implement a better feature selection approach. Manual feature selection or different approaches and check which approach will result in a better prediction of sales price.
4. Xg boost technique can also be implemented. When i have studied this i feel that xg boost with lasso regression combination might yield proper results and better accuracy when compared to the approach we used.
5. Knn can be used for replacing the missing values, which will be a better replacement of missing values instead of giving mean and mode.
6. Decision trees can also be used for predicting the correct values of missing data. This can be compared with various other techniques and the one with the better accuracy can be chosen.
7. Data normalization or data standardization of the variables can be improved by centering the data, substracting the mean and normalize it dividing by the variance (or standard deviation too).

8. Appendix

We have written program in python and to implement various regression models we have imported packages from scikit.
Language: Python
All the programs are written in python.
Database SQL : Stored the data in MySQL to understand the data and perform some initial analysis like how many missing values are represent?. All Queries are written in SQL Language.
Packages used: Imported package to implement linear regression and random forest. The packages are as follows. [6]

1. Imported pandas.
2. Imported numpy.
3. Imported RFECV package from Sk Learn to implement RFECV feature selection.
4. Imported random forest regressor from sk learn.ensemble to implement Random forest.

Data used: We have used the data provided in the kaggle site and the data file names are as follows.

1. Datadescription.csv- To understand the description of the data in detail like the importance of attributes in the data.
2. Train.csv- This data file is used in training our model.
3. Test.csv- Using this data file we tested we tested our model.

Acknowledgments

This project would not have been possible without the kind support and help of Associate instructors. We would like to express our special thanks to Hasan Kurban and Kurt who helped us in understanding this properly and were always there when we needed guidance in doing this project. We are highly indebted to the professor Mehmet Dalkilic, Associate professor of Informatics and computing for his guidance and constant supervision as well as for providing the necessary information regarding the project.

References

[1] Random Forest. https://en.wikipedia.org/wiki/Randomforest.

[2] https://www.cs.cmu.edu/~kdeng/thesis/feature.pdf. CS CMU Feature.

[3] Feature Selection. http://machinelearningmastery.com/feature-selection-in-python-with-scikit-learn. Machine Learning Mastery.

[4] Kumar Tan, Steinbach. *Introduction to Data Mining*. 2012.

[5] http://www.statstutor.ac.uk/resources/uploaded/pearsons.pdf.

[6] Linear Model. http://scikit-learn.org/stable/modules/linear_model.html. Scikit Model.