House Price Predictions in Iowa

Team Members
Akshith Meghawat ammeghaw
Harika Malapaka hsmalapa@ncsu.edu
Sanya Kathuria skathur2@ncsu.edu

Date : September - December (2018)

Abstract

The house prices of a certain city can be predicted using many features like presence of basement and proximity to railroad, apart from common features like square feet of the house. People in general wouldn't think of such features. The goal of this project was to conduct EDA, data cleaning, preprocessing, and develop models to predict hoouse prices given the features of the house, and interpret the models to find out what features add value to a house. To make use of all these variables and predict the house prices, neural net (an advanced machine learning technique) would be a good approach. Regression using Neural nets proved to be a good model when applied on housing data of Iowa state.

1 Introduction

From the class project point of view, after attending our peers' presentations, we wanted to deep dive on neural networks as part of this project. After learning the basics from the class, applying them on such project would develop our machine learning skills. From business point of view, this model would be helpful for people who are into investment on houses and mainly who want to take help of internet and find best sales price rather than relying on agencies.

The purpose of our project is to build a neural network model to predict the sales prices of each residential house in Iowa taking into account all the aspects that many users might not consider (dataset comprises of 79 variables) such as height of the basement ceiling or the proximity to an east-west railroad. Moreover, we will also carry out an evaluation of neural networks in estimating the sales price for our model

2 Previous Work

Many researchers have done work on the subject of predicting house prices like [1]. Here, the technique used was Linear Regression. The results were analyzed using RMSE, MAPE, MAE.

Another such work was [2] where technique used was regular hedonic regression. The results were anlayzed using MSE. The problem of predicting house prices using machine learning techniques has been implemented by many people/teams before. There is a competition on kaggle (https://www.kaggle.com/c/house-prices-advanced-regression-techniques) for the same. Most of the kernels posted on kaggle implement advanced regression techniques, random forests and gradient boosting to predict house prices.

These advanced regression techniques included Ridge Regression, Lasso Regression and ElasticNet Regression

The Ames Housing dataset used for our problem statement was compiled by Dean De Cock for use in data science education.

3 Literature Study

Machine Learning

The term Machine Learning refers to the concept of learning itself. Machine learning has emerged mainly from computer science and artificial intelligence, and draws on methods from a variety of related subjects including statistics, applied mathematics and more specialized fields, such as pattern recognition and neural computation. Applications are, for example, image and speech analysis, medical imaging, bioinformatics and exploratory data analysis.

Machine learning algorithms can be classified into 2 types of problems : Classification and Regression. Since this project is based on Regression, that would be elaborated in further sections.

The classification and Regression problems are again sub divided into Supervised, Un-supervised and Semi-supervised.

Supervised learning is a type of system in which both input and desired output data are provided. Input and output data are labelled for classification to provide a learning basis for future data processing. After learning from sufficient observations, algorithm must be able to distinguish and categorize unlabelled items.

Unsupervised Learning Here, the dataset would not have the target variables which we are planning to predict. It's like learning without a teacher. The model fitting in this problem set would be done differently when compared to Supervised Learning

Semi-supervised Learning. Here, we have a mix of Supervised and Unsupervised Learning. Few of the data points would be having labels - the target variable and few wouldn't have. These kind of problems would be approached in a completely different way.

Regression

Regression is a statistical approach for modelling the relationship between a predictor variable X and a response variable Y. It assumes there is a linear relationship between these two variables and we use that to predict a quantitative output. Specifically, it's a functional relationship between two or more correlated variables that is often empirically determined from data. The overall idea of regression is to examine two things: (1) does a set of predictor variables do a good job in predicting an outcome (dependent) variable? (2) Which variables in particular are significant predictors of the outcome variable, Three major uses for regression analysis are

(1) determining the strength of predictors
(2) forecasting an effect
(3) trend forecasting.

The different types of Regression are :

1. Linear Regression
2. Logistic Regression
3. Polynomial Regression
4. Stepwise Regression
5. Ridge Regression
6. Lasso Regression
7. ElasticNet Regression

Simple and Advanced Regression

Simple linear regression is a very simple approach for supervised learning. But, even though it's probably the simplest and most straightforward method, it is widely used and quite easy to interpret. In this approach, we explore the relation between input features and target variable (Here SalesPrice) with a linear equation. The algorithm fits the training set in a way such that it learns how to predict for the test set. This gives us an idea of how the model would perform with new data.

In advanced Regression techniques, we use more layers - Neural Net where many layers of processing is done on the input data and then predictions are delivered from the final layer.

4 DataSet Description

Dataset link :
`https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data`

There are 2 csv files - train.csv and test.csv
The training dataset (train.csv) comprises of categorical and continuous features along with some missing values. This data set describes the sale of individual residential property in Ames, Iowa from 2006 to 2010. The data set contains 1460 observations and a large number of explanatory variables - 80 (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) involved in assessing home values.

Example : Neighborhood, Lot Space, Year Built, RoofStyle and many other features that influence price negotiations.The target variable will be Salesprice which means the final price of each in Ames, Iowa. So in total it has 81 columns with 80 features and 1 target variable.

The test dataset (test.csv) has 1459 observations and 80 variables which are same as training set data. But it doesn't have the target variable - 'Salesprice'. So the total number of attributes it has is just 80.

5 Setup

The programming part for this project was done using Python language,
The Python version used was 3.5.2.
Since the work was distributed, 2 team members were using Ubuntu machine (16.04 version) in which Python 3.5.2 would be installed by default.
The third member used Python Notebook on Mac machine.

The packages used in Python are :

- pandas
- numpy
- tensorflow
- keras
- matplotlib
- sklearn

6 Proposed Plan

Regression is a technique for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. To predict the house prices, we had many features (independent variables). So this project would start off with analyzing the target (dependent) variable, how it's correlated to other independent features. We also put effort in cleaning this huge feature set so that predictions would be more stable. After cleaning, transformations are applied so that the independent features are on same scale in numerical terms.

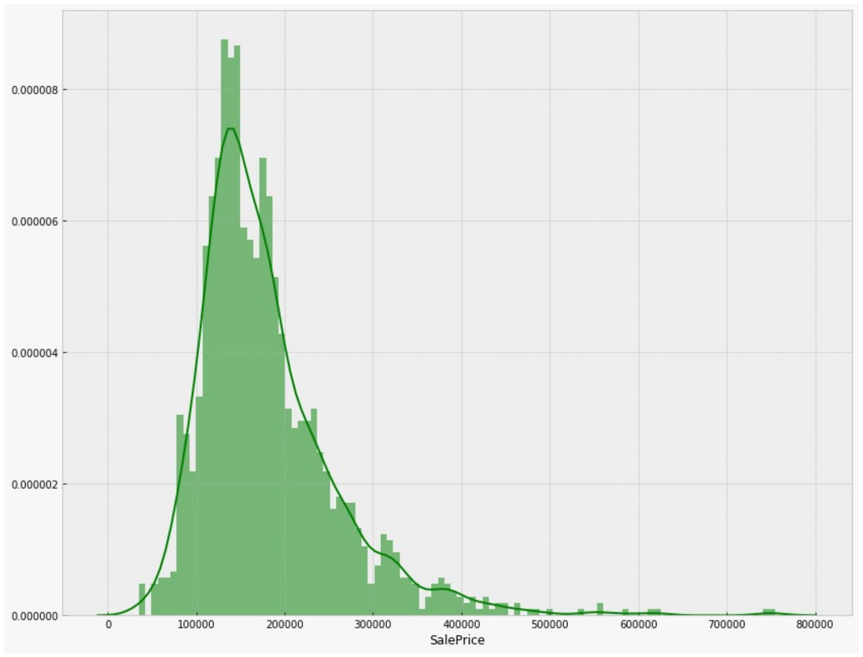
Now, the actual regression takes place on this preprocessed dataset, and several regression techniques were used. Baseline being Linear Regression, it's used to compare with advanced regression techniques like DNN regression

7 Plan and Experiment

7.1 Exploratory Data Analysis

7.1.1 Target variable Distribution

The distribution of the target variable can be understood by the following graph.



Target Variable Distribution

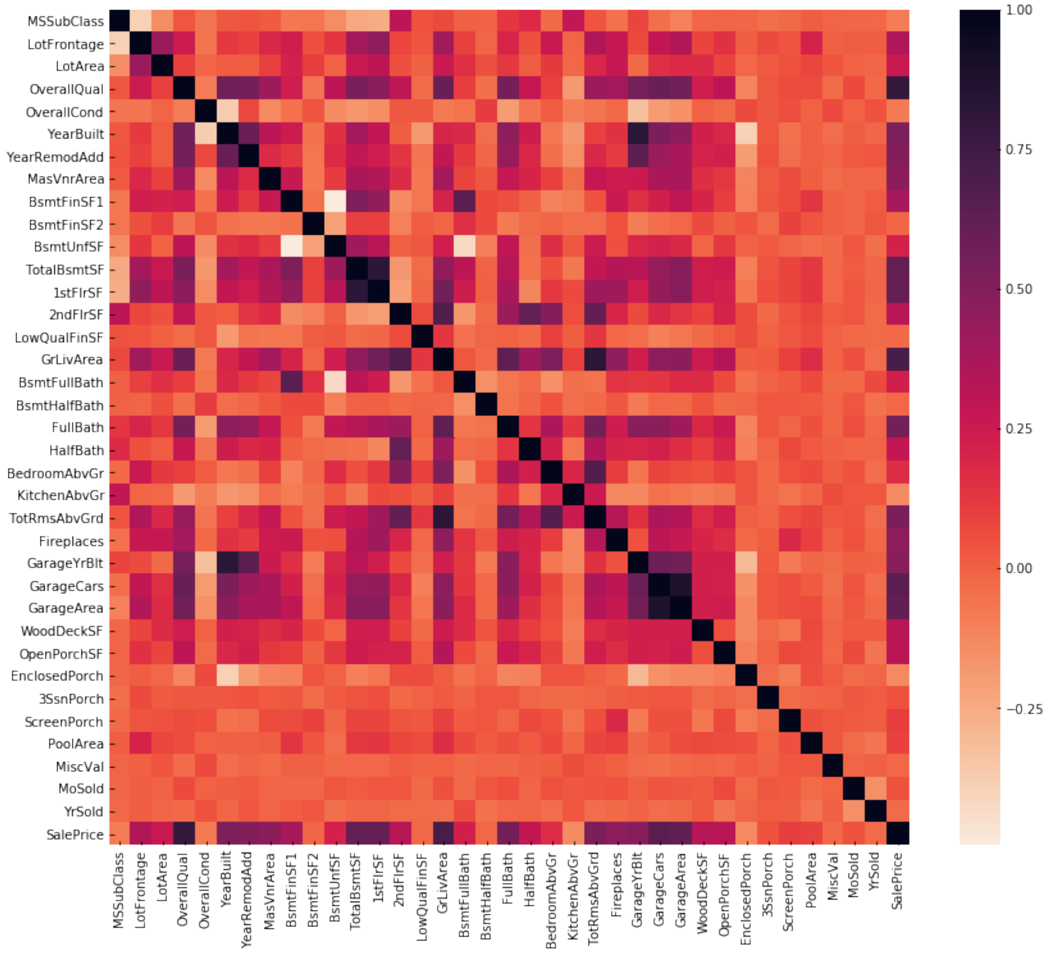
As we can see from the graph, the distribution is gaussian (Normal) distribution. There are few outliers which we can see after the value of 5000 on X-Axis. They are further discussed in next sections.

7.1.2 Correlation with Target variable

The top 10 columns which were found to be highly correlated with target variable were:

Feature	Pearson Coefficient
OverallQual	0.790982
GrLivArea	0.708624
2ndFlrSF	0.673305
GarageCars	0.637095
TotalBsmtSF	0.609681
GarageArea	0.608405
1stFlrSF	0.605852
FullBath	0.574563
TotRmsAbvGrd	0.533723
YearBuilt	0.522897

The heatmap for all the features corelation with target variable is shown below



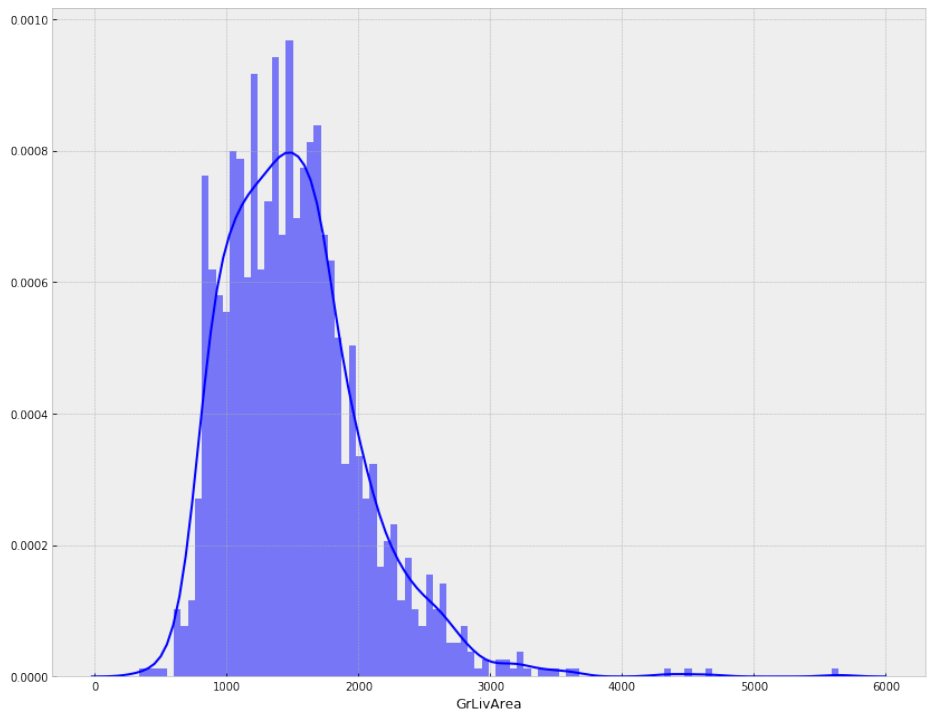
7.2 Data Cleaning

1. The top few features which are highly correlated to target variables were selected in training data. In case if they have outliers- they were removed.

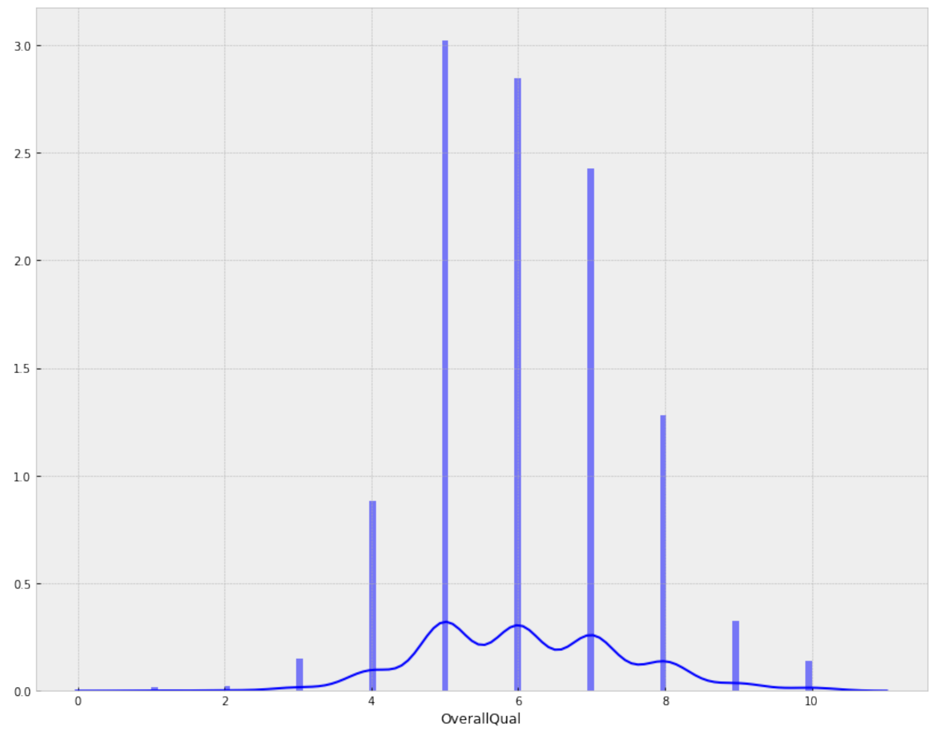
   Outliers might be existing in many features, but the concentration was on top 2 features which have high correlation with target variable.

   The Features that were selected in this step were : GrLivArea (Ground Living Area) and OverallQual (Overall Quality of the house).

   If the GrLivArea value was greater than 3000, they were removed. The graph below shows the distribution.



   If the Overall Quality of the house was less than 3 points, they were removed. The graph below shows the distribution.



NOTE : this step was applied only on train dataset.

2. Now, the column 'Id' was removed from train and test test since it's not required for analyzing the data. It's just a unique identifier for each datapoint.

3. The following columns were removed as they have many missing values (above 80%) from both train and test data sets
'PoolQC', 'MiscFeature', 'Fence', 'Alley'. This step was done after checking the value counts of each column.

4. If there are any missing values still existing in both train and test datasets, they are replaced with mode of that particular feature. The 'mode' was selected because replacing with the missing value with the value which is highly frequent wouldn't affect the distribution of data. For example, NaN values were filled with a string value of the most common value that was occurring, to prevent data from skewing.

5. Aggregating on few features : The columns : 'GarageArea', 'WoodDeckSF', 'OpenPorchSF', '3SsnPorch', 'PoolArea' all represent only area outside the house. So they are summed up and a new variable 'Outside-sqft' is created with the summed value.
Eventually the 4 variables used for summing were dropped.

   The columns : 'Basement_sqft', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF' all represent the area of basement in every floor. they are summed up and the the new feature 'Basement_sqft' was created with the summed up value. Eventually the 4 variables used for summing were dropped.

The columns : ’1stFlrSF’, ’2ndFlrSF’, ’EnclosedPorch’, ’ScreenPorch’ all represent the area in every floor. They all are summed up and a new column ’Inside_sqft’ is created with the summed value. Eventually the 4 variables used for summing were dropped.

This process was repeated on both train and test datasets.

7.3 Data Pre-processing

Once the data was cleaned, transformations were done on it.

Min-Max Scaling was applied on them. In Min-max scaling, the data is scaled to a fixed range - usually 0 to 1 which can suppress the effect of outliers. The following data pre-processing steps were followed:

1. Initially, the features which are categorical are separated from numeric ones. Using only the features which are numeric, Min-max scaling was applied on both train and test datasets.The scaler was fitted using train set and transformations were done using the same model for train and test datasets. These are stored as train_numeric.csv and test_numeric.csv.
2. Now, using only features which are categorical, the similar process of transformation was applied on both train and test set. These are stored as train_categorical.csv and test_categorical.csv. Before the transformations were applied, the categories were changed to continuous values using Label Encoding. Basically, a dictionary of items and a value for it would be built and all the discrete values are converted to continuous numeric values
3. Now all the features - continuous and categorical are used and similar process of transformation is done on both train and test datasets. These are stored as train_all.csv and test_all.csv

Finally, the structure of the above 6 datasets are:

train_numeric : 1417 rows, 28 columns

test_numeric : 1459 rows, 27 columns

train_categorical : 1417 rows, 39 columns

test_categorical : 1459 rows, 39 columns

train_all : 1417 rows, 67 columns

test_all : 1459 rows, 66 columns

Since the focus is on regression in this project, the categorical featured datasets were not used in any of the techniques although they were generated.

7.3.1 Regression

All the below algorithms were applied using a train test split of 2/3 and 1/3. The datapoints were selected randomly with a random state set to 42.

All the methods were applied on both :

1. When only Continuous Features were used

2. When both Continuous and Categorical Features are used

Linear Regression (Baseline Model)

Before starting with experiments and results for complex and advanced neural network models, a baseline regression model is built which is a Simple Linear regressor.

Here again, the experiment was done using first only continuous values dataset and then for both categorical and continuous. As this is simple model and no layers would be there, just fitting of selected input features with target variable which is SalesPrice here, is done on training data.

Then the same model would be used to predict the output variable for validation and test data. This informs us about the performance of our model. MSE was calculated for both the experiments. It turned out to be 0.002765 for continuous values and 0.002423 for both continuous and categorical values respectively.

DNN Regression

After experimenting with Linear Regression, we implemented Deep Neural Networks to predict the Sale price. Our goal of this experiment and the next was to check if any non-linear dependency would help in predicting the target variable. For this purpose, we employed ReLU (Rectified Linear Unit) activation function to introduce non-linearity in the process.

Weights Initialization:

DNN regressor uses Glorot Uniform Initializer (also called Xavier Uniform Initializer) by default. It draws samples from a uniform distribution within $[-limit, limit]$ where limit is $sqrt(6/(fan\_in + fan\_out))$ where fan_in is the number of input units in the weight tensor-1 and fan_out is the number of output units in the weight tensor-2. Training neural networks of any kind face a lot of issues when the weights are not initialized well. If the weights are initialized too small, they might converge to zero after a few epochs. If too large, some weights will sky-rocket whereas others will drastically drop down to zero.

Optimizer:

DNN regressor uses AdaGrad by default. In gradient descent we simply update the weights using the learning rate and the gradient calculated with respect to loss. In AdaGrad, we add element-wise scaling of the gradient based on the historical sum of squares in each direction. Pseudo Code for AdaGrad:

```
$grad\_squared = 0$
$while(True)\{$
$dx = compute\_gradient(x)$
$grad\_squared+ = dx * dx$
$x- = learning\_rate * (dx)/\sqrt{(grad\_squared + 10^{-7})}$
$\}$
```

AdaGrad is designed in such a way that features that are more sparse in the data have a higher learning rate which translates into a larger update for that feature. One disadvantage of AdaGrad is that due to accumulation of gradients along the iterations, it becomes very large and hence each parameter update during the later iterations is very small.

Neural network architectures:

We experimented with multiple neural nets to arrive at an optimum structure. Our final optimal network would be chosen by the mean square error on validation set and the completely unseen test set given by kaggle.
As discussed above in section 6.3, we created two types of datasets: only numerical and both(numerical + categorical). We will be training neural networks on both types of datasets.
In both cases we implemented 1) complex, 2) medium and 3) small sized networks.

1. Complex model (deep network with high number of neurons):
   Number of neurons in hidden layers respectively: $[100, 50, 25, 12, 6]$

2. Medium complexity model (deep network with high/less number of neurons):
   Number of neurons in hidden layers respectively: $[50, 25, 12, 6, 3]$

3. Small network:
   Number of neurons in hidden layers respectively: $[20, 10, 5]$

Shallow Neural Networks Apart from implementing Deep Neural Networks, we also tested if shallow networks could do the trick. We wanted to check our previous models were overfitting by any chance. We trained neural networks with just one hidden layer with the following number of neurons: $[5, 12, 25, 50, 100, 200, 400]$

8 Results

Using Continuous Features only

The below table shows the results of MSE (Mean square error) for all the above methods when only continuous variables were used (train_numeric.csv) dataset.

Regressor	MSE	Neural Net Layers	Number of Neurons in each layer
DNN Regression (complex)	0.0026	5	100, 50, 25, 12, 6
DNN Regression (medium)	0.0040	5	50, 25, 12, 6, 3
DNN Regression (small)	0.0038	3	20, 10, 5
Shallow Regression	0.0063	1	5
Shallow Regression	0.0035	1	12
Shallow Regression	0.0028	1	25
Shallow Regression	0.0038	1	50
Shallow Regression	0.0027	1	100
Shallow Regression	0.0026	1	200
Shallow Regression	0.0026	1	400
Linear regression	0.0028	–	–

Using Continuous and Categorical Features

The below table shows the results of MSE (Mean square error) for all the above methods when both continuous and categorical variables were used (train_all.csv) dataset.

Regressor	MSE	Neural Net Layers	Number of Neurons in each layer
DNN Regression (complex)	0.0020	5	100, 50, 25, 12, 6
DNN Regression (medium)	0.0027	5	50, 25, 12, 6, 3
DNN Regression (small)	0.0016	3	20, 10, 5
Shallow Regression	0.0021	1	5
Shallow Regression	0.0018	1	12
Shallow Regression	0.0019	1	25
Shallow Regression	0.0019	1	50
Shallow Regression	0.0019	1	100
Shallow Regression	0.0019	1	200
Shallow Regression	0.0020	1	400
Linear regression	0.0024	–	–

9 Observations

The following artifacts while observed while experimenting with regression :

1. Roofing a home with clay tile removes the most value. Interestingly, being next to a park or other outdoor feature also reduces the value of the home. Alternately, there are a few neighborhoods that increase the value. The most valuable feature in this case is GrLivArea.

2. Mean Squared Error (MSE) is a common loss function used for regression problems (different than classification problems).

3. Evaluation metrics used for regression differ from classification. A common regression metric is Mean Absolute Error (MAE). When input data features have values with different ranges, each feature should be scaled independently.

4. As we increase the number of neurons in a single layer in shallow net in both (using only continuous and using both categorical and continuous), we observe that after a point, the MSE stops deceasing. And after this point, it might increase or remain the same. While only continuous features are used, the MSE remained same after neurons were increased from 200 to 400, while when 2 kinds of features were used, the MSE is increased by 0.0001.

10 Conclusion

When regression is applied to a set of features, the more deep the network is, the predictions might vary slightly more than actual ones - MSE would be more. Shall networks or a deep network with minimum layers and neurons are seeming to perform better than deep neural networks considering this particular dataset.

It's clear that the more complicated the model is, it might learn even noise and outliers and thus perform poorly on unseen data.

Before applying any regression techniques, all the data has to be in proper structure and more than half of the effort goes for this task.

Considering Linear Regression as a base model, we have concluded the following things.

1. A small network would perform better, if there is not much training data, with few hidden layers to avoid over-fitting.

2. Early stopping would be a useful technique to prevent over-fitting.

3. When we use only continuous features the MSE values are little more than what we get when both kinds of features are used.

4. When we use both continuous and categorical features, the results are much better. The MSE values are low compared to when only continuous features are used. When only 1 hidden layer was used (Shallow net), the highest MSE was 0.0020 while this is the least when only continuous features were used.

5. The model when both kind of features were used performing better than only when one kind of features were used is pretty much obvious because the more features we use, the training and fitting would be better. This affirms when we are not using reluctant features. In this case, since we applied dimensionality reduction, we wouldn't have so many reluctant features.

11 Acknowledgments

A special thanks to North Carolina State University and especially Computer Science department. We would also like to thank Dr.Dennis Bahler and Teaching Assistant Jianxun Wang for constantly guiding us and providing feed back at every stage of the project. Your support was a driving energy for our accomplishments.

12 References

[1] Malang, East Java, Indonesia, "Modeling House Price Prediction using Regression", Year : 2017

[2] Hao Wu , Hongzan Jiao , Yang Yu, , Zhigang Li , Zhenghong Peng , Lingbo Liu, Zheng Zeng , "Influence Factors and Regression Model of Urban Housing Prices Based on Internet Open Access Data", Year : 2018

[3] Petra Vidnerova , Roman Neruda "Evolving keras architectures for sensor data analysis ", Year : 2017 Federated Conference on Computer Science and Information Systems (FedCSIS)

[4] Xavier Glorot, Yoshua Bengio "Understanding the difficulty of training deep feedforward neural networks"
`http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf`

[5] Hrushikesh Mhaskar, Tomaso Poggio, "Deep vs. shallow networks : An approximation theory perspective", Year : 2016

[6] Tomaso Poggio1 Hrushikesh Mhaskar2, 3 Lorenzo Rosasco1 Brando Miranda1 Qianli Liao1, "Why and When Can Deep-but Not Shallow-networks Avoid the Curse of Dimensionality: A Review", Year : 2016

[7] O. Delalleau, Y. Bengio, "Shallow vs. deep sum-product networks. In Proceedings of Advances in Neural Information Processing Systems", Year :2011

13 Appendix

Few things were changed after we submitted the proposal because, as we moved along with the project, our approach towards it changed little.

Here are few changes :

1. We mentioned that only data preprocessing would be done, but extensive cleansing of data was also done as it was required to get a reliable model.
2. We mentioned that all the features would be used. But we tried to subset into 2 datasets - only only containing numerical data and the second containing all features. Since we applied Regression techniques, dealing only with numerical data gave us some interesting insights.
3. We included a baseline regression model - Linear Regressor so that we can compare these results with the Advance Regression techniques.

14 GitHub Link

The link to this Project repository is:

`https://github.ncsu.edu/hsmalapa/AML-Project`