**Housing Trends Prediction**

**Abstract:**

The major problem these days is that people in housing sector don’t get to know that investment on which particular house would give them long term benefits. More precisely investment on which particular house would be the most feasible option. Considering these issues we thought to do a research with a purpose to provide customers with a system which would enable them to know about the prices of the houses in future. It will enable them to make calculated decisions to invest on a particular house. Moreover another purpose of this project is to provide customers with houses which have similar characteristics while buying or renting within a particular Zip Code. This feature would enable them to have multiple options to choose from. The solution which we proposed was to build models on Deep Neural Network Techniques and Machine Learning Techniques. We used Artificial Neural Networks and Support Vector Machine (SVM) to predict house prices beforehand with in a particular Zip Code.

**Introduction:**

This paper brings together latest techniques and methodologies for house price prediction which were not adopted before to take calculated decisions for investment in housing sector. Before this either people used to randomly decide to buy which particular house or they didn’t use efficient techniques, required to build models which resulted in unreliable results[1]. It was a major problem because people in housing sector didn’t get to know that investment on which particular house would be the most feasible option.

The main difference between this and other researches is that people used to build models using huge sizes of historic datasets to predict house prices[1]. What we did was that we divided the data of a particular state into Cities and Zip Codes. Our final Model predicted House Prices with respect to each Zip Code. This enabled us to get into depth of housing trends within every Zip Code. It enabled us to shortlist houses which have similar characteristics while buying or renting. Ultimately this can provide customers with multiple options to choose from.

Apart from that the algorithms which we used for house price prediction were very different and efficient compared to traditional linear algorithm[2]. Linear algorithms became too main stream for prediction and as the size of data is increasing rapidly day by day so we thought to build a model which would work in a more efficient way compared to the traditional ones. So, we decided to go with Artificial Neural Networks (ANN) and Support Vector Machine (SVM) models. Detailed comparison is carried out between both the algorithms in this paper and the best algorithm is chosen ultimately.

**Problem Formulation:**

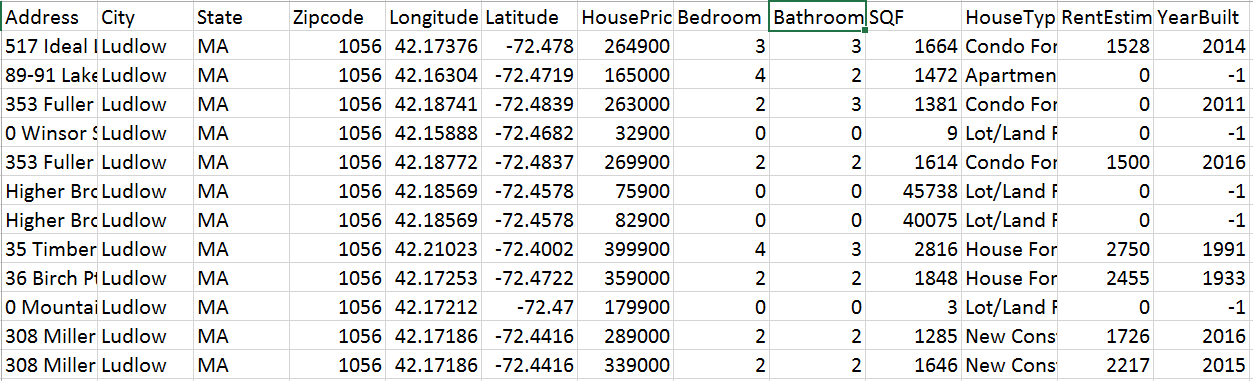
**Decision Variables and Data Set Description:**

We had 2 separate Zillow house datasets which we used for price prediction. Both the datasets were from Massachusetts but they were having different cities and zip codes. In these dataset we had a total of 13 attributes. Out of which some were numerical and some were textual attributes. These final 13 attributes are as follows:-

1. ID
2. Address
3. Zip Code
4. City
5. Bedrooms
6. Bathrooms
7. Estimated Price
8. Area Space
9. Year Built
10. Estimated Rent
11. Mortgage
12. Price Per Square Feet
13. Status of House(Condo/Town House/ Full house)

**Pre-Processing and Partitioning of data set:**

In our datasets there were a number of records which had Rent Estimate to be zero. To fill up those numerical missing values we used the median technique to impute them. We confirmed this in the Data Exploration phase. We cannot simply impute these values with mean or median. It is always good to leverage the existing knowledge from the data to impute missing values. Following image explains the issue.

****

We imputed missing values based on the house price. For example - if the house price falls in the $100,000-$200,000 bracket, the median rent estimate is $1421. We calculated the median rent estimate based on its price bracket. We calculated this median based on price bracket because whenever the rent was increasing then house price was increasing too in our data set but there could be a scenario when rent estimate is not directly proportional to price. In that particular case we will be needing to come up with a more efficient technique. Following is a code snippet of how we achieved this:

****

We partitioned the data into training and validation set with a ratio of 80% training data and 20% validation data. We had no test data set for this research because it was a case of unsupervised learning.

**Dealing with outliers:**

In case of outliers we straight away removed those entries as it caused very absurd and variable end results. Moreover outliers have a big effect on accuracy too. So we decided altogether to get rid of them.

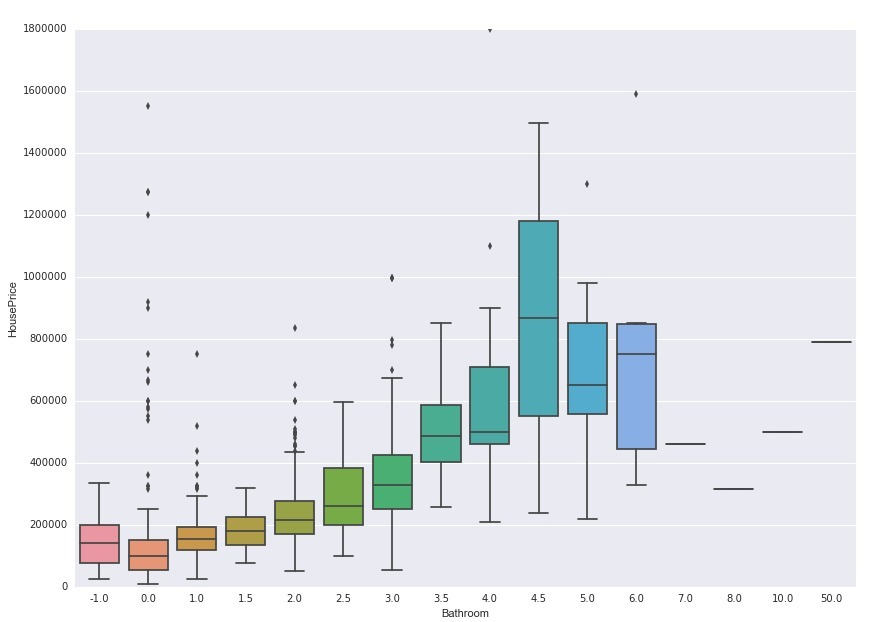
**Data Normalization:**

After that we used python’s standard scalar normalization to normalize the data. This enabled us to further improve model’s accuracy. We used Z-score normalization too but we were able to achieve better accuracy using standard scalar Normalization.

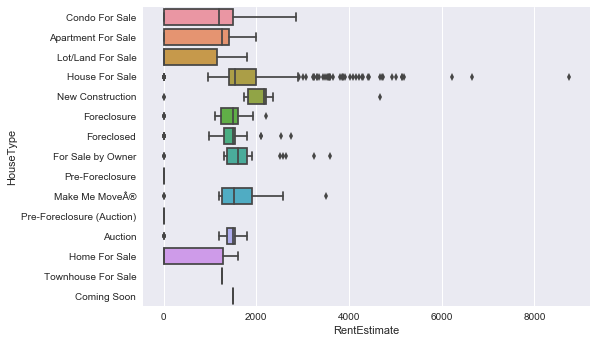
**Summarization and Visualization:-**

**Data Exploration:**

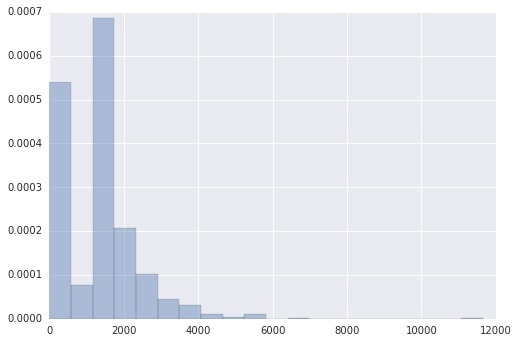
We started this project with certain assumptions about the dataset which we wanted to validate. The best way to do that is through Data Exploration.

**Bathrooms Vs House Price: -**

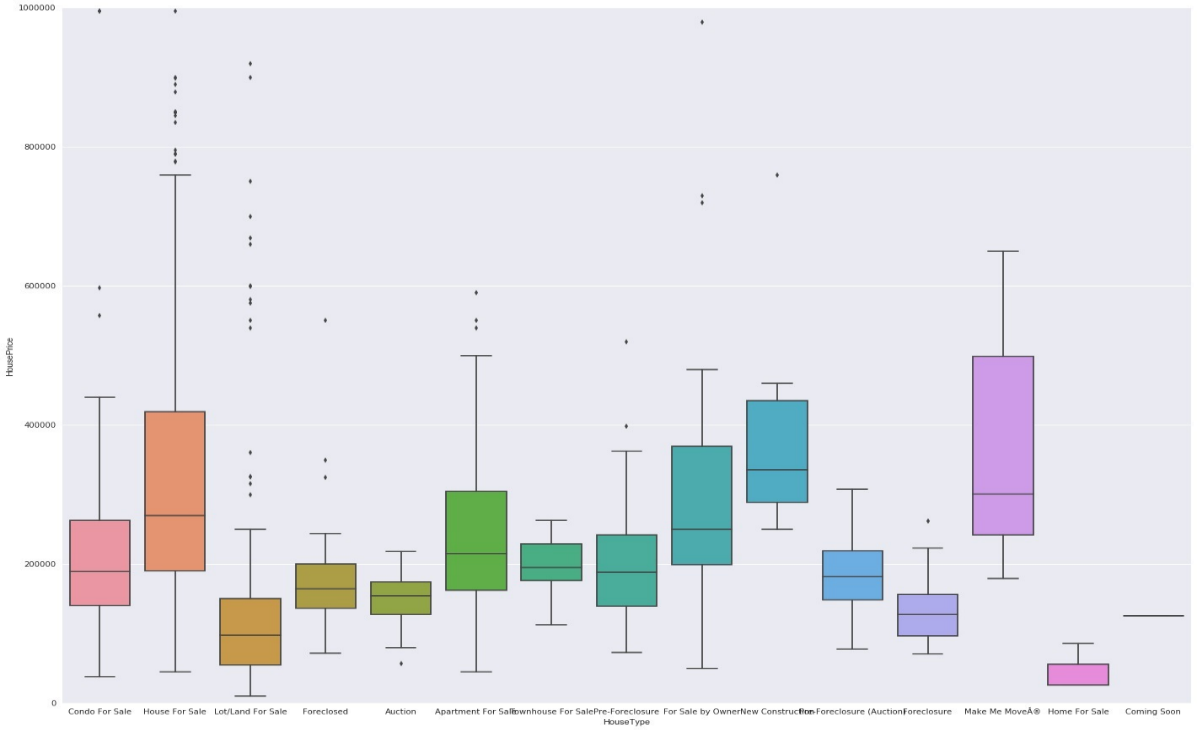
In the above visualization we can see that as the number of bathrooms increase, the house price increases which makes intuitive sense. It means that house with more number of bathrooms are expensive. We can see some outliers too which explains sparsity in the data. Outliers were dealt by the way stated previously in the paper.

**House Type Vs Rent Estimate Challenges: -**

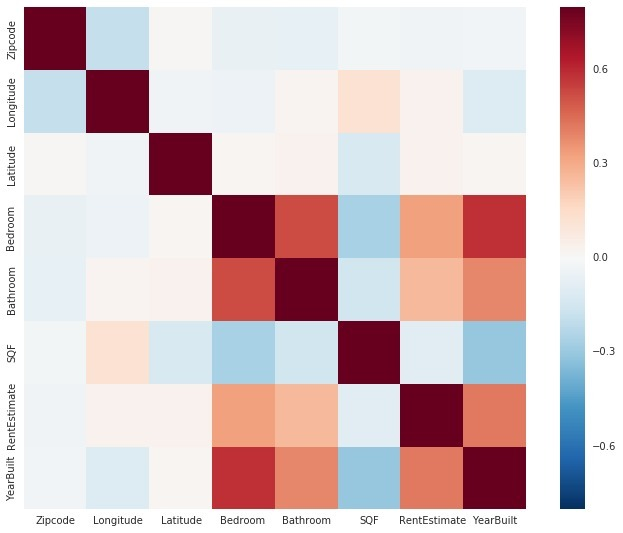
In this above plot there are a number of outliers in the House for sale section. Since the number is very high, we cannot simply remove them. We had to come up with an effective technique to impute the missing values which was explained in the previous section.

**Rent Estimate Plot: -**

For further analysis of Rent Estimate attribute we analyzed the above bar chart. This plot tells us about the situation of **missing values/0 values** in Rent Estimate attribute. As we can see the rent estimate has a lot of missing values. These bars show that majority of the values are zero. We used feature engineering techniques to impute missing ‘rent estimate’ values as stated previously.

**House Type Vs House Price:-**

We plotted this box plot to understand the distribution of house prices according to house types. In addition to the distribution of house prices, this plot also helps us detect outliers and gives us statistical metrics like mean, median, 25th percentile, 75th percentile, etc.

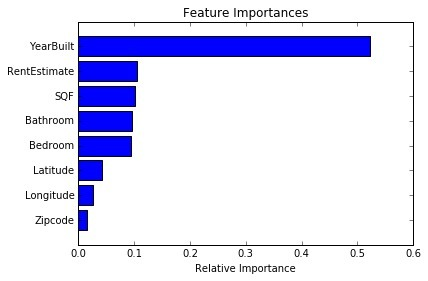
**Co-relation Heat Map: -**

The purpose of this heat map is to identify the correlation between all variables of our dataset. Bedroom is positively correlated to Year Built which makes intuitive sense. Other than that, there is no correlation which stands out.

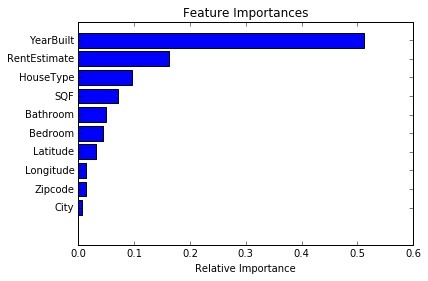
**Statistical summary:**

**Feature Selection:**

For feature selection we used Random Forest Algorithm. That algorithm gave us the importance of each feature in two separate datasets. Ultimately we went with all the important feature in both data sets as shown in the following figures. Feature with importance closer to 1 is the most important feature.

**Dataset 1:**

**Dataset 2:**

****

**Methods:**

Algorithms which we used for prediction were as follows:

* Artificial Neural Networks (ANN)
* Support Vector Machine (SVM)

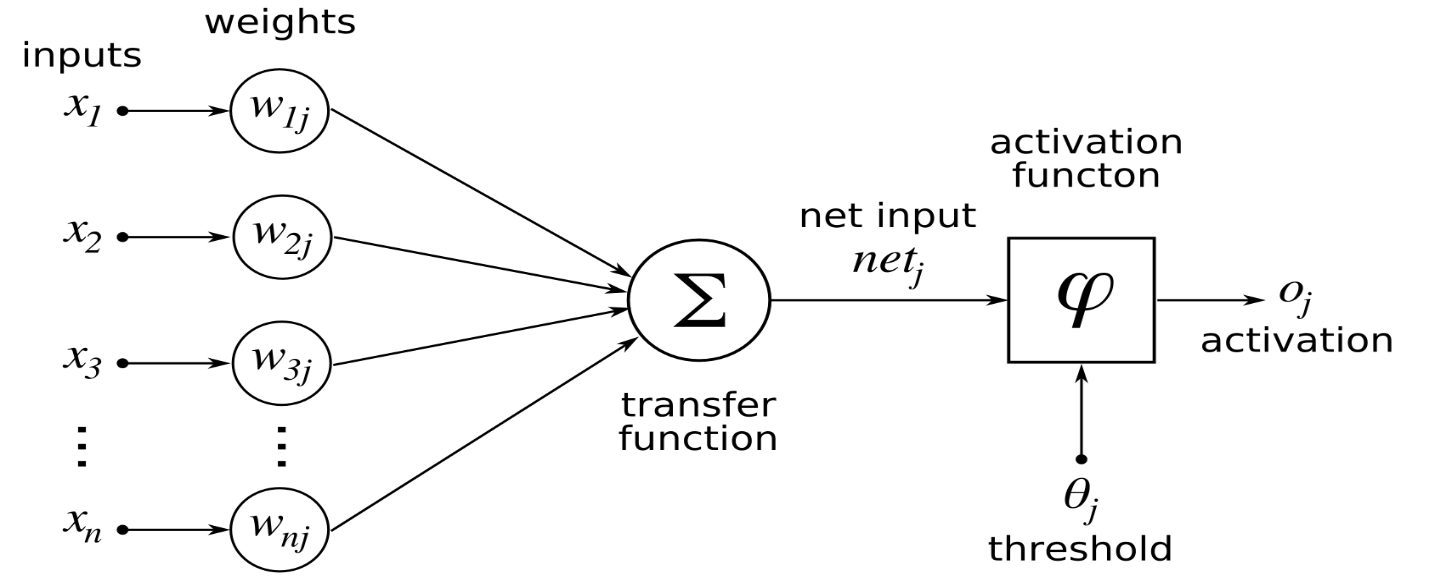
**Artificial Neural Networks (ANN):**

ANN is Machine Learning Algorithm which was designed to replicate the structure of a human brain. The Learning phenomenon of this algorithm is based on the phenomenon of actual human brain. This model is basically based on 3 different types of layers. Input Layers (attributes), Hidden Layers (Compute Layer) and Output Layer (Estimated House Price). ANN is interconnected network of neurons which adjust the weights of the connections between the units in response to input data[3]. The following Figure 1 explains the structure of a Neural Network with one hidden Layer.

**Figure 1**: ANN with one hidden Layer[4]



Every artificial neuron in the hidden layers has a set of input connections that receive signals from other neurons, a set of weights for input connection and bias adjustment. Then there is a transfer function that transforms the sum of the weighted inputs and bias to decide the value of the output from computational unit which in our case would be the estimated house price. Figure 2 Further explain this phenomenon.

**Figure 2:** Computation Unit[5]

The output of transfer function here is the summation of all inputs from each connection (Xi) times the weighted connection between node j and connection i. Furthermore, Output (Oj) here is from activation function (ϕ). Following equations 1 and 2 can be consulted for further explanation of this phenomenon.

Sumj = Σj (Wji) (Xi) [1]

Oj = ϕ (Sumj) [2]

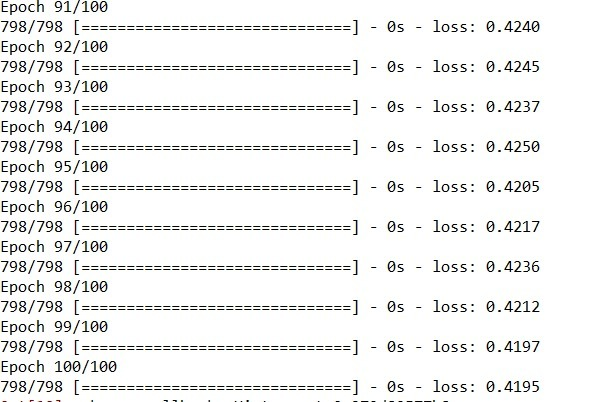
**Support Vector Machine (SVM):**

Support Vector Machine works on the concept of hyperplanes. SVM constructs a hyperplane or a number of hyperplanes in a high dimensional plane which can be used for classification or regression. It works by separation achieved by a hyperplane that has the largest distance to the nearest training data point. The ultimate goal is to reduce the generalization error and larger the margin between training points and the hyperplane the lower the generalization error.

**Results:**

**Artificial Neural Network (ANN):**

For ANN algorithm we used loss function as Root Mean Squared Error and we used Adam Optimizer with Neurons equal to the number of features which we considered 8. We used 100 Epochs here. Epochs are the number of time the algorithm back propagates to reduce the error/cost function.

 **Data Set 1’s Loss Function:**

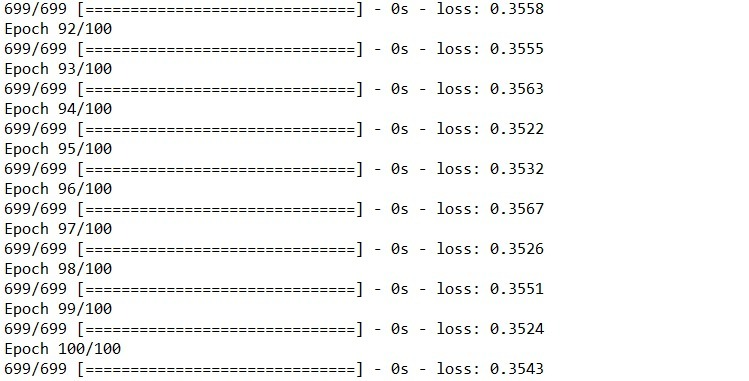
This above image depicts that with every epoch training loss function is decreasing. The last loss function value is 0.419

**Mean Squared Error:**

MSE value in data set 1 is 0.51

**Root Mean Squared Error:**

RMSE value in data set 1 is 0.644. Which is pretty less and it means algorithm is producing some very good results on data set 1.

**Data Set 2’s Loss Function:**

This above image depicts that with every epoch training loss function is decreasing. The last loss function value is 0.35.

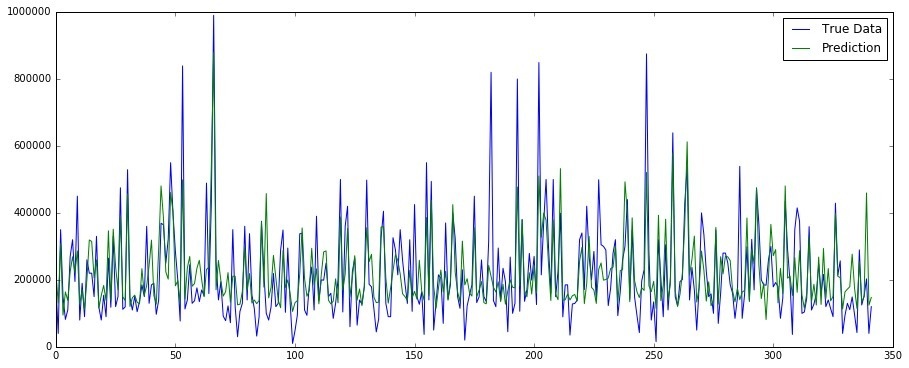
**Mean Squared Error:**

MSE value in data set 2 is 0.63.

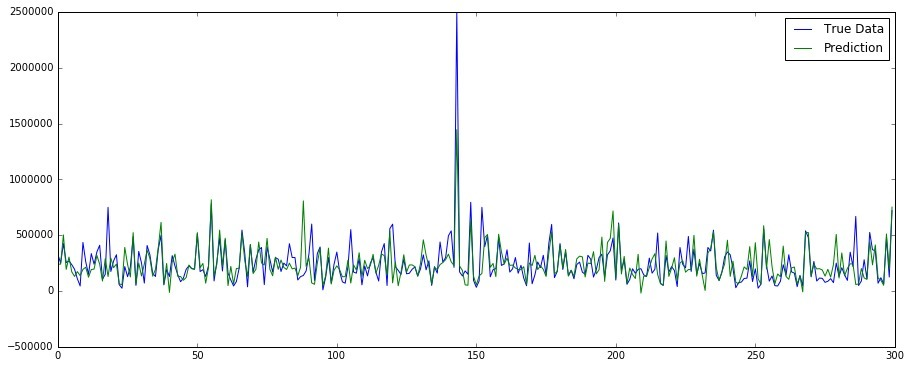
**Root Mean Squared Error:**

RMSE value in data set 2 is 0.6462. Which is also pretty less and it means algorithm is producing good results on data set 2.

**ANN Predicted Vs Actual Visualization:**

**Dataset 1:**

Here the blue line represents the original data and green line represents the predicted data. This plot shows a very respectable prediction trend. Which tells us that results are very reliable.

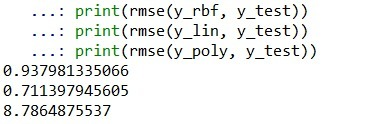
**Dataset 2:**

Here the blue line represents the original data and green line represents the predicted data. This plot also shows a very respectable prediction trend. Which tells us that results are very reliable.

**Support Vector Machine (SVM):**

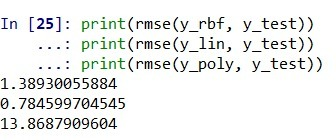
For SVM we used 3 different types of kernels linear kernel, Polynomial kernel and RBF kernel. Linear Kernel is used for linear hyperplanes, Polynomial kernel is used for fitted hyperplanes and RBF kernel is used for transformed hyperplanes. Apart from that we used 4 different cost functions which were C: 0.1, 1, 100, 500, 1000. The best accuracy we achieved was with C:1000. The following results are with 1000 cost function.

**Dataset 1:**

****

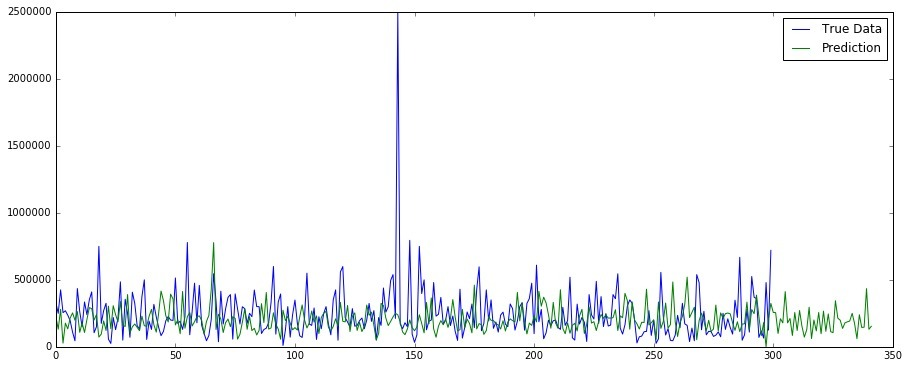
The least RMSE we got was 0.711 and that was with linear kernel with 1000 cost function value. Which tells us that linear kernel performs the best in data set 1.

**Dataset 2:**

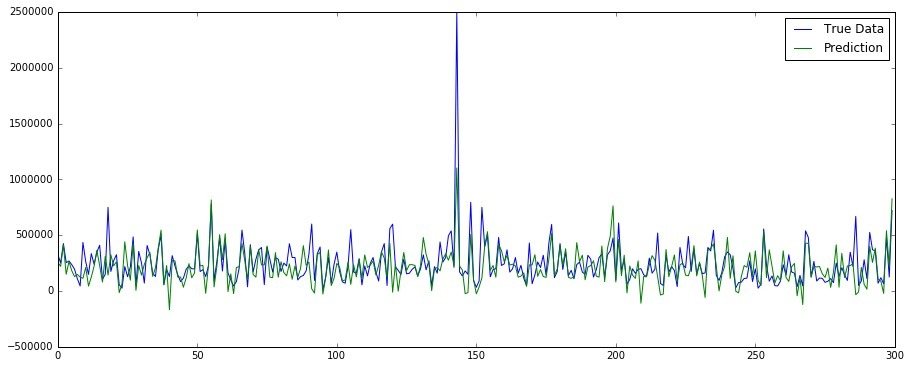
****

The least RMSE we got was 0.784 and that was with linear kernel with 1000 cost function value. Which tells us that linear kernel performs the best in data set 2 also.

**SVM Predicted Vs Actual Visualization:**

**Dataset 1:**

Here the blue line represents the original data and green line represents the predicted data. This plot shows a very respectable prediction trend. Which tells us that results are very reliable. We can spot one outlier here at around 2500000 price. The extra green line represents a longer prediction duration compared to the test set.

**Dataset 2:**

Here the blue line represents the original data and green line represents the predicted data. This plot shows a very respectable prediction trend. Which tells us that results are very reliable. We can spot one outlier here at around 2500000 price.

**Evaluation:**

**Artificial Neural Network (ANN):**

In Data Set 1 ANN produced RMSE of 0.644 using 8 Neurons, 100 Epochs and 2 hidden layers.

In Data Set 2 ANN produced RMSE of 0.6462 using 8 Neurons, 100 Epochs and 2 hidden layers. These results can vary with further parameter tuning.

So it means that ANN performs better on Data Set 1 as the value of RMSE in Data Set 1 is less as compared to Data Set 2.

**Support Vector Machine (SVM):**

In Data Set 1 the least RMSE we got was 0.711 and that was with linear kernel with 1000 cost function value.

In Data Set 2 the least RMSE we got was 0.784 and that was with linear kernel with 1000 cost function value.

So it means that SVM performs better on Data Set 1 as the value of RMSE in Data Set 1 is less as compared to Data Set 2.

**After the result evaluation of both algorithms it is very clearly interpretable that ANN performs better as compared to SVM and results produced from Data Set 1 are a little better as compared to results produced in Data Set 2.**

**Conclusion:**

This paper shows a comparison between the results of Artificial Neural Networks (ANN) and Support Vector Machines (SVM) in house price prediction. ANN and SVM were tested on two separate datasets and ultimately we deduced that ANN performed better compared to SVM. Moreover, results produced using Data Set 1 were also a little better compared to the results produced using Data Set 2.

Challenges which we faced during this research were related to

1. Dealing with inconsistency of data values.
2. Treating Outliers.
3. Dealing with interpretation of Neural Network Results.
4. Understanding and comparing the results from two different datasets.

There is, however, a limitation in this paper. We calculated Rent Estimate median based on price bracket because whenever the rent was increasing then house price was increasing too in our data set but there could be a scenario when rent estimate is not directly proportional to price. In that particular case we will be needing to come up with a more efficient technique.

**References**

[1] N. Bhagat, A. Mohokar, and S. Mane, “House Price Forecasting using Data Mining,” *Int. J. Comput. Appl.*, vol. 152, no. 2, 2016.

[2] R. E. Lowrance, “Predicting the Market Value of Single-Family Residential Real Estate,” PhD Thesis, New York University, 2015.

[3] V. Limsombunchai, “House price prediction: hedonic price model vs. artificial neural network,” in *New Zealand Agricultural and Resource Economics Society Conference*, 2004, pp. 25–26.

[4] “300px-Colored\_neural\_network.svg.png (300×361).” [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/thumb/4/46/Colored\_neural\_network.svg/300px-Colored\_neural\_network.svg.png. [Accessed: 31-Jan-2018].

[5] “ArtificialNeuronModel\_english.png (1682×799).” [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/6/60/ArtificialNeuronModel\_english.png. [Accessed: 31-Jan-2018].