Project Title: House Price Prediction

Sudipta Singha Roy, Ananda Mohon Ghosh, Arnab Mallik
Student ID: 251053613, 251014554, 251052315
University of Western Ontario
Email: ssinghar@uwo.ca, aghosh45@uwo.ca, amallik3@uwo.ca

Abstract: The price of the housing plays a vital role in modeling the current state of the economy. The price of the housing is determined based on the facilities, condition, total area and the location of the housing. This prediction is vital for the customers as well as the sellers. An individual customer needs to take a decision whether he/she should buy any housing at the sellers' demanded price. At the same time, this prediction system helps the seller to determine the price at which the owner should sell the housing. We will try to apply different regression models over house pricing data to predict the price of the housing based on the facilities, area and so on [1,2].

Problem Description:

House price prediction is a real-life problem that is often faced by the customer as well as the seller when any house-sell deal is made. The seller goes through a dilemma about setting the price of the house he/she wants to sell. The proper price setting is very important for the seller because of two issues. Firstly, over-price setting for any house will end up being an unsuccessful deal as the customer won't agree to buy the house at that price. Secondly, lower price than the usual one will make the seller to go through financial loss. Inadequate knowledge over the price of any housing can lead the customer towards a financial loss as well. So, having a proper idea about the price of the housing is important for both the seller and the customer.

A set of issues have the impact over the price of any housing. The location, the current condition of the housing, the facilities and utility services provided with it, number of bedrooms, total area of the residence, the condition and space of the kitchen and bathroom, number of fireplaces, area and type of the garage etc. defines the price of any housing.

The aim of this project is to predict the price of the housing based on the above-stated criterions and to find out which facts have more impact on deciding the price of the housings for any deal. For this reason, we have applied Gradient Boosted Regression Tree (GBRT) [3], Support Vector Regression (SVR) [4], Linear Regression [5], Neural Network [6], K-Nearest Neighbour [7], and Random Forest [8] models over the house pricing data for the prediction task and few visualization techniques to find out which criterion has how much impact over the price of the housing.

Dataset Description:

For the completion of this project, we have used "Ames Housing Dataset" dataset [9] which is available on Kaggle for predicting sales price of the houses and practicing feature engineering. Each data sample here has 79 variables which describes the aspects of any housing. The aim is to predict the prices of the houses analyzing these 79 attributes. There are in total 2917 samples in the dataset. These data samples are divided into two groups. 1717 samples are kept into the training set to train the models and the remaining are for evaluating the performance of the

model. Figure 1 shows the two-dimensional representation of the training data samples. From the figure it can be seen how much spread out the data are throughout the sample space.

Data Preprocessing:

Our dataset of 2917 observations were split into Training, Validation and Testing with a percentage of 60, 20, 20 accordingly. Initially there are 79 features. But, some of these data have some missing values in some attributes. Figure 2 shows some missing values in some attributes. For Numeric data, NA values were replaced with the average value of the feature. For categorical data, the highest occurring data was placed in place of NA. Then, in the dataset there were 59 categorical features. These features were broken into 0s and 1s using one hot encoding using dummies package. One sample one hot encoding is demonstrated in Figure 3. The resulting dataset had 243 features in total. This was done using boruta package in R. Boruta algorithm uses random forest. Then, for every feature, a corresponding shadow feature is created. The dataset is scrambled up and if the performance (with respect to output label) of shadow feature is better than the original feature, then that feature is deemed to be unimportant. This number of features would have made the model complex as well as too many features would have reduced the prediction capabilities of our model. Moreover, some features were redundant or not important. So, we tested which are the important features and which are not so. This was done using dummies package in R, which is based on random forest and creating shadow attributes. We could reduce the number of attributes to 84 in total. The feature selection process is demonstrated in Figure 4. In the figure, the blue ones in the image are important features. The red ones and the features with no value are discarded ones.

Regression Models for House Price Prediction:

This study tries to determine the price of the houses based on the given attributes using six regression models. In addition to this explicit intention, this study also tries to analyze the performance of the regression models over this dataset. The analyzed regression models in this study are: i) Gradient Boosted Regression Tree (GBRT) [3], ii) Support Vector Regression (SVR) [4], iii) Linear Regression [5], iv) Neural Network [6], v) K-Nearest Neighbour [7], and vi) Random Forest [8].

Simple, linear regression is used at first for the price prediction task here. This model draws a predictor in the data space to make the prediction.

Another model used in this study is the K-Nearest Neighbour (KNN). Number of neighbours determines how many boundaries it will create or how much similar data could be separated in a specific region. Lower value of K makes more flexible boundaries, but overfits as well. Higher number of K overcomes the problem of overfitting, but the model may face lower performance. So, choosing the optimal K which gives the most accurate prediction is the main challenge here. For our experiment the size of the K is set to 1 at first. Then, we tried 5, 10 and 15 as K's value. Till, K=10 the error was minimizing, but at K=15 the error was increasing again. So, the number of K was reduced again and at 13 the optimal error was found.

Like classification problem, Support Vector Machine (SVM) can also be used for regression problem and called Support Vector Regression (SVR). It also preserves the same principles of SVM which is maximizing the decision margin. While doing regression, an edge of tolerance, which is

called the epsilon, is approximated according to the problem. SVR tries to define individual prediction boundaries with maximum possible decision margin, at the same time, ensuring minimal error. In our experiment, the RBF kernel is used with epsilon equal to 0.2. Penalty parameter (C) belongs to the error term. The penalty is a squared "l2" (least squared) penalty. L2 Loss Function is utilized to limit the error which is the entirety of the all the squared contrasts between the genuine value and the anticipated value. The bigger this parameter, the less regularization is required for training the model. We tried different number of C, in the range of 1 to 1000000, and show how the performance of SVR differs with different C.

A simple neural network (NN) is used also for the regression problem. The architecture of NN contains one input, 2 hidden and 1 output layer. The input layer has 84 neurons to accommodate 84 input features. The first hidden layer has 64 neurons where the following hidden layer contains 32 neurons. Throughout the model ReLU activation function is used for non-linearity. The difference between simple NN for classification task and regression is that for classification task in the end softmax is used to normalize the values into probability so that the class label can be measured, where while using regression model this softmax isn't used as this time the predicted value should be a real number not the probability. We varied the learning rate for parameter tuning.

All the models described till now are skeleton models. Later we used two more models which are ensembled and tree-structured. The first ensemble model we used is Gradient Boosted Regression Tree (GBRT). GBRT is an ensemble model with multiple decision trees. These individual trees are dependent on the previous one. The first one tries to make prediction and then the following one tries to develop the model by reducing the over-all loss made by the previous one. The following one then works to minimize the loss made by the previous two models and thus the following trees works to minimize the overall cost of the model. The problem arises, as the increasing number of trees can leads the model towards overfitting. So, choosing the number of trees (or the number of estimators) is a vital issue for building the most suitable prediction model. Over any cost function multiple decision trees are optimized here to make the prediction or classification. Even multiple cost functions can be applied for different decision trees. In this study, GBRT is used for making prediction. Root Mean Square Error is used here as the cost function. The learning rate is set to 0.01. The depth of the individual trees is set to 2 to 35 and the number of estimators was set to in the range of 50 to 200. Here, number of estimators refers to the number of trees used to build the ensemble model.

Like GBRT, Random Forest is also an ensemble model with multiple decision trees. Unlike GBRT, random forest incorporates multiple independent decision trees. The constraint here is all the trees here are of equal shape and depth. Each individual tree is assigned with previously isolated subset of the main dataset for being trained. After the prediction made by each of them, their results are combined to make the final prediction. Random forest model faces overfitting when the depth is too high as the trees can then use the redundant information at different tests. The parameters of the random forest are the number of estimator and the depth of the trees. We tried different depths as well as different number of estimators for different trials. The range of depth was set in the range of 2 to 36 and the number of estimators was used in the range of 50 to 300.

Experimental Setup and Result Analysis:

For the price prediction task, we applied six different models with different parameters to generate the best possible outcome with minimal error. For all the models the root mean square error is used as the cost function.

For the prediction problem, we implemented the simplest model, linear regression, at first. Figure 5 shows the predictor drawn by the linear regression model throughout the training data samples. As the data samples are plotted in the two-dimensional space the predictor learned by the linear regression model throughout the training process is represented as linear line in the figure. The RMSE value over the test set found in this case was 41163.23764482.

The next skeleton predictive model we used was K-nearest neighbour. We started the experiments with K=1 (one nearest neighbour) and the RMSE value found for it was 54552.46557. Later we increased the number of K to overcome the problem of overfitting. We tried with K= 5, 10 and 15 then and the corresponding RMSE were 52695.489, 48229.9639 and 49136.2531. So, it is clearly observable that the RMSE at these scenarios are lower than at K=1. Till, K=10, the RMSE value was decreasing with higher value of K. But, at K=15 the RMSE value was slightly higher than the previous cases. As at K=15 the RMSE value is found higher than the value at K=10, we tried to find the best value of K in the range of 10 to 15 and at K=13 we achieved the best result over the test set. The RMSE value for K=13 is 47765.5413. Histogram in the Figure 6 shows the RMSE values for different Ks.

After linear regression and K-nearest problem, we implemented support vector regression (SVR) for the house price prediction task. We tried different values of C to allow different penalties. For C=1 (with minimum penalty allowed) the RMSE value was 59877.3122 which is much higher than the previous all models. Then we implemented SVR with 10, 100, 1000, 10000 and 100000 C values. With higher values of C the RMSE decreased. For C=100000 the best result for SVR over the test set was found and it was 52602.4655 which is the worst performance and much higher compared to all other models previously stated. That's why no more variation of C is tested further. Throughout the experiments, squared l2 penalty was used with fixed epsilon valued (0.2) RBF kernel. The RMSE values found for C= 1, 10, 100, 10000 and 100000 are represented in Figure 7.

The fourth and the final skeleton model used for the experiments is the neural network of the architecture 84-64-32-1. The weights were set randomly and throughout the learning process the weights were updated. During the training of the neural network the weights were gone through the adaptation process till 100 epochs. Figure 8 shows the RMSE loss value of the model for both training and validation up to 100 epochs. From the image it is clearly visible that both the losses were decreasing till iterations. And the RMSE value of both training and validation were almost equal which indicates the proper training of the model. Throughout the experiment, the learning rate was varied from 0.01 to 0.1 and the best training vs validation error was found for learning rate 0.01 which is demonstrated in Figure 8. Figure 9 (blue color: predicted price of the housing and red color: original price of the housing) shows the original vs predicted values of houses where the x-axis represents the index of different houses and in the y-axis the prices of them are represented. The RMSE value over the test set found with this setting was 41760.2730680.

The previously stated four model are skeleton models and among them the best performance was found for the simple linear regression model with RMSE value 41163.23764482. The other two

models used in our experiments are tree structured ensemble models. The first one applied was Gradient Boosted Regression Tree (GBRT). We tuned the two parameters: number of estimators (individual decision trees) and maximum depth (what should be the maximum depth of individual trees). As the individual trees tries to overcome the error made by the previous ones, with the increment of number of trees reduces the error. However, higher number of trees can also lead the model towards overfitting. So, to find out the optimal structure we tried for different number of estimators (50, 100, 150 and 200) and for each number of estimator six maximum depths (2, 5, 10, 15, 25 and 35) were tested. For all the experiments with GBRT the learning rate was fixed to 0.01. For 50 number of estimators the best performance was found at maximum depth 35 with RMSE value 33451.7896. Even for 100 estimators the best performance was achieved for maximum depth 35. The RMSE value was 33013.926 this time and this is the best performance we got from all the settings we applied. For both 150 and 200 estimators the best performance was shown for maximum depth 15 and the RMSE values were 33117.058 and 41338.902 respectively. Figure 10 shows the RMSE values for each setting. From the chart, it is clearly visible that the best performance was achieved by the setup number of estimators equal to 100 and maximum depth equal to 35. And this model's performance is better than the previously stated skeleton models. From this experiment we can also show the proof of two characteristics of GBRT presented in [3]. The first one is, with increasing number of estimators the best performance can be achieved at lower depth and the second one is, higher number of estimators can overfit the model and make the performance lower.

The final model we tried is another ensemble model named Random Forest. This model accommodates multiple same structured and independent small decision trees. The parameter we tuned here are also the number of estimator and depth. We tried six number of estimators: 50, 100, 150, 200, 250 and 300. For all the number of estimators we varied the depth of the decision trees (2, 5, 10, 15, 25 and 35). Figure 11 shows the RMSE values for all the setups. For 50 estimators the best RMSE value we found is for depth 35 and that is 32843.3894. Even for 100 estimators the best performance is found for depth 35 as well. This time the RMSE value we get is 31930.83318 which is better than the previous one. Then we tried with 150, 200, 250 and 300 estimators. In all cases the lowest RMSE values are found for depth 35. For 150, 200, 250 and 300 estimators the RMSE value at depth equal to 35 are 31769.660913, 31589.153589, 30558.721 and 30558.72. As for both 250 and 300 number of estimators the RMSE remains same, we tried with different number of estimators to find out from where the convergence started. And we got that for 227 estimators the random forest model started to show convergence. Then we lower the depth of the decision trees and got till depth 33 the result remained the same. For lower depth less than 33 the performance decreased then. So, the structure of 227 estimators with depth 33 gives the best performance with least complexity and we picked this structure as the final model.

Figure 12 shows the histograms of the errors of the models. To demonstrate here, we have chosen only the best setup of each individual regression models. From the histograms we can see that, the random forest model with 227 estimators and depth 33 has more frequency counts in the range of lower errors than the others which gives the support of the fact that this model has shown the best performance. For random forest model the maximum error is 150063.24 where the maximum errors for linear regression model, SVR, KNN, NN, and GBRT are 364400, 504024.21, 442040.59, 391062.28 and 280027.81 respectively. So, from the information in the histogram it is also visible that the outcomes produced by random forest is more likely to the original ones.

Table 1 shows the performance comparison of all the models used throughout the experiment. We tried various structures and parameters for all the six models, but for the result comparison we choose to show only the best one of each. From the table 1 it is clearly visible that, among the models both the ensemble models show better performance than the skeleton models. The reason behind this scenario is that, ensemble model incorporates the strengths of the individual ones to improve the performance. Among the skeleton models the simplest one, the linear regression model, shows the best performance. Though the training and validation curve of the neural network is close to each other and the error is very less, still the deep learning-based models require a lot of data which is not provided in this dataset. That's why its performance is not very good this time. Both GBRT and Random Forest showed very good prediction accuracy. Among them the performance of Random Forest is slightly better than the GBRT. The reason is still the amount of data in the dataset. GBRT works quite in the similar way as the deep learning-based models do. With more data, GBRT would have a chance to beat Random Forest.

Conclusion:

House price prediction problem is a real-life problem. The price of a house depends on a lot of criteria like condition of the house, location, facilities etc. In our experiment we implemented six regression models with different parameters to find the best-suited model. From our experiment we have concluded that, Random forest model with 227 estimators and depth 33 is the best suitable model among the tested ones. If it is possible to get a bigger dataset with more features, deep learning-based models can be applied here for the prediction task. For that, data needs to be augmented at first and then Recurrent Neural Network [10], Long Short Term Memory (LSTM) [1] etc. can be used.

References:

[1] X. Chen, L. Wei, J. Xu, "House Price Prediction Using LSTM", *The Computational Research Repository,* https://arxiv.org/pdf/1709.08432;House, 2017.

[2] Steven C. Bourassa, Eva Cantoni, Martin Hoesli, "Spatial Dependencies, Housing Submarket, and House Price Prediction", *The Journal of Real Estate Finance and Economics*, vol. 35, issue 2, pp. 143-160, 2007

[3] J. Elith, J. R. Leathwick, and T. Hastie, "A working guide to boosted regression trees", *Journal of Animal Ecology*, vol. *77, issue* 4, pp. 802-813, 2008.

[4] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression", *Neural Information Processing-Letters and Reviews*, vol *11, issue* 10, pp. 203-224, 2007.

[5] G. A. Seber, and A. J. Lee, "Linear regression analysis", *John Wiley & Sons,* vol. 329, 2012.

[6] D. F. Specht, "A general regression neural network", *IEEE transactions on neural networks*, vol. 2, issue 6, pp. 568-576, 1991.

[7] M. Maltamo, and A. Kangas, "Methods based on k-nearest neighbor regression in the prediction of basal area diameter distribution." *Canadian Journal of Forest Research*, pp.1107-1115, vol. 28., issue 8, 1998.

[8] Liaw, A. and Wiener, M., "Classification and regression by random Forest." *R news*, vol. *2,* issue 3, pp.18-22, 2002.

[9] Ames Housing Dataset, https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

[10] A. Petrosian, D. Prokhorov, R. Homan, R., Dasheiff, and D. Wunsch II, "Recurrent neural network based prediction of epileptic seizures in intra-and extracranial EEG", *Neurocomputing,* vol. 30, pp.201-218, 2000.

Appendix:

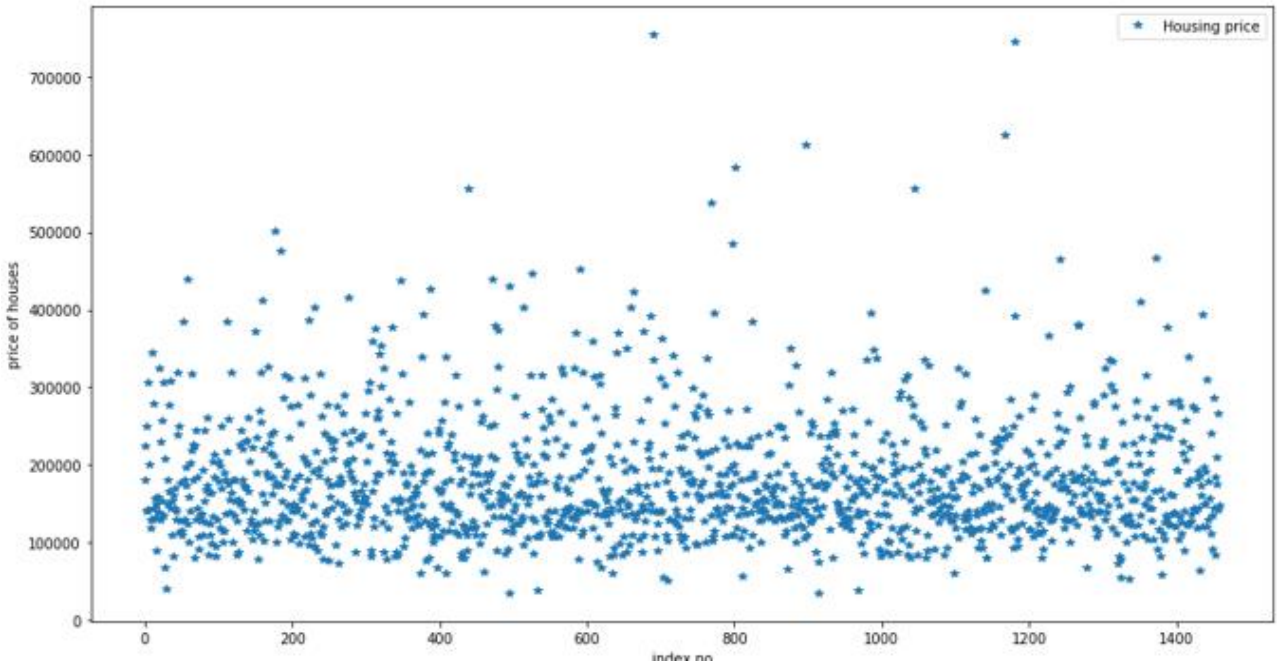

Fig. 1. Data distribution over the two-dimensional space

LotFrontage	FireplaceQu
65	NA
80	TA
68	TA
60	Gd
84	TA
85	NA
75	Gd
NA	TA
51	TA
50	TA
70	NA
85	Gd
NA	NA
91	Gd
NA	
51	
NA	
72	

Fig. 2. Missing values in the data

#	Street
1	Grvl
2	Pave
3	Pave

#	Street_Grvl	Street_Pave
1	1	0
2	0	1
3	0	1

Fig. 3. One-hot encoding to handle categorical data

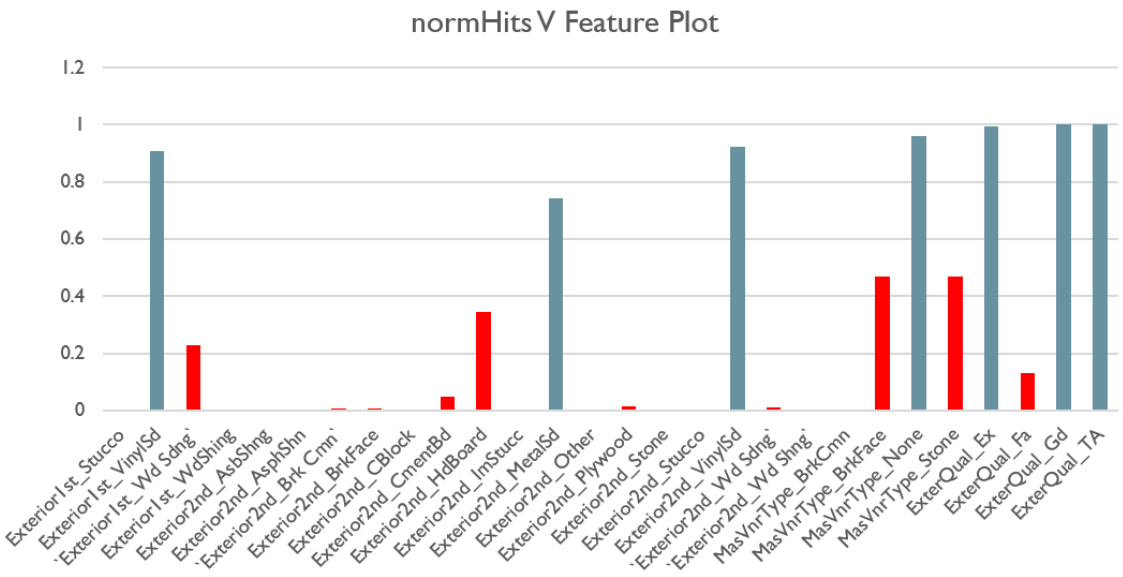

Fig. 4. Feature Selection

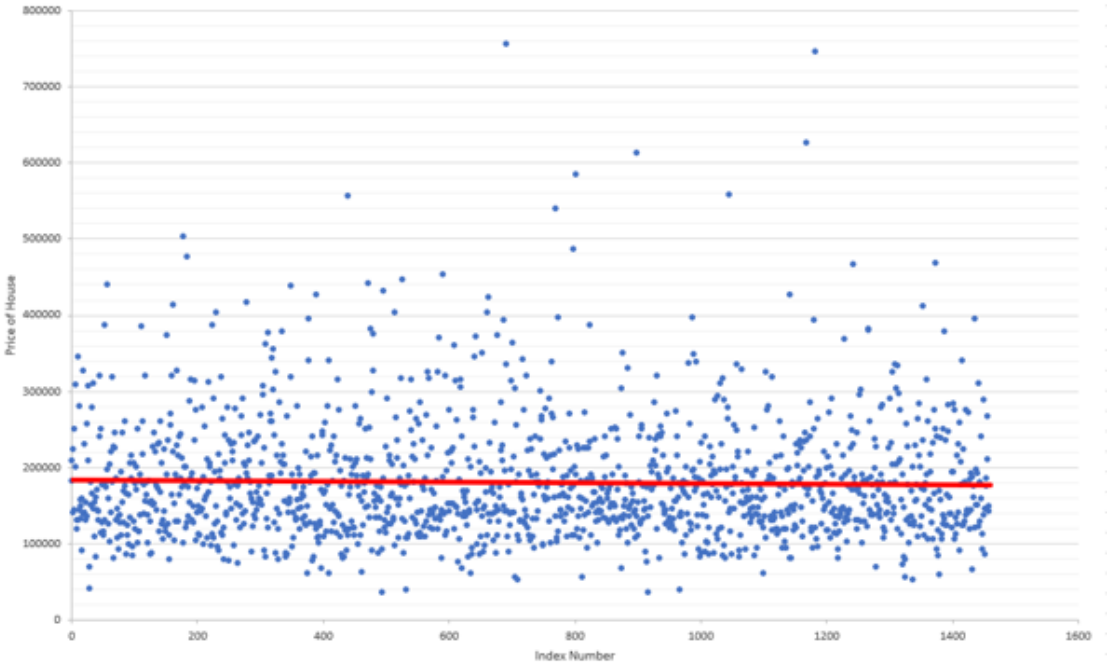

Fig. 5. Two-dimensional representation of the predictor drawn by the linear regression

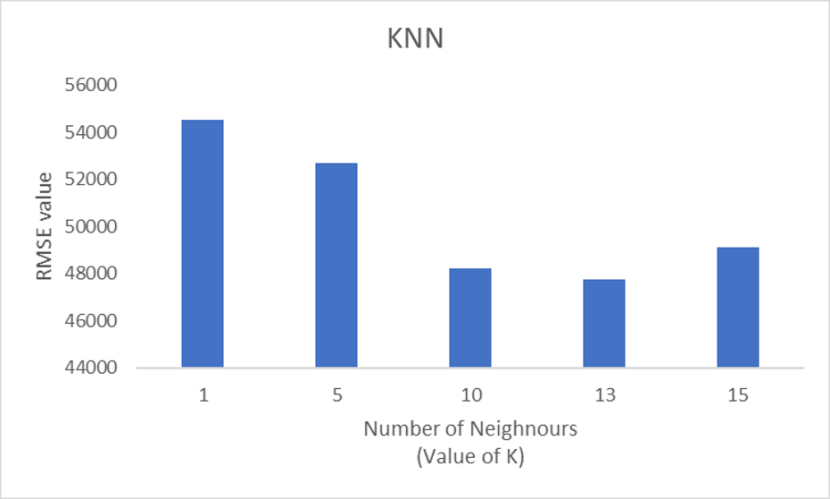

Fig. 6. RMSE values for different values of K with KNN

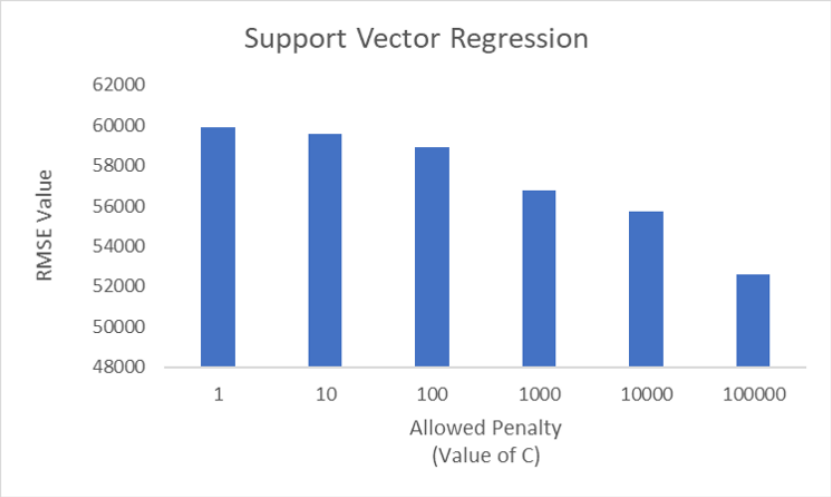

Fig. 7. RMSE values for different values of allowed penalty with SVR

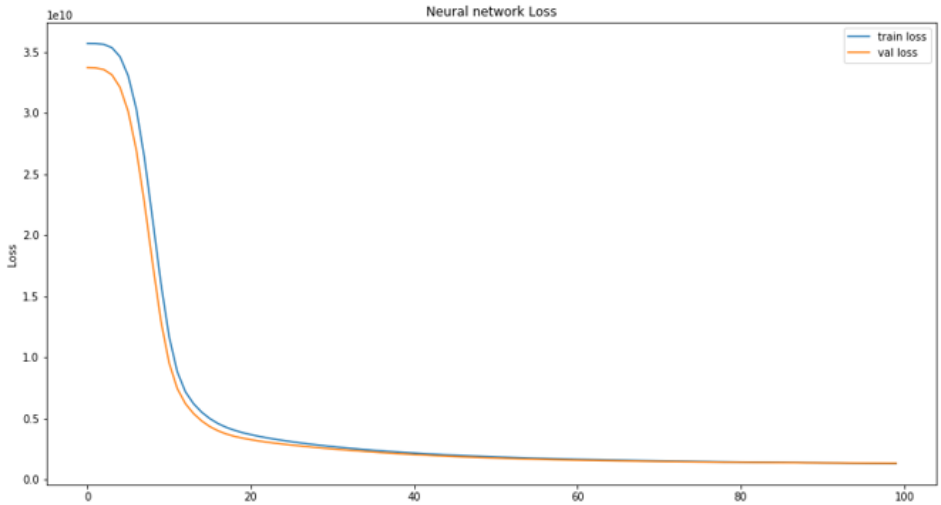

Fig. 8. Training vs Validation curve with epochs for neural network

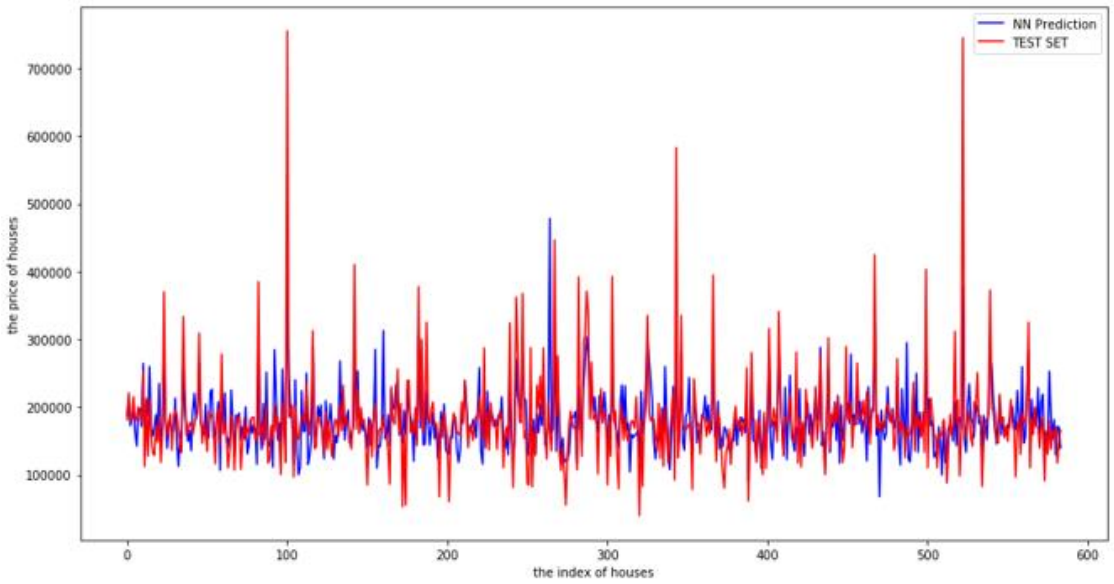

Fig. 9. Actual vs predicted prices of housing for neural network

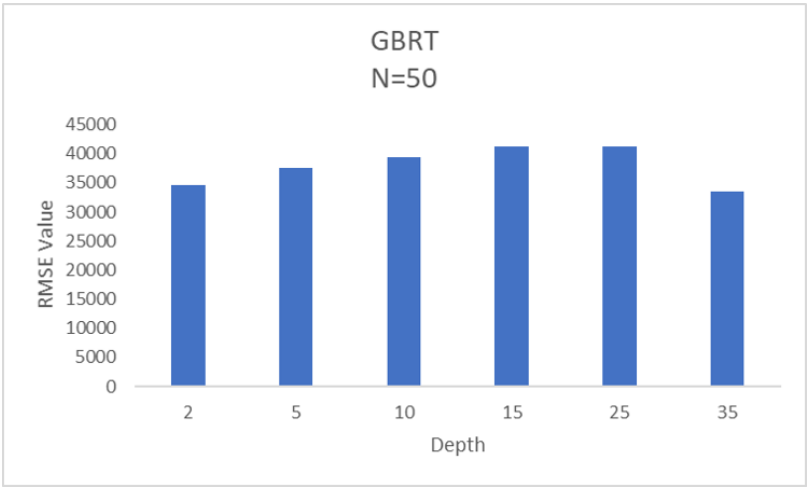

Fig. 10 (a). RMSE values for different depths of GBRT with 50 estimators

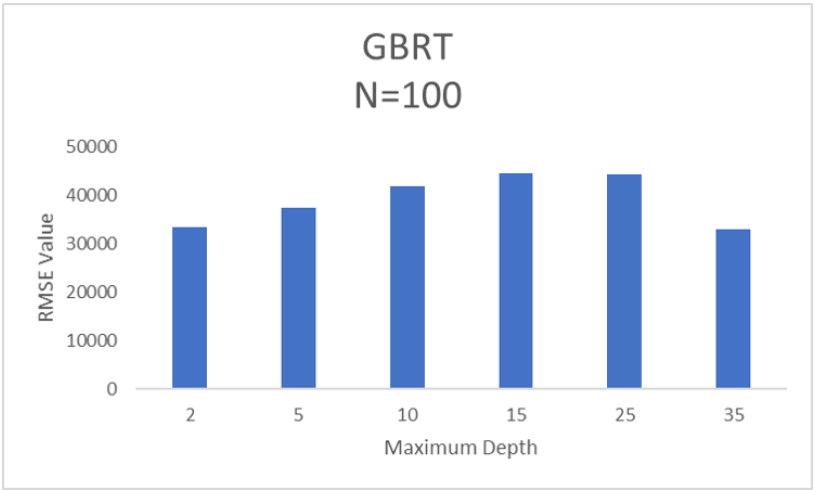

Fig. 10(b). RMSE values for different depths of GBRT with 100 estimators

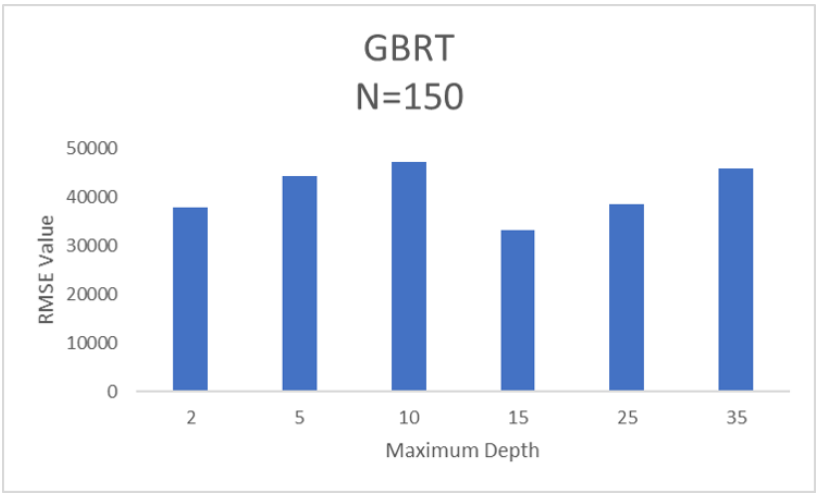

Fig. 10(c). RMSE values for different depths of GBRT with 150 estimators

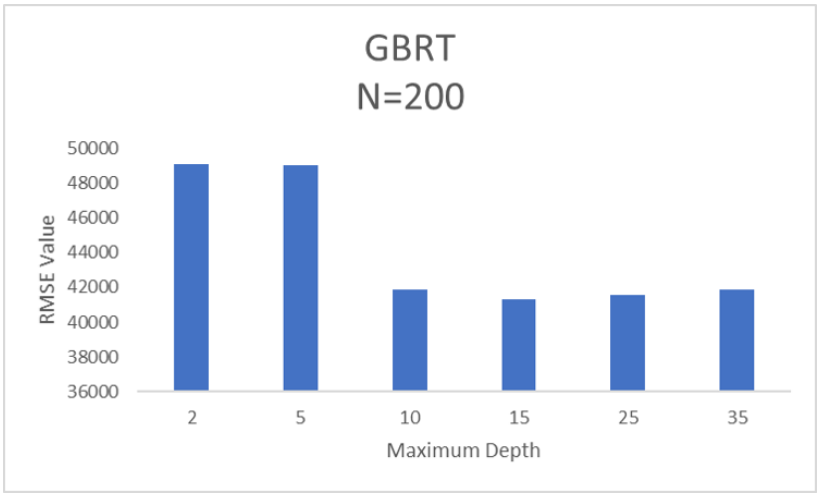

Fig. 10(d). RMSE values for different depths of GBRT with 200 estimators

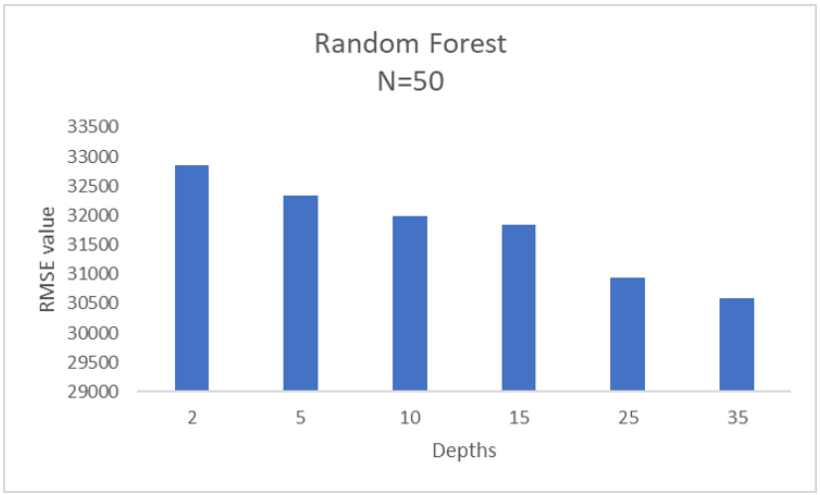

Fig. 11(a). RMSE values for different depths of Random Forest with 50 estimators

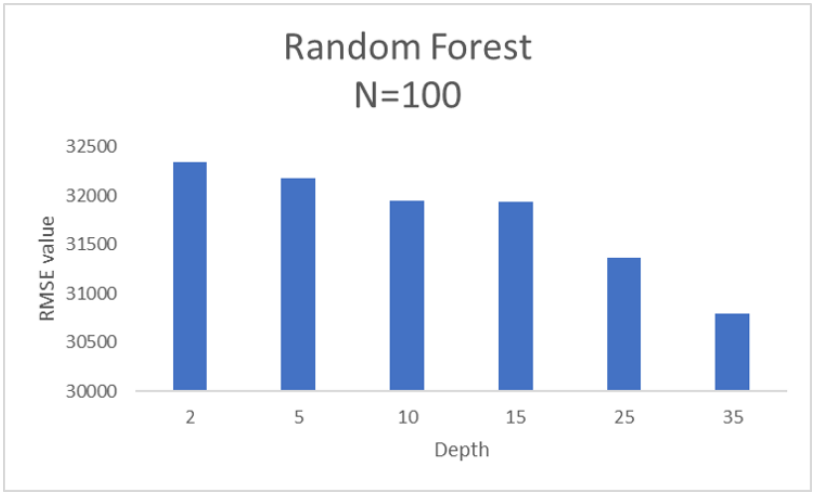

Fig. 11(b). RMSE values for different depths of Random Forest with 100 estimators

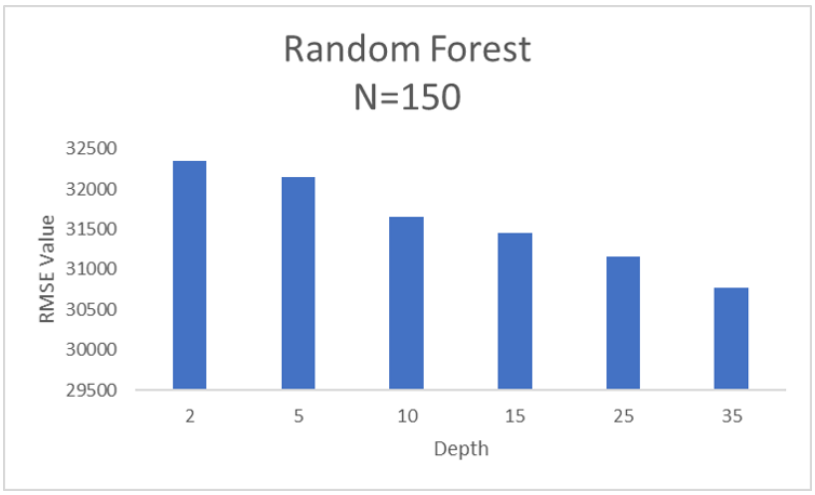

Fig. 11(c). RMSE values for different depths of Random Forest with 150 estimators

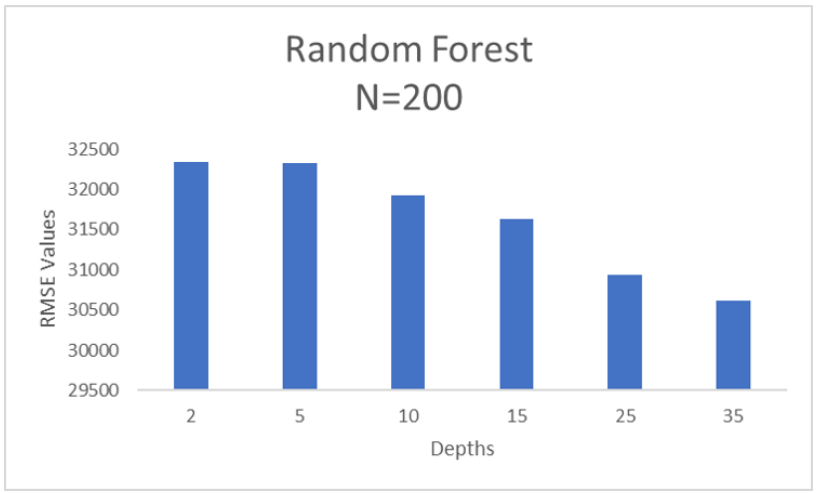

Fig. 11(d). RMSE values for different depths of Random Forest with 200 estimators

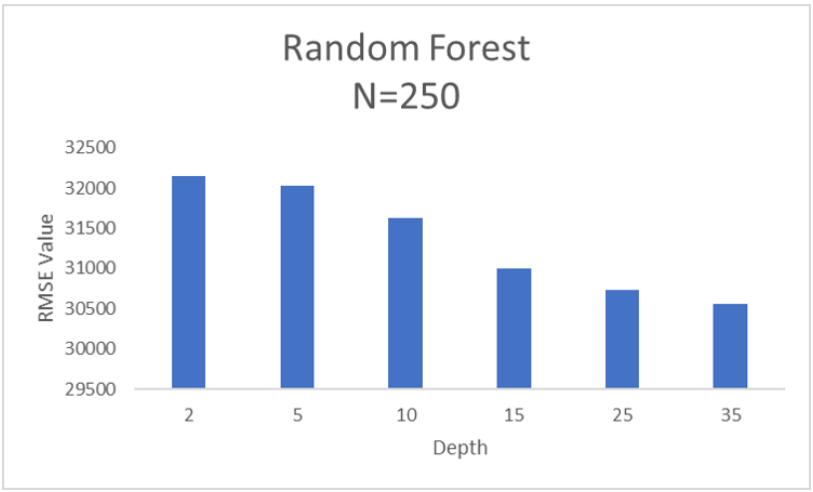

Fig. 11(e). RMSE values for different depths of Random Forest with 250 estimators

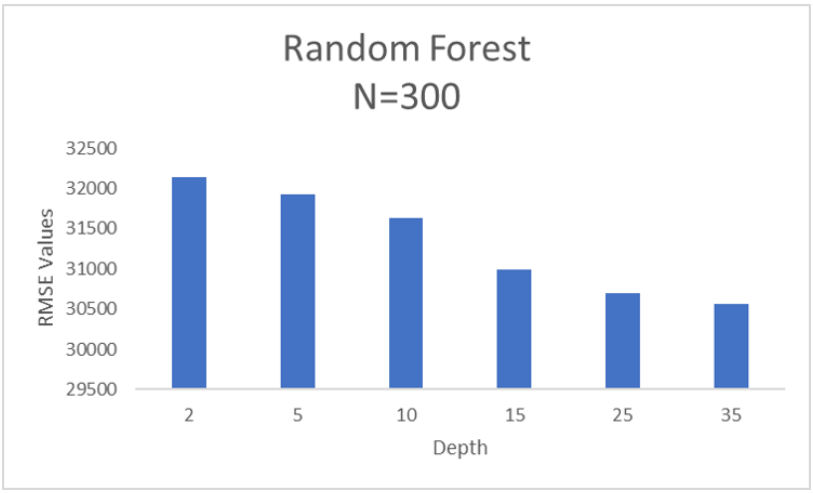

Fig. 11(f). RMSE values for different depths of Random Forest with 300 estimators

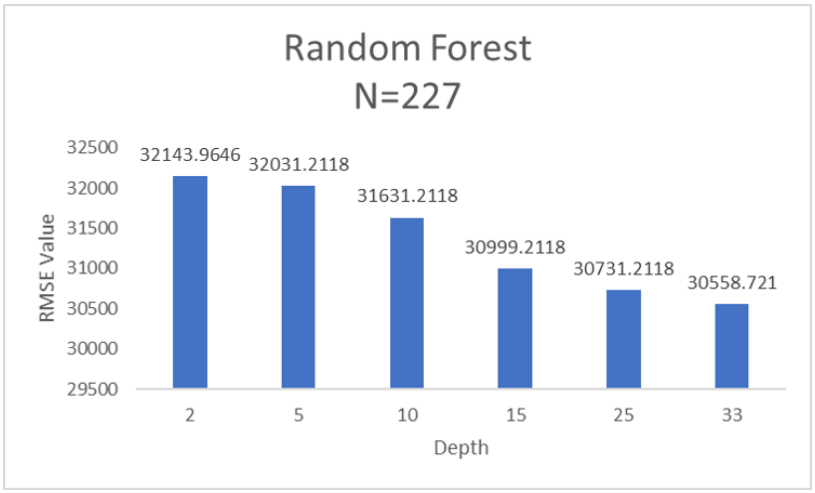

Fig. 11(g). RMSE values for different depths of Random Forest with 227 estimators

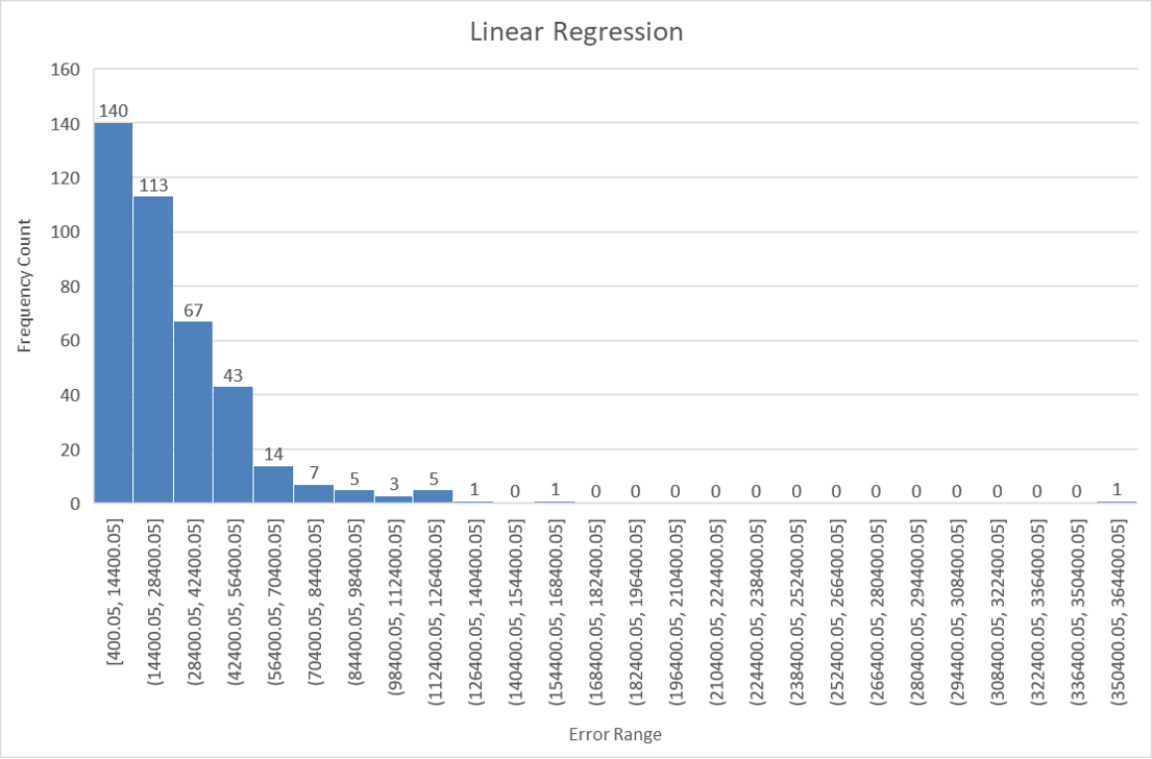

Fig. 12(a). Histogram of errors for linear regression

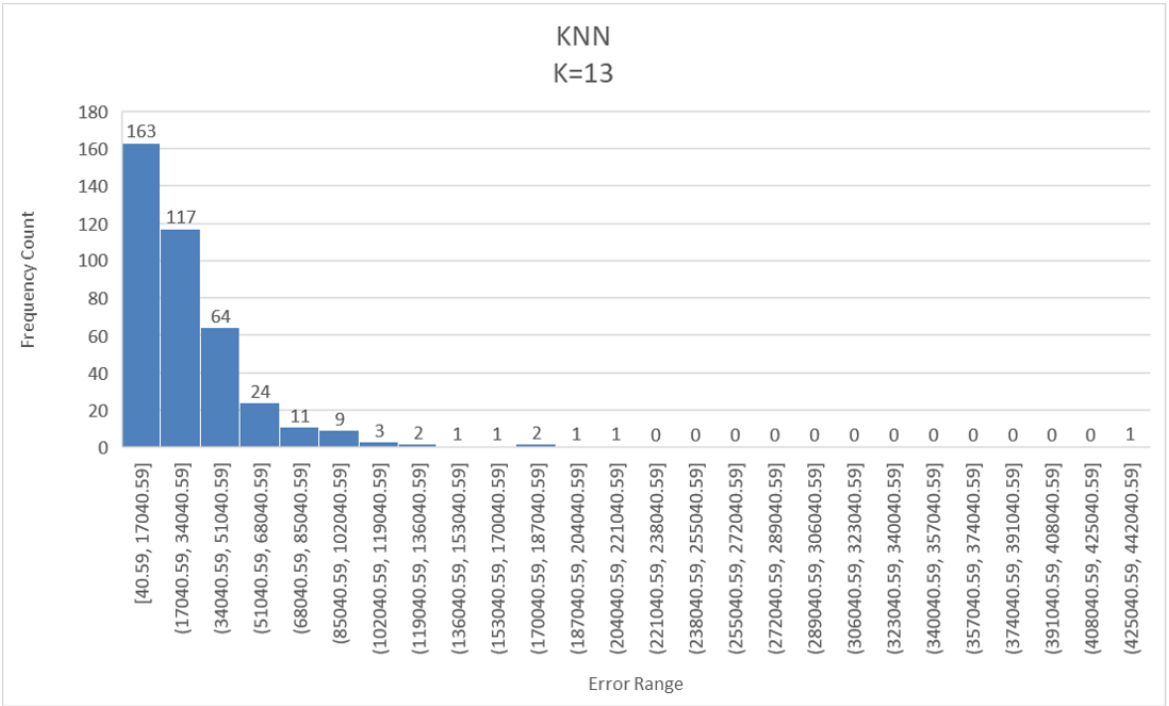

Fig. 12(b). Histogram of errors for KNN

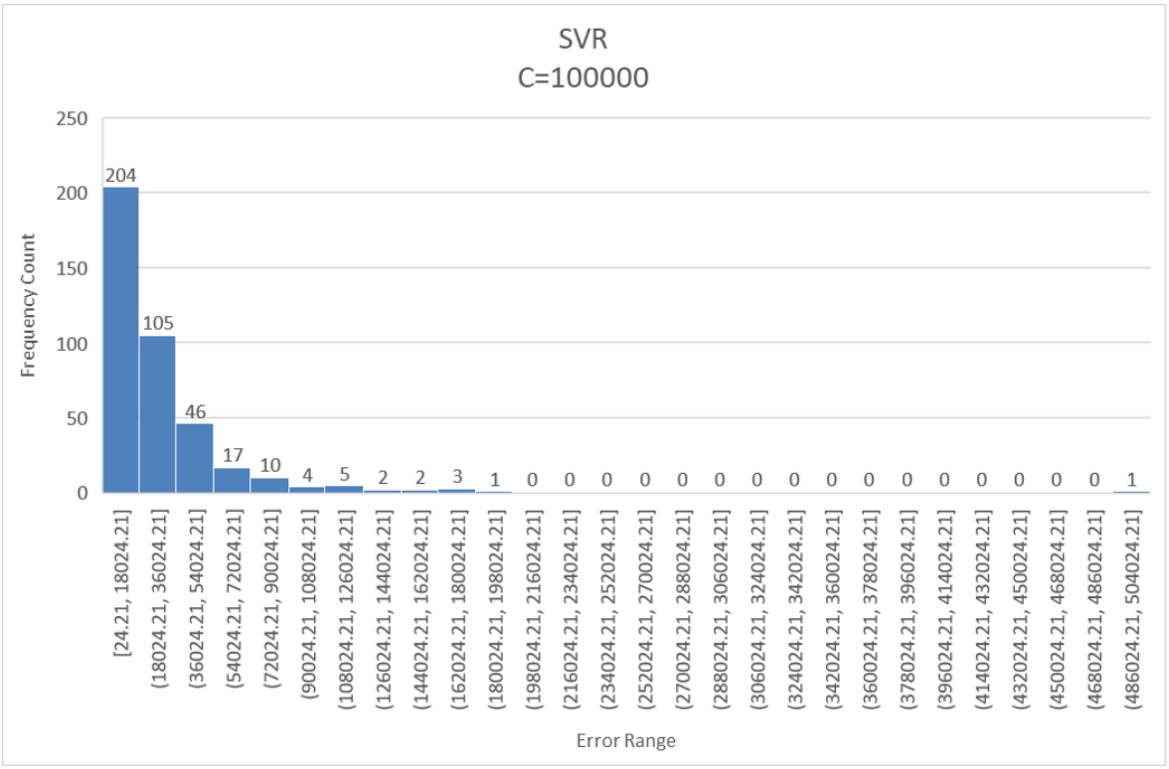

Fig. 12(c). Histogram of errors for SVR

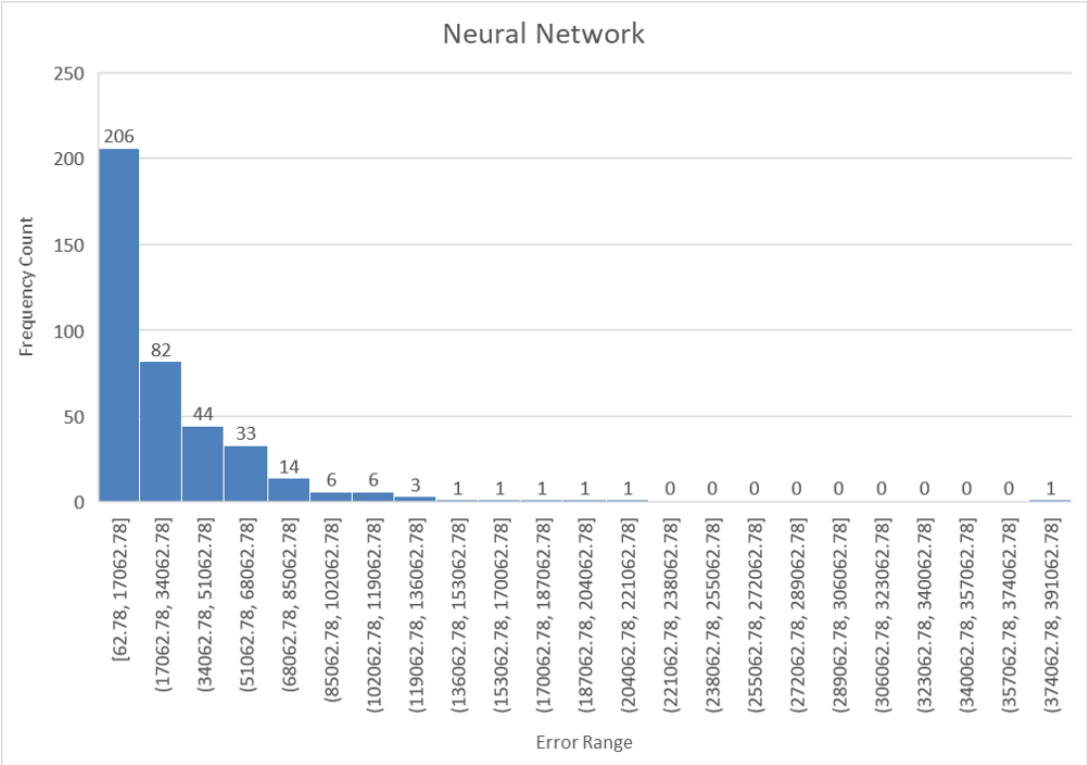

Fig. 12(d). Histogram of errors for NN

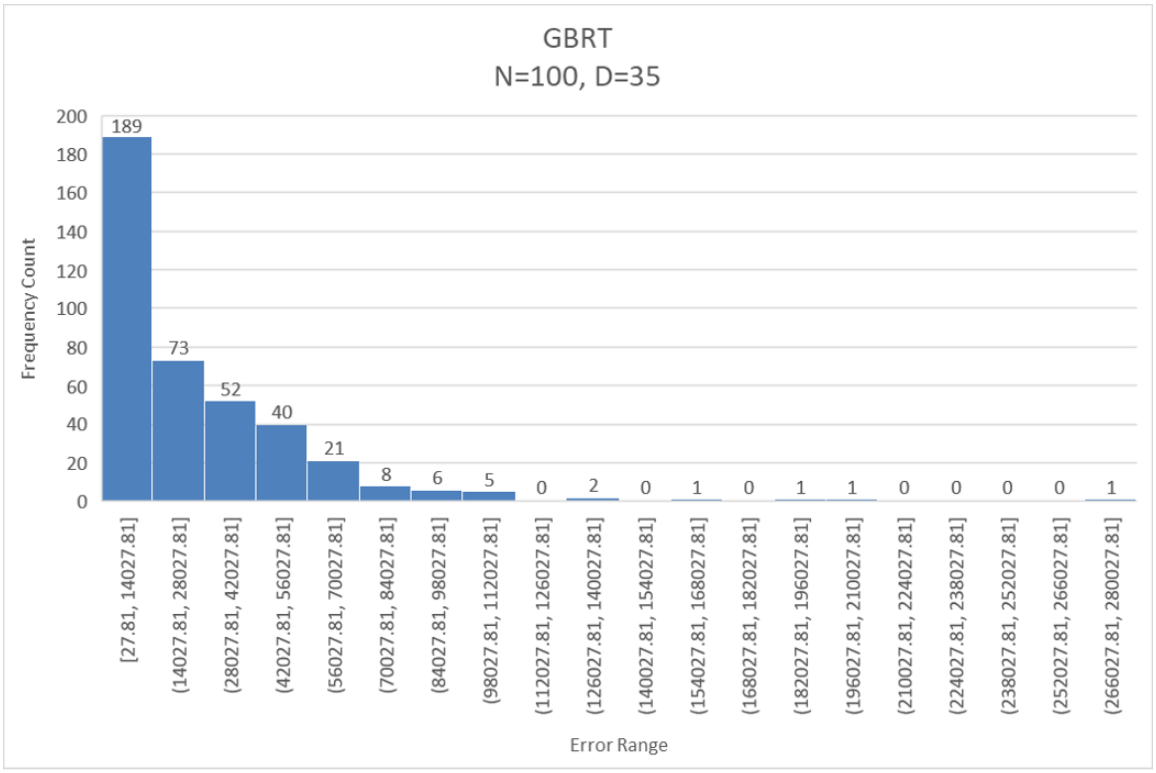

Fig. 12(e). Histogram of errors for GBRT

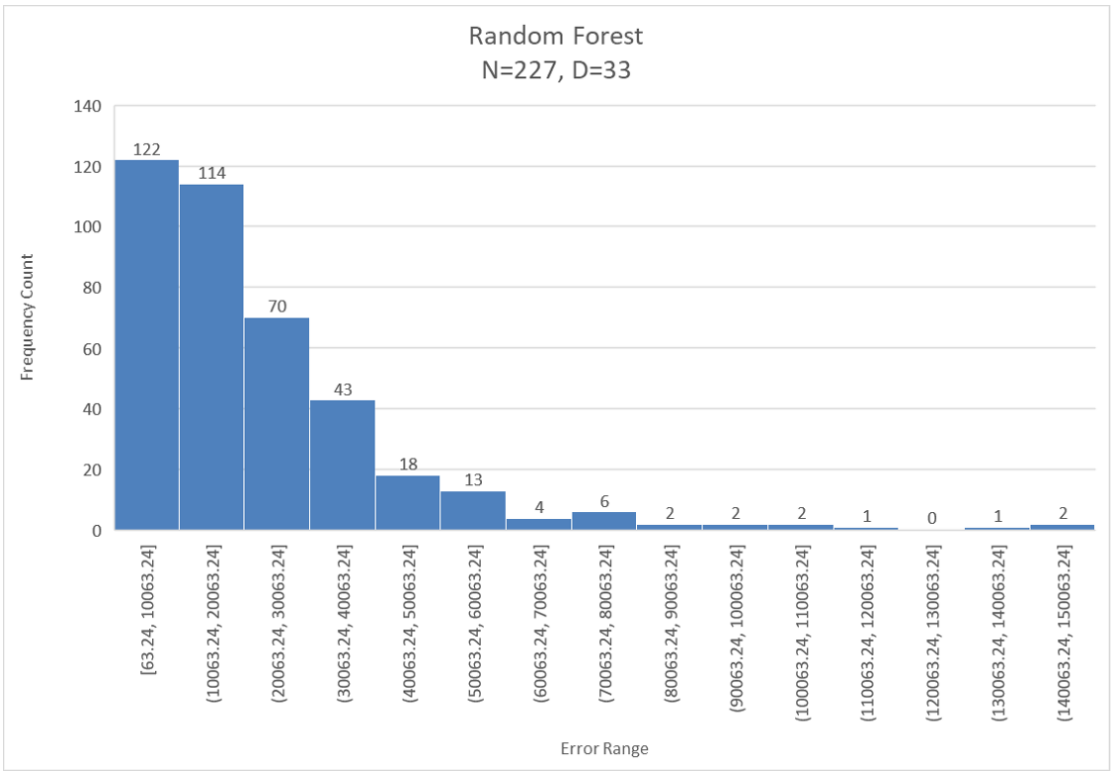

Fig. 12(f). Histogram of errors for Random Forest

Model	Error (RMSE)
Linear Regression	41163.23764482
k-Nearest Neighbour	47765.54137982
Gradient Boosted Regression Trees	44671.54111611
Random Forest	**30558.721**
Support Vector Regression	52250.7601211
Neural Network	41760.2730680

Table 1. Performance comparison of the used models