Predicting House Prices in Iowa using Advanced Regression Techniques

Nick Biso
Hong Li
Srishti Shetty
Yangsheng Lin
Rohan Das

Abstract

Predict housing prices in Ames, Iowa. Compare and contrast the different machines learning algorithms. Explain their philosophies and methodologies. Test their strengths and weaknesses. Possibly ensemble the models to produce a more accurate prediction.

Introduction

There are 80 independent variables and a dependent variable which is the sale price. 1500 observations were listed in this dataset.

Data Exploration

This is where we put our insights. Lets keep it short because this is not the focus of our presentation.

Feature Engineering:

Feature Creation:
We added the following macroeconomic variables: GDP-US, GDP-Iowa, GDP-Iowa Private, GDP-Real Estate, Unemployment Rate, CPI, Wilkshire 5000 Return, Wilkshire 5000 Index Values, Dow Close, Dow Volume. We also the different components of the time series like the trend, seasonal and cyclical. We also added the normalized version of the variables that are not normally distributed.

Then, we turned categorical variables into dummy variables. Now we have 180 variables.

Feature Selection:
With such sparse matrix, we want to remove the redundant variables and thus we can interpret the result easily and get useful insight. Besides that, there are some other benefits we can get from feature selection:

- shorter training times
- to avoid the curse of dimensionality
- enhanced generalization by reducing overfitting

One method is to computing mean absolute correlation and remove variables which have correlation higher than cut-off(default cutoff = 0.75). By doing this, we successfully reduce 30 variables from 180.

The importance of features can be estimated from data by building a model. For example, we could use decision trees and find out the variables importance ranking. For other algorithms, we could use a ROC curve to estimate importance ranking.

Modeling

I.Regression (benchmark)

II. Instance Based
-Kth Nearest Neighbor (benchmark)
-Support Vector Regression

III. Neural Network

IV. Tree Based Ensemble
-Random Forest
-Gradient Boosting
-XGboost

OLS Regression

Ordinary least squares regression assumes that it's independent variables are a linear combinations of it's dependent variables. It is a 200 year old algorithm however it is also the most well researched. OLS regression is best known for its interpretability. This model was able to predict with an MAPE of less than 1%.

K-nearest neighbors

This algorithm is one of the most simple and versatile algorithms ever created. It simply finds the *k* nearest neighbors using a defined distance metric. K is a parameter that the user has to define and can drastically increase or decrease this algorithm's accuracy. It then averages the dependent variables of these neighbors(for regression) or gets the majority vote (for classification) to predict the dependent variable. This model was able to predict with a MAPE of less than 1%.

Support Vector Regression

Like random forest, support vector machine, as one of the most popular classification models, also could be use in predicting housing price. The basic idea of support vector regression is somewhat similar to support vector machine but with different constraints. Here is the model:

$$\text{minimize } \frac{1}{2}\|w\|^2$$
$$\text{subject to } \begin{cases} y_i - \langle w, x_i \rangle - b \le \varepsilon \\ \langle w, x_i \rangle + b - y_i \le \varepsilon \end{cases}$$

Where xi is the vector of variables of observation i and yi is the price of observation.

In this model, we can transfer the categorical variables into dummy variables and reduce the total number of variables by 30.

Artificial Neural Networks (ANN)

An Artificial Neural Network (ANN) is a computing system that is modelled to mimic how the human processes information. Similar to how neurons in the brain are structured, ANN’s are composed of a large number of highly interconnected processing units. Neural networks are organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'.

ANNs, like people, learn by example. An ANN is designed for a specific function and then trained by processing a set of known inputs and outputs. Similar to how a child is taught to identify an object (i.e. animals, fruits, cars, etc) by being shown a variety of examples of a given object, the ANN “learns” how to classify or predict based on the provided input. **(you can remove if necessary)**

Although most commonly used as a classifier, ANN’s may be used as a regressor. This is the case in predicting housing prices for the Ames, Iowa dataset. For the ANN to be trained the dataset need to be preprocessed, the following data preparation steps were implemented:

Replace all missing values →Discard low variance predictors→Remove outliers→,Keep only numeric and categorical variables → Standardized continuous variables →Convert all categorical variables using ‘one hot’ encoding.**(you can remove if necessary)**

Once the data is ready to be fit we fit our model using the ‘neuralnet’ package in R. Due to the high computational cost of training ANNs, the ‘doParallel’ package in R was used. Tuning can be a rather complicated process for ANN’s but in this case was limited to the number of hidden nodes. In this case the lowest MAPE and log(RMSE) was found using 3 hidden layers and were 14.09% and .1857 respectively.

Limitation to applying neural networks on this housing set may include but not limited to the fact that (i)ANN’s typical work better as a classifier. (ii) This is a time series regression problem and the model should be more finely tuned. (iii) ANN’s proneness to overfit.

Interestingly, applying PCA to variables before fitting the neural network has often lead to better results, something to perhaps look into while further fine tuning the model.

CART Ensembling

Decision/Regression Tree is a predictive model that asks a series of questions that can be answered with a yes or a no about the data and gives a prediction given a sequence of responses. It's greatest strength is in it’s very interpretable nature. However, it has a reputation of overfitting.

Bootstrapping is the taking of a sample of size N with replacement from a population N. This technique allows estimation of the sampling distribution of almost any statistic using random sampling methods. As an example, population N = {1,2,3,4,5} has a bootstrap sample of {3,2,2,2,4}. Given that a population is sufficiently large, it has an almost Zero probability of having duplicate bootstrap sample sets. The bootstrap was published by Bradley Efron in "Bootstrap methods: another look at the jackknife" (1979).

Bagging is short for bootstrap aggregation. This is the process of training a model for each bootstrap sample that is available. The result of each model is then later averaged (for prediction) or majority voting (for classification). It is beneficial for unstable procedures including neural networks, classification/regression trees etc. but degrades the performance of stable algorithms like k-nearest neighbors. For Random Forest we will be using classification/regression trees.

Boosting stems from Bagging. It makes bagging even more accurate, reduces its bias and variance at the expense of overfitting. How it does this is that it adds weights to each of the bagged prediction. It determines the weight that it gives to each tree depending on the trees accuracy. Every time a new tree is added, it recomputes the weight that is given to each specific tree.

Random Forest

Random Forest is an ensemble learning method that can be used for both classification of regression. It is composed of Bagging and Random Subspace Method.

One of the issues of bagging is the correlation of our trees. The first few splits will be at the same variables resulting in similar results. This is caused by having the same set of variables. The average of similar predictions will output the same or even worse predictions. This is where random subspace comes in. Instead of just taking random observations from our population with replacement (bagging), it will also subset the variables. This will result in dissimilar trees yet weaker that is once averaged, will produce significantly better predictions.

Random Forest has the following parameters:
Number of variables per sample - A forest that has a low number of variables per split is less correlated because the rank of the split the variable will always be similar.
Number of Trees - Intuitively, the more trees you have the better but it does have diminishing returns in accuracy. Furthermore, a significant amount of trees can be computationally expensive.
Tree Depth - Tuning this parameter is the same as tuning the number of trees.

Grid Search can be used to find the optimal values for these parameters although it can be very computationally expensive.

Strengths:
-It is fairly easy to implement
-Automatically balances the dataset
-Out of the box accurate

Weaknesses:
-Can be computationally expensive
-It can be hard to explain to stakeholders

Our final model used 50 variables with 500 trees and mtry at 10. Our final MAPE is at 1%.

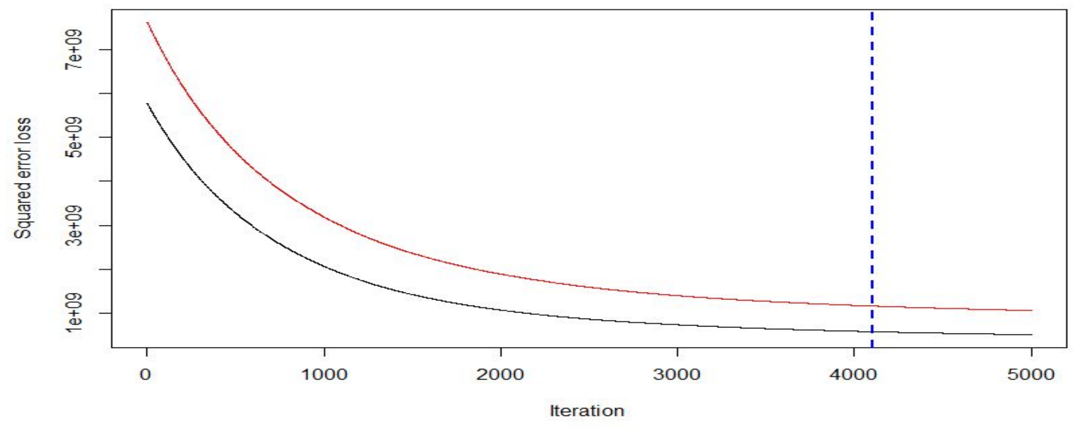
Gradient Boosting Machine

Gradient Boosting Machine(GBM) is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models. So, each new tree is a master of the errors of its predecessors.

While using GBM it is very essential for one to select optimum number of trees, number of interaction variables required for the trees and the learning rate of the model. A very small number of trees can cause low variance to be captured in our data and hence the least squared errors for the test can be high. However, a very large number of trees can also be lethal since it can cause overfitting in the model which can lead to errors.

To tackle the parameter optimization in our problem, we used the *expand.grid* function in R that looked for various combinations of number of trees, number of interaction variables and learning rate of the model (shrinkage).

By doing parameter tuning, we got 4102 trees, 3 variables that interact with each other and a shrinkage rate of 0.001. Using these parameters in our data we got a MAPE of 12.6% for the test set and a RMSE of 0.042.



The red line in the above diagram shows the test error and the black line shows the train error. The blue line indicates the optimum number of trees.

XGboost

XGboost is short for extreme gradient boosting. It is an implementation of the gradient boosting framework, which is a library designed and optimized for boosted tree algorithms. The main purpose of XGboost algorithms is to push the extreme of the computation limits of machines to provide a scalable, portable and accurate for large scale tree boosting. With the high accuracy of prediction, XGboost has gained popularity by winning numerous machine learning competitions more recently.

The XGBoost library implements the gradient boosting decision tree algorithm, it used ensemble technique where new models are added to correct the errors made from existing models until no further improvements can be made. New model is created to predict the residuals of prior model. Then, XGboost combines these weak models into a strong model to have the final prediction, which is highly accurate compared to simple model.

1. Objective Function

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$
$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

- Training Loss (Predictive Power): Root Mean Square loss, Logistic loss, etc.
- Regularization (Simplicity and Overfitting): depend on the number of leaves and score that assigned to each leaf.
- Balance between Training loss and regularization is important because we want a high predict accuracy model without overfitting.

2. Boosted Tree

   I. Regression Tree(CART)

   · Decision rules same as in decision tree

   · Contains one score in each leaf value

   II. Tree Ensemble: A set of weak learners

   III. Additive training

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\cdots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

Start from zero and keep adding new function which minimize the objective function.

$$Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T$$

And using Taylor expansion to get final objective function.

IV. Greedy Learning of the tree

Starting from tree with depth 0, add a split for each leaf node of the tree and calculate the Gain.

$$Gain = \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda} - \gamma$$

Then choose the one with largest Gain value.

After running the XGboost model in R, we got the a table results using 1460 samples with 79 features, and the cross-validation is 7 fold, repeated 3 times.

eta	max_depth	colsample_bytree	subsample	nrounds	MAPE
0.3	1	0.6	0.5	50	0.12773963
0.3	1	0.8	0.75	50	0.12250411
0.3	2	0.6	0.75	150	0.10062907
0.3	2	0.6	1	50	0.10644554
0.3	2	0.6	1	100	0.10059691
…	…	…	…	…	…
0.3	2	0.8	0.50	50	0.11197314
0.3	2	0.8	0.50	100	0.10611256
0.3	2	0.8	0.50	150	0.10379730
0.3	2	0.8	0.75	50	0.10793367
0.3	2	0.6	1	150	0.09786556

RMSE was used to select the optimal model using the smallest value. The final values used for the model were:

eta	**max_depth**	**colsample_bytree**	**subsample**	**nrounds**	**MAPE**

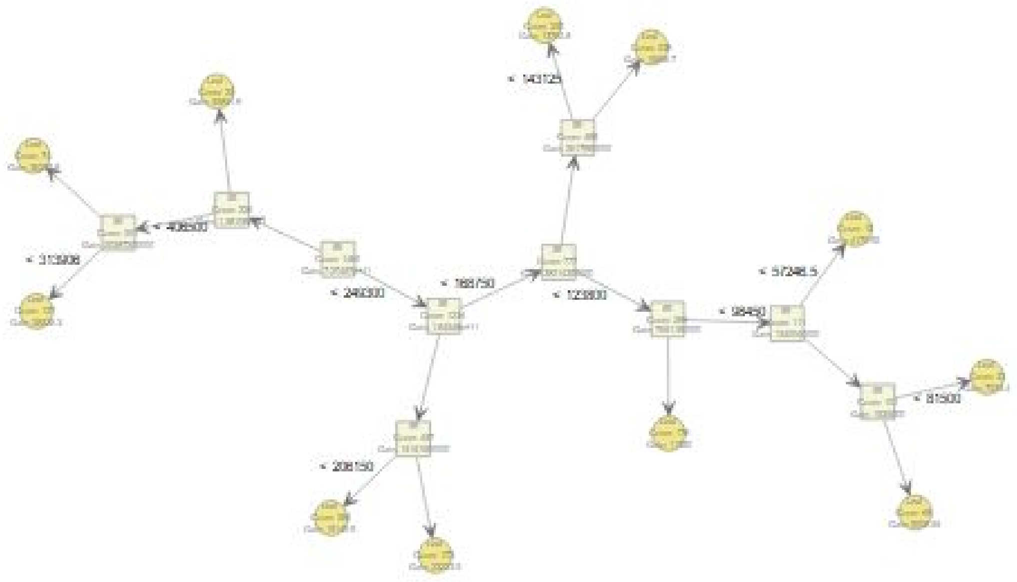
0.3	**2**	**0.6**	**1**	**150**	**0.09786556**

Which gave us an accuracy of 0.90213444, approximately 90.21%.

And Most Important, we found the feature importance set based on importance matrix and top three features below :

- Square feet of House
- Neighborhood
- Overall quality of the House

In addition, we plotted the first regression trees that made by the Xgboost which are showed below. Then, combining all 150 base learning trees as a final “learner”, we would have the final predicted value.



Conclusion

After trying various algorithms to predict the house prices in Iowa, we have concluded the following,

- Since our models are highly correlated with each other, using ensemble of models will give nearly the same accuracy as the individual models.
- Across all the models, variables like Neighbourhood, Total square feet of the house and the Overall Quality was important.
- Tree based models work better for predicting the house sale prices.

Individual Contribution

Sr no	**Group Member**	**Contribution**
1	Nick Biso	Group Leader, Macroeconomic Variables Data Collection, OLS Regression, K Nearest Neighbours and Random Forest,, Presentation and Paper Writing
2	Hong Li	Support Vector Regression modelling, Ensemble Model, Presentation and Paper Writing
3	Srishti Shetty	Gradient Boosting Machine modelling, Ensemble Model, Presentation and Paper Writing
4	Yangsheng Lin	XGboost modelling, Presentation and Paper Writing
5	Rohan Das	Artificial Neural Network modelling, Presentation and Paper Writing

**Paper writing division
50% : Nick Biso & Srishti Shetty
50%: Hong Li, Yangsheng Lin & Rohan Das