Assignment 1 Supervise Learning

Wenjun Jiang GTID: 903369916

Executive Summary

The purpose of this project is to explore some supervised learning methods we have learned in class and compare their performance under different circumstances. Those supervised learning methods include: (1) Decision trees (with / without pruning); (2) Neural networks; (3) Boosting; (4) Support vector machines (SVM); and (5) K-nearest neighbors (KNN).

Programming language: R

Related R packages: C50 (to build decision tree), Caret (for cross-validation, and parameter tuning), corrplot (to plot correlation), and plot3D (for 3D scatter plot).

1. Selection of Data Sets

   Two data sets from UCI machine learning repository are selected in order to test the 5 supervised learning techniques as mentioned above.

   (1) Housing

   - It includes housing values in Boston suburbs, with 14 attributes (13 continuous attributes, 1 binary attribute named CHAS, which is Charles River dummy variable).
   - Download address: http://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data
   - Problem researched:
     Based on median value of owner-occupied homes (MEDV), I classified MEDV values into 5 levels. Then, a multiple classification model is built to estimate MEDV levels based on given features.

   (2) Wholesale

   - It includes annual spending in monetary units on different product categories, with 8 attributes (6 continuous attributes, 2 categorical attributes).
   - Download address: https://archive.ics.uci.edu/ml/machine-learning-databases/00292/Wholesale%20customers%20data.csv
   - Problem researched:
     Based on Channel attribute, I consider it as a binary classification problem, which trains model and predicts customers' channel based on given features.

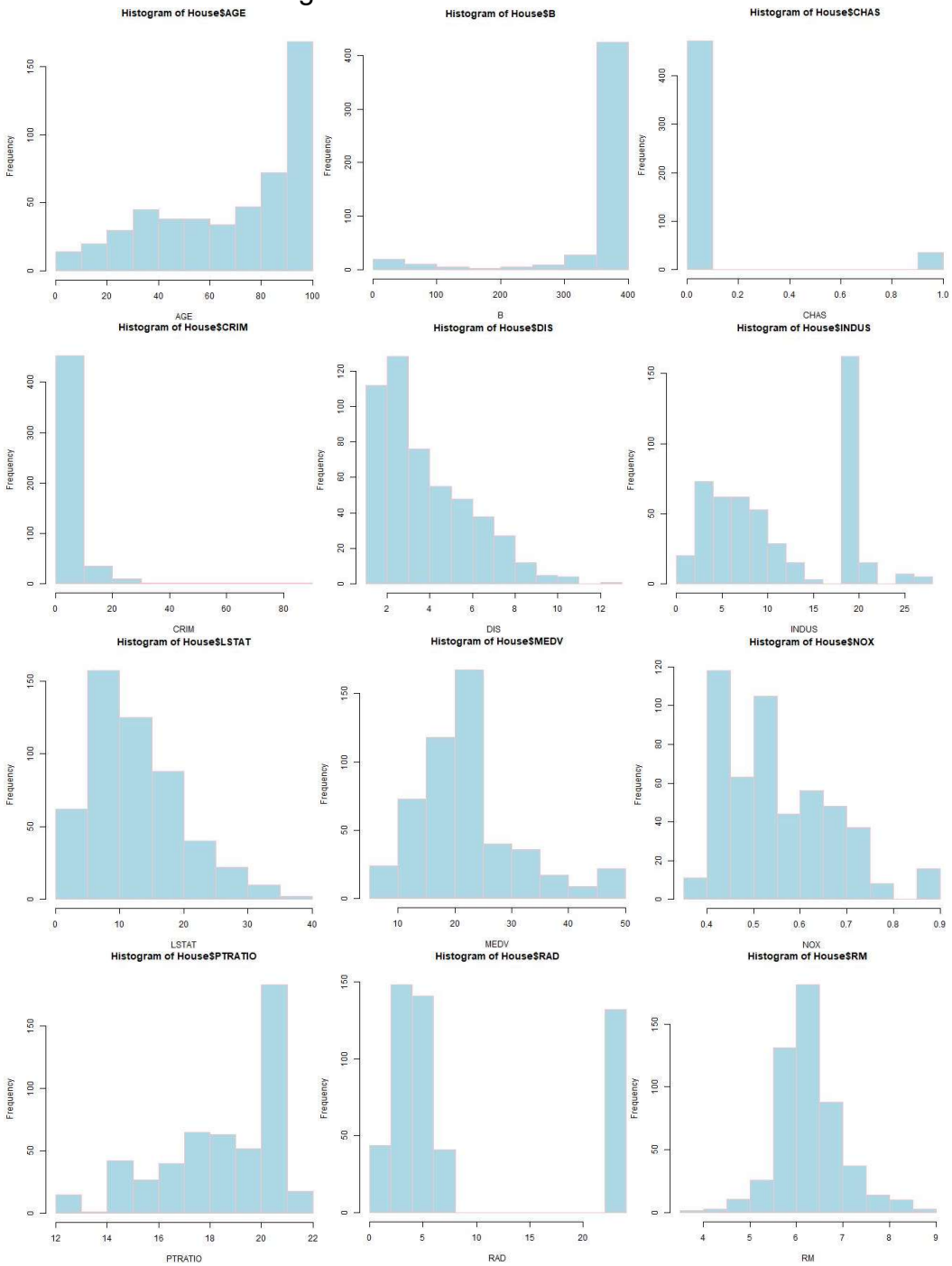
2. Data Exploration

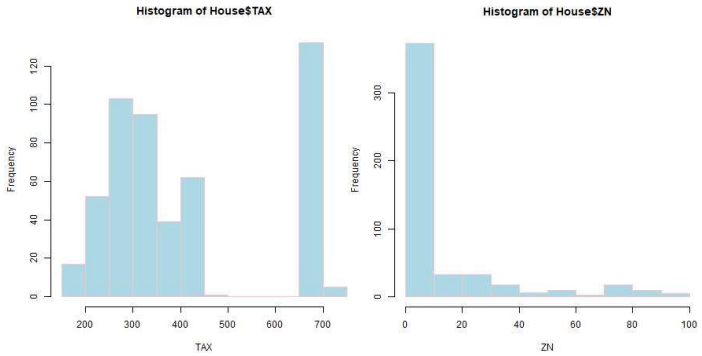
   (1) Histogram

Both datasets are explored and histograms are drawn for each attribute.

1) Housing

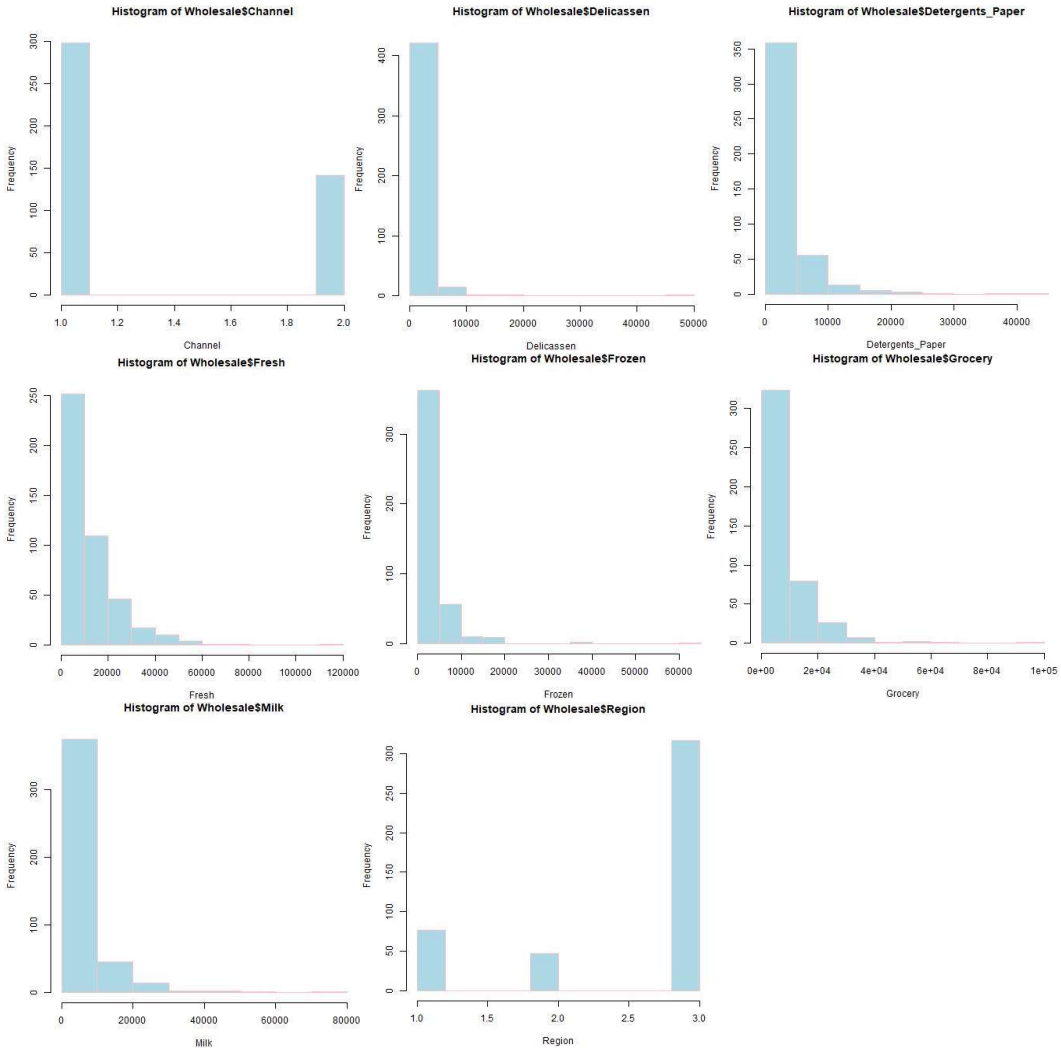
   As shown in figures below, CHAS is the only binary-valued attribute, CRIM, ZN, and B have very concentrated distribution, while the rest are continuous attributes with wide ranges of distribution.





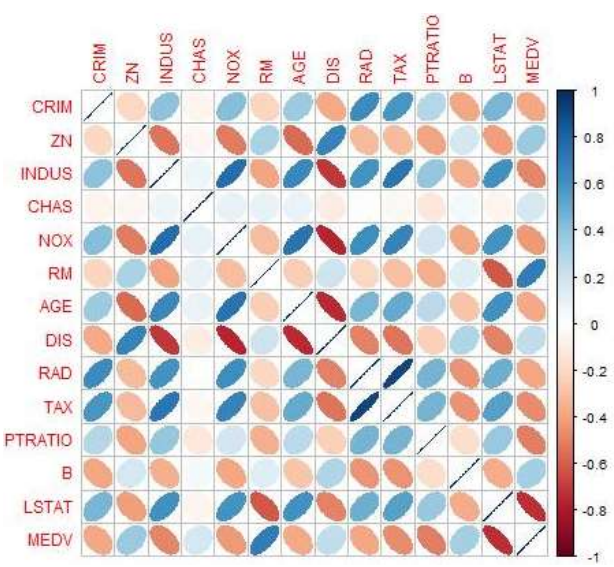
2) Wholesale

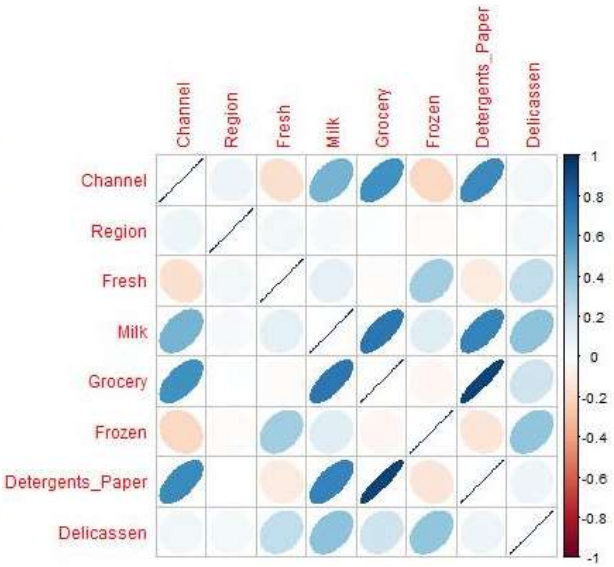
As shown in figures below, Channel is a binary-valued attribute, Region is a tripe-valued attribute, while the rest are all continuous attributes with quite concentrated distribution.



(2) Correlation Plot

Correlation between each pair of attributes are plotted.





1) Housing
   As shown in the plot on left, 5 pairs have strong positive correlation: (NOX, INDUS), (INDUS, TAX), (NOX, AGE), (MEDV, RM), (RAD, TAX), 4 pairs have strong negative correlation: (IDUS, DIS), (NOX, DIS), (AGE, DIS), (MEDV, LSTAT).

2) Wholesale
   As shown in the plot on right, 2 pairs have strong positive correlation: (Milk, Grocery), (Detergents_paper, Grocery).

3. Data Preparation

(1) Housing
For the MEDV attribute in housing dataset, 5 levels are separated based on MEDV values: (<10, 1), (10~20, 2), (20~30, 3), (30~40, 4), (>40, 5).

(2) Training / Testing Data Partition
Random sampling method is used to separate both datasets into training data (80% of all dataset) and testing data (the rest 20% of dataset). Random seed number is set to 2018.

4. Method 1: Decision Trees

R's C50 package is used to build decision trees. 2 methods, without pruning and with pruning, are used. Two parameters are used for pruning purpose (confidence factor, minimum cases).
R's caret package is used for cross-validation purpose with 10 iterations to calculate accuracy and kappa values.

(1) Decision Trees without Pruning

As shown in the table below, for both training and testing results, wholesale dataset returns better accuracy compared to housing dataset. This is because that wholesale model, which uses a binary-valued Channel attribute, is a simple binary classification problem. By contrast, housing model is a multiple classification problem, thus it is harder to train and predict. Another possibility is the overfitting occurred in the housing model, as there are in total only 506 observations with 14 attributes, while for wholesale model, 440 observations are there with only 8 attributes are included.

Model	Training		Testing	
	Accuracy	Kappa	Accuracy	Kappa
Housing	0.8968	0.8460	0.7475	0.6203
Wholesale	0.9490	0.8839	0.9195	0.8139

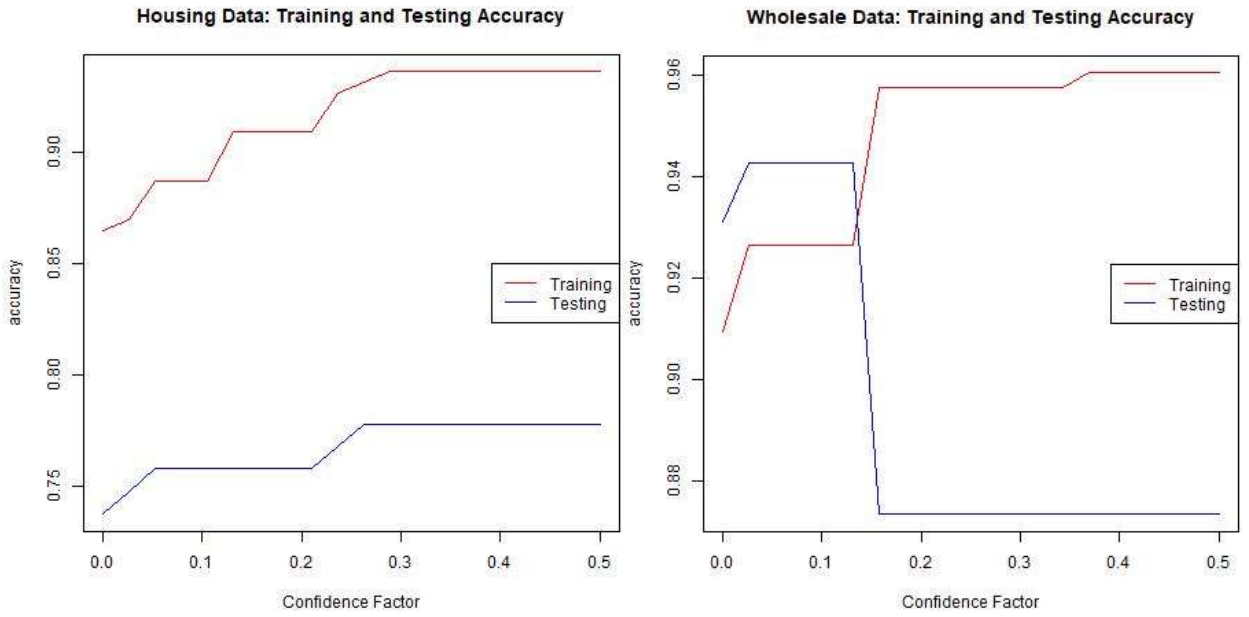
(2) Decision Trees with Pruning

1) Pruning with Confidence Factor

Confidence factor (CF) ranges from 0 to 0.5 is used to prune the decision tree for both datasets. Accuracy vs. CF is plotted as shown below.

From both plots, it is clear that when CF is larger than 0.25, accuracy in testing data doesn't improve anymore.

For housing model, changing CF between 0 to 0.25 doesn't change the accuracy of both training and testing data very much. By contrast, wholesale model is very sensitive to CF level from 0 to 0.25. This is probably because more data are pruned when CF increases, thus reducing the accuracy in testing dataset.



Thus, for prediction purpose, CF=0.25 is used. Accuracy and Kappa values for both training and testing datasets are calculated as shown in the table below. It is clear that with pruning on confidence factor, accuracy for both

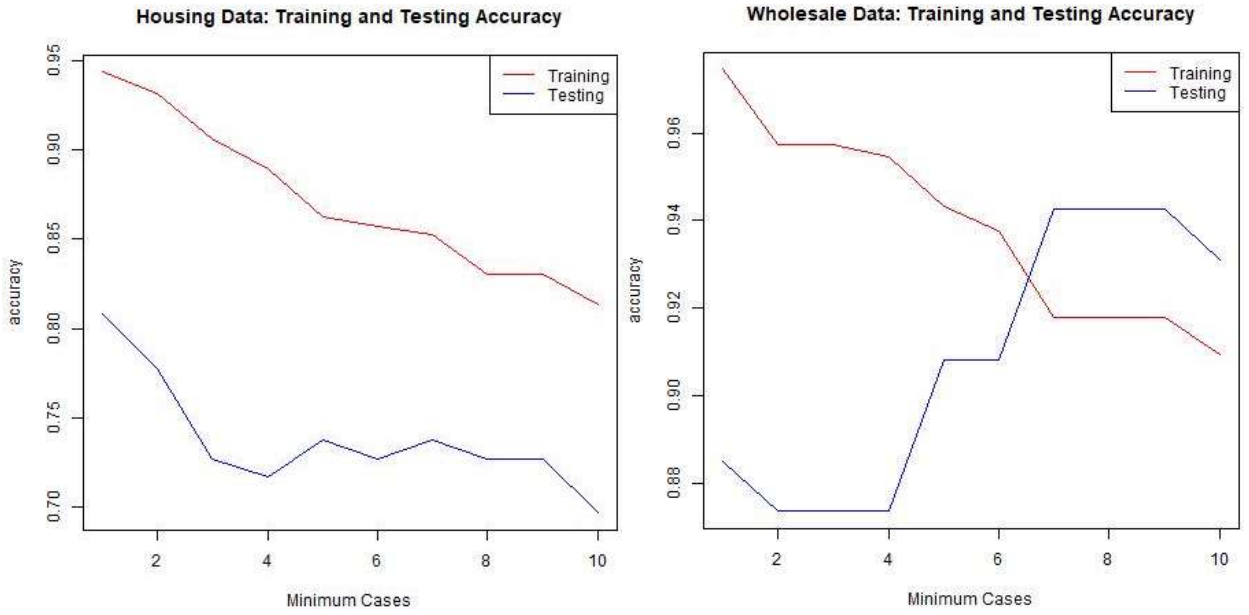
training and testing datasets improves for housing model, but only testing dataset improves for wholesale model.

Model	Training		Testing	
	Accuracy	Kappa	Accuracy	Kappa
Housing	0.9312	0.8946	0.7778	0.6575
Wholesale	0.9263	0.8361	0.9425	0.8719

2) Pruning with Minimum Cases

Minimum cases (MC) ranging from 2 to 11 are also used to prune the decision tree for both datasets. Accuracy vs. CF is plotted as shown below. For housing model, increasing MC results worse accuracy values. Thus, the best MC value is 2.

For wholesale model, increasing MC results worse accuracy in training data but better in testing data. Thus, the best MC value of 7 is used for future calculation.



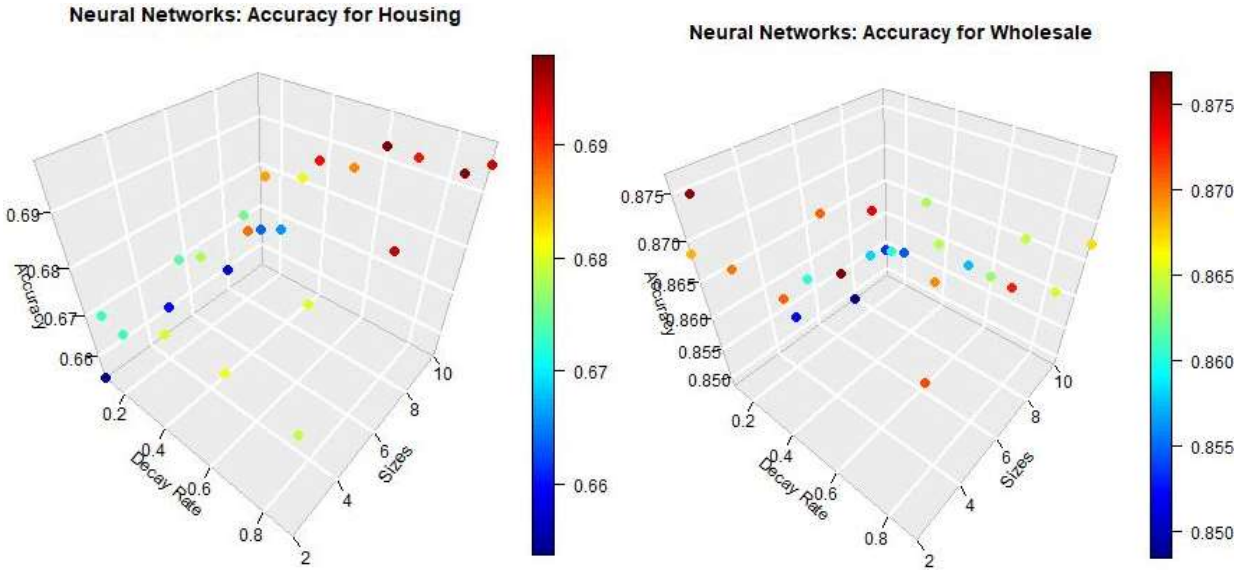
For prediction purpose, (housing, MC=2) and (wholesale, MC=7) is used. Accuracy and Kappa values for both training and testing datasets are calculated as shown in the table below. It is clear that with pruning on minimum cases, accuracy for both training and testing datasets improves.

Model	Training		Testing	
	Accuracy	Kappa	Accuracy	Kappa
Housing	0.9312	0.8946	0.7778	0.6575
Wholesale	0.9178	0.8117	0.9425	0.8696

5. Method 2: Neural Networks

Neural networks are built using R's caret package. Two parameters, decay rate, and hidden units, are tuned.

From the 3D plots as shown in the figures above, we can find the best combination of (decay rate, hidden units) to be (0.7, 8) for housing model and (0.7, 2) for wholesale model, which are used for prediction purpose.



As shown in the table below, for both housing and wholesale models, applying neural networks doesn't improve accuracy. Interestingly, however, accuracy of testing data is higher than that of training data for wholesale model. This is probably due to the limited amount of observations, which are not large enough to build neural networks model.

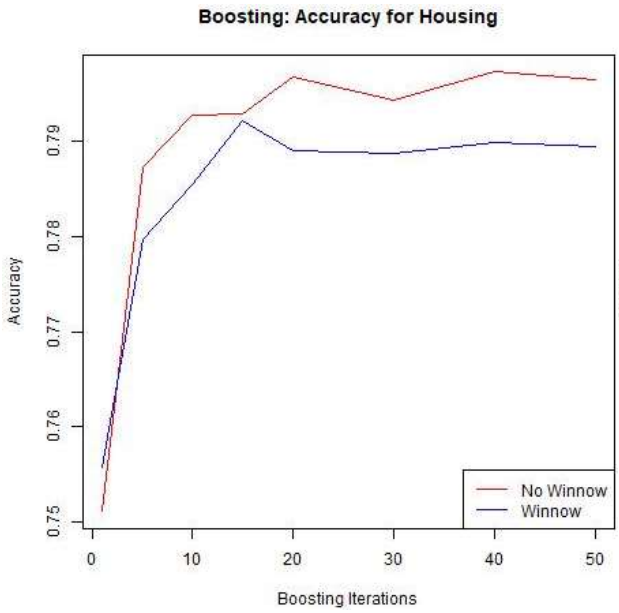
Model	Training		Testing	
	Accuracy	Kappa	Accuracy	Kappa
Housing	0.7396	0.6048	0.6970	0.5246
Wholesale	0.8952	0.7723	0.9425	0.8645

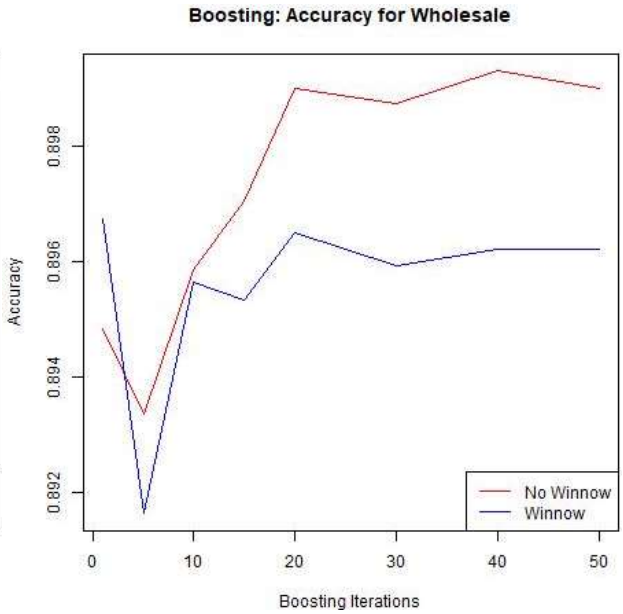
6. Method 3: Boosting

Boosting decision trees are built using R's caret and C50 packages. Two parameters, winnow and trials, are tuned.

Since winnow is a logic parameter, accuracy vs. iterations with winnow=True/False is plotted for both datasets.

As shown in the figures below, for both models, accuracy improves as boosting iterations increase, and winnow=False performs better than winnow=True.





Model	Training		Testing	
	Accuracy	Kappa	Accuracy	Kappa
Housing	0.9975	0.9962	0.7071	0.5319
Wholesale	0.9999	0.9999	0.9425	0.8645

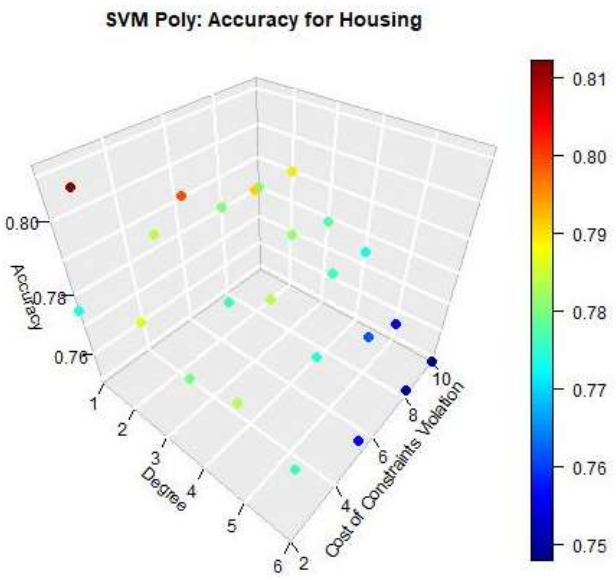
For prediction purpose, (winnow, iterations)=(False, 20) and (False, 40) are used for housing and wholesale models, respectively. It is clear from the table shown above that, for both models, training data result in very high accuracy. However, the accuracy in testing data is not good enough. It is mainly because of the overfitting problem due to the limited amount of observations.

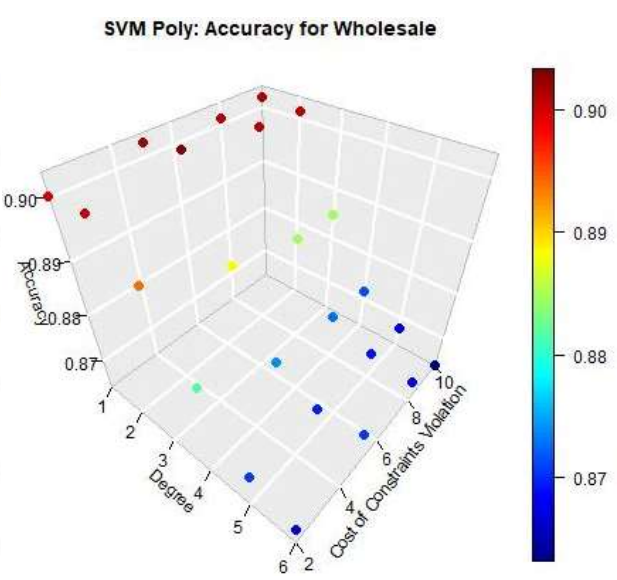
7. Method 4: Support Vector Machines

   Support vector machines (SVM) are built using R's caret package. Two different kernels, polynomial kernel and radial basis function kernel, are used to train SVM model.

   (1) SVM with Polynomial Kernel

   Two parameters, cost of constraints violation (C) and degree of polynomial, are tuned. As shown in both figures below, accuracy decreases as degree increases. This is mainly due to the overfitting problem in training dataset.





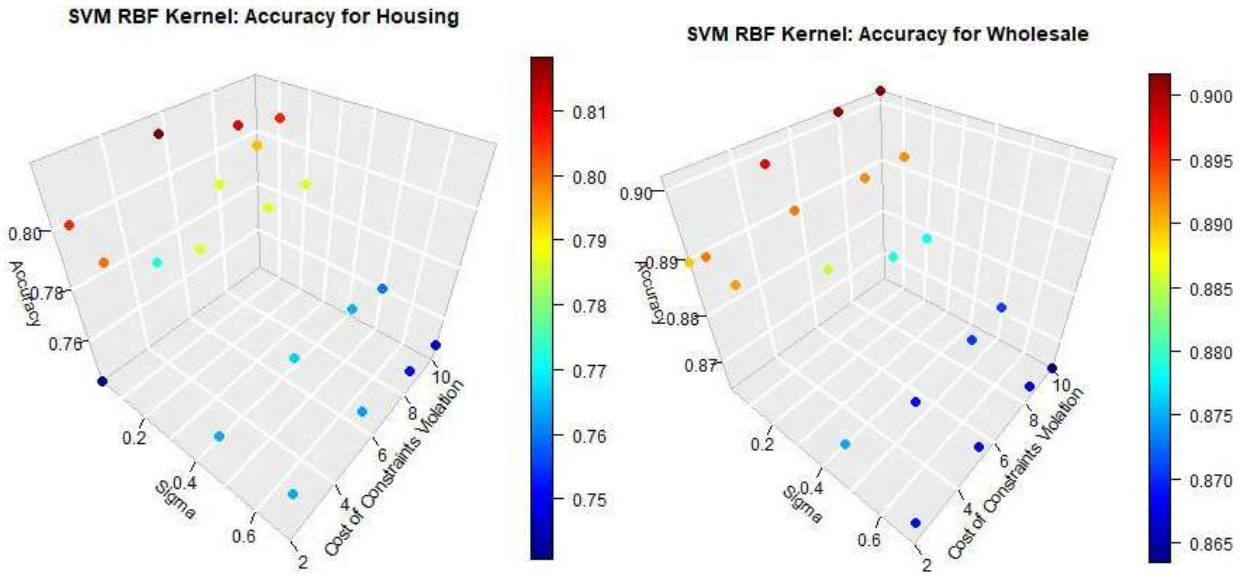
For prediction purpose, (C, degree)=(2, 2) and (5, 1) are used for housing and wholesale models, respectively.

Model	Training		Testing	
	Accuracy	Kappa	Accuracy	Kappa
Housing	0.8870	0.8253	0.7071	0.5380
Wholesale	0.9065	0.7877	0.9310	0.8358

As shown in the table above, accuracy for training data improves, while the accuracy is not good enough for testing data. Again, it is mainly due to the limited amount of observations which lead to overfitting.

(2) SVM with Radial Basis Function Kernel

Two parameters, cost of constraints violation (C) and inverse kernel width (sigma), are tuned. As shown in both figures below, accuracy first increases then decreases as sigma increases. This is mainly due to the overfitting problem in training dataset.

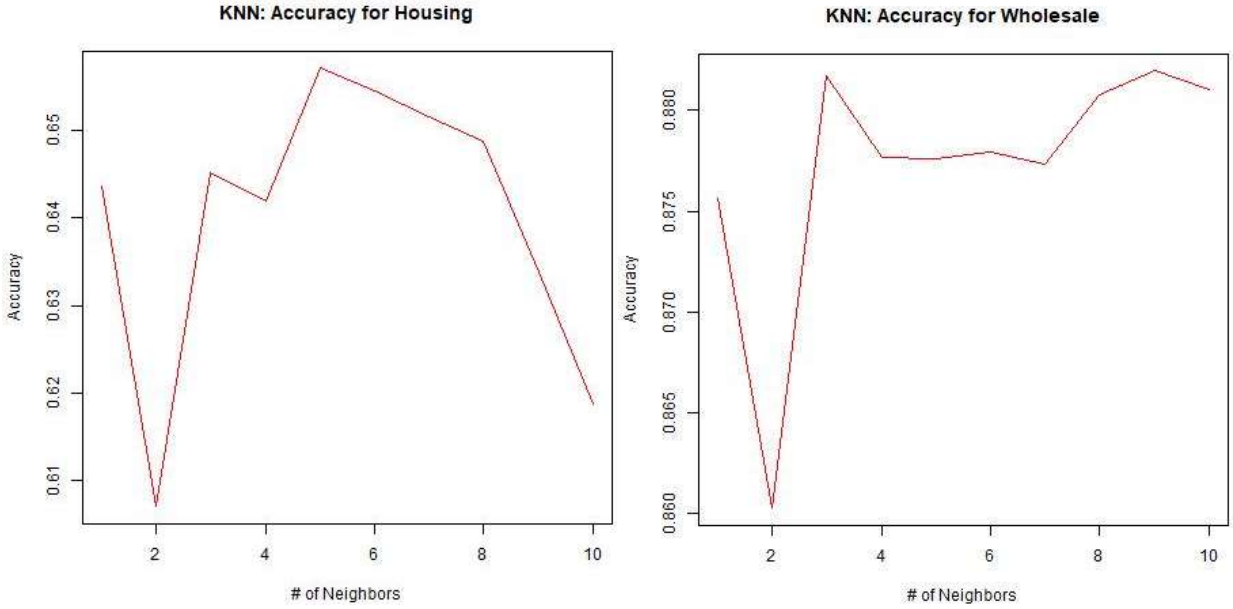


For prediction purpose, (C, sigma)=(5, 0.1) and (10, 0.01) are used for housing and wholesale models, respectively.

Model	Training		Testing	
	Accuracy	Kappa	Accuracy	Kappa
Housing	0.9115	0.8637	0.7374	0.5881
Wholesale	0.9207	0.8186	0.9310	0.8326

As shown in the table above, for both models, accuracy for training data improves, while the accuracy is not good enough for testing data. Again, it is mainly due to the limited amount of observations which leads to overfitting.

8. Method 5: K-Nearest Neighbors



For K-Nearest Neighbors (KNN), the only parameter to tune is the number of neighbors, K. As shown in figures above, accuracy reaches its best value when K=5 for housing model, and K=9 for wholesale model, which are used for prediction.

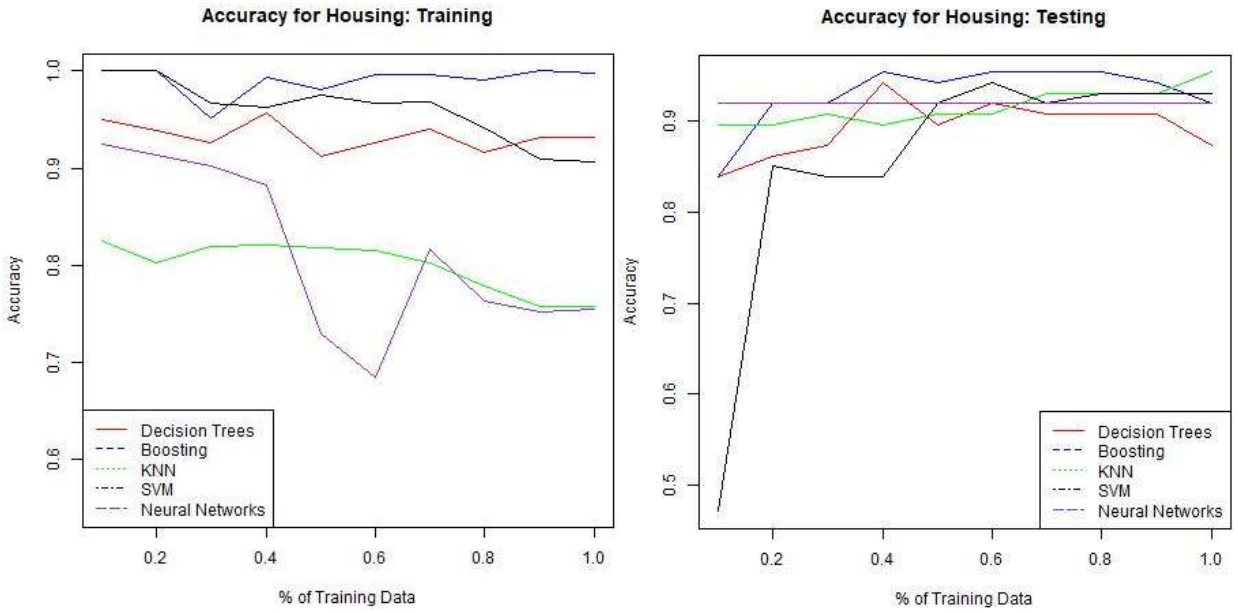
Model	Training		Testing	
	Accuracy	Kappa	Accuracy	Kappa
Housing	0.7666	0.6357	0.6364	0.3839
Wholesale	0.9348	0.8527	0.9540	0.8966

As listed in table above, accuracy for housing model is very low, while for wholesale model, accuracy is relatively good. The reason is that wholesale's Channel attribute is binary classification, while housing's Price_class is multiple classification. Thus, it is always easier to train binary than multiple classification.

9. Summary

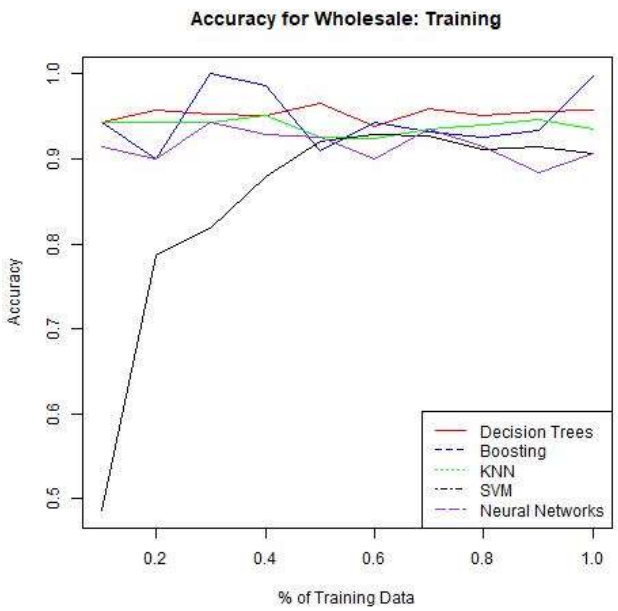
(1) Learning Curve

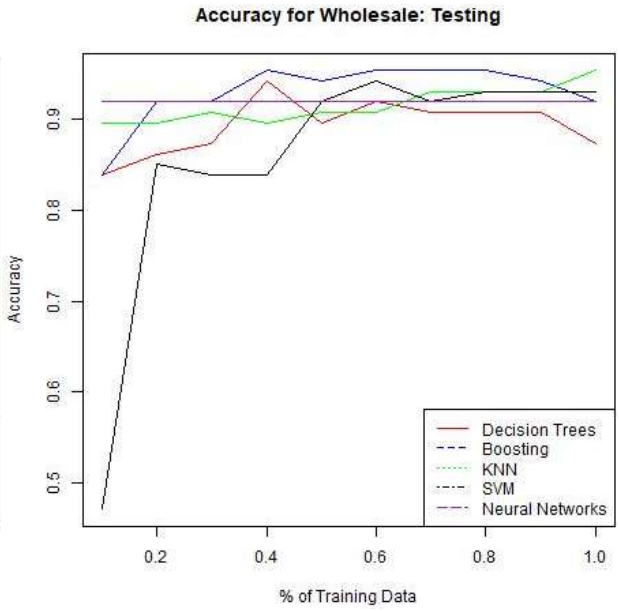
Accuracy vs. training size is plotted for both datasets, with all five supervised learning methods as discussed above. The results are shown below.



For housing model's training data, boosting achieves the best accuracy, followed by SVM and Decision Trees. KNN and Neural Networks return relatively bad accuracy. In general, accuracy decreases as the training size increases.

For housing model's testing data, Decision Trees and Boosting perform the best.





For wholesale model, both training data and testing data achieve stable and good accuracy for all methods except SVM. Among them, Boosting performs the best in both training and testing data.

(2) Comparison of Training Time

Average training time is recorded using R's sys.time() function. As shown in table below, decision trees runs the fastest, while SVM and KNN are significantly slower compared to other methods.

Decision Trees	Boosting	Neural Networks	SVM	KNN
0.8758	0.9487	1.9269	8.6402	6.8921