Machine Learning Project 2

Anthony Vasquez
Ph:313-920-8877
Email: avasque1@jh.edu

Editor: Anthony Vasquez

Abstract

This document presents findings for classification and regression methods run on several unique datasets. The k-nearest-neighbors (KNN) algorithm paired with different techniques are explored to learn the similarities and differences between the techniques. The techniques include edited and condensed KNN, k-neighbor finetuning, Euclidean distance calculation to approximate nearest neighbors, supplementation of gaussian kernel optimization for regression, and plurality testing for classification. The results show that not all of the

Keywords: Classification, Plurality test, Regression, Gaussian Kernel, Euclidean Distance, Condensed, Edited, Finetuning, Mean-Squared-Error, Accuracy

1 Introduction

Using a well-known method, like K-Nearest-Neighbors (KNN) to adequately train an algorithm is often not enough to develop a model that performs well. Whether it be finetuning or adding an additional discriminator, doing so often leads to better performance metrics. Four approaches to increasing performance including, fine-tuning k, fine-tuning sigma, condensed KNN, and edited KNN. It likely that these methods will increase the performance beyond the performance of the vanilla KNN described in later sections

2 Data

This section discusses the data, preprocessing, and split techniques used in this project.

2.1 Data Used

The datasets used for classification in these experiments include breast-cancer-wisconsin, car, and house-votes-84. For regression, abalone, forestfires, and machine datasets were used. See the appendix for more information on each of the datasets.

2.2 Data Preprocessing

All datasets underwent standardization by using the equation $z(x) = \frac{x-\mu}{\sigma}$, where x is the data value, μ is the mean, and σ is the attribute standard deviation. Categorical attributes were encoded by using non-binary encoded values prior to standardization.

2.3 Stratified K-fold Cross Validation

The data was separated into several stratified train and validation sets for k-fold cross validation, where k = 5. The full data was stored as a pandas dataframe but was subsequently broken into smaller dictionaries for the randomized sampling of the data and each dictionary was appended to a list that was iterated through to obtain the five train and validation sets.

3 Algorithms

This section discusses the machine learning algorithms throughout this experiment.

3.1 K-Nearest-Neighbor (KNN)

The KNN method is a linear method that attempts to catalog input values that are most like a query-value, by using a distance discriminator like Mahalanobis, Hamming, or Euclidean distances. The experiments represented in this document only uses the Euclidean distance to determine similarity. The Euclidean distance is represented as: $\boldsymbol{d} = \sqrt{\sum_{i=1}^{N}(\boldsymbol{x_q} - \boldsymbol{x_i})^2}$, where x_q is the query vector and x_i is the input vector. Once distances are calculated, the results for each row with respect to the query point are appended to a python list and stored in a dictionary. The list is sorted in ascending order to reveal the closest vectors to the query value, and only the top k-values are used to determine class or regression values.

For classification, a plurality test was used to determine which class is most representative of the nearest neighbor list. Using the following for regression Gaussian Kernal, $\boldsymbol{K}(\boldsymbol{x_q}, \boldsymbol{x_i}) = \boldsymbol{e}^{-\frac{(x_i - x_q)^2}{2\sigma^2}}$ **(McCormick, 2014)**, where x_q is the query value, x_i is a data step, and σ^2 is the variance, weighting for regression values were determined as regression value coefficients.

3.2 Fine-tuning KNN

The optimal value of K was found by preselection of an array of k values ranging from one to ten. Classification test sets were fine-tuned according to results of accuracy obtained by using the test sets along with the new k-value. For the regression, the average value obtained by averaging the k-value for classification was used for each of the regression test sets. This was intentional, because the finetuning of regression sigma was also needed.

3.3 Fine-tuning Bandwidth (σ)

The σ – value was finetuned in a similar way as the k-value in the previous section but was only done for the regression datasets. Here, the test metric for determining σ values was mean-squared-error (MSE).

3.4 Edited KNN (E-KNN)

The defining feature of E-KNN is that it uses the same algorithm as KNN, however, if the predicted value is incorrect, it is dropped from the train set until the train set either does not change, or until a predetermined maximum allowed drops are made. The test set is then used to test the model. All the datasets were tested, and results can be seen in the results section.

3.5 Condensed KNN (C-KNN)

The defining feature of C-KNN is that it uses the same algorithm as KNN, however, if the predicted value is incorrect, it is dropped and added to a list. Once the length of the list remains constant, the train set is terminated, and the test set is then used to test the model. All datasets were tested, and results can be seen in the results section of this document.

4 Results

This section discusses the accuracy results of using the two naïve methods mentioned in 4.1 and 4.2 vs the accuracy of randomly sampling a class instead of the naïve approach.

4.1 Fine-Tuning k-value Results

The classification datasets were all tested with different k-values, and the average k-value was used for each of the regression datasets. The range of sigma values used was arbitrarily determined to be ten and accuracy was used as the test metric. The baseline for each set is set at k = 1 for consistency. The average value between the three datasets is k = 7, which is the value to be used for the regression datasets.

The following visualizations shows that fine-tuning k was an effective way to increase test metrics as the optimal k value had a positive effect on the accuracy and MSE calculations. See Table 1, Figure 1, Figure 2, and Figure 3 for visual analysis.

k-value	breast-cancer-wisconsin	Car	house-votes-84
k1	89.1	32.4	78.9
k2	92.0	34.2	81.3
k3	92.6	35.9	84.5
k4	94.9	37.0	86.5
k5	95.5	38.7	88.8
k6	96.1	38.9	89.3
k7	96.2	39.0	91.9
k8	96.3	39.2	92.5
k9	96.3	39.3	92.6
k10	96.3	39.3	92.6

Table 1. The k-values were tested from k1 to k10 to determine the optimal k-value for each dataset.

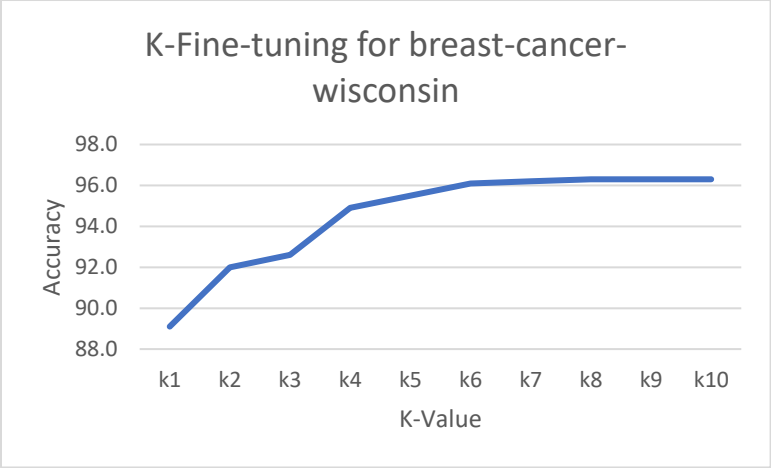

Figure 1. The optimal k-value for the breast-cancer-wisconsin dataset using the elbow method is k = 6.

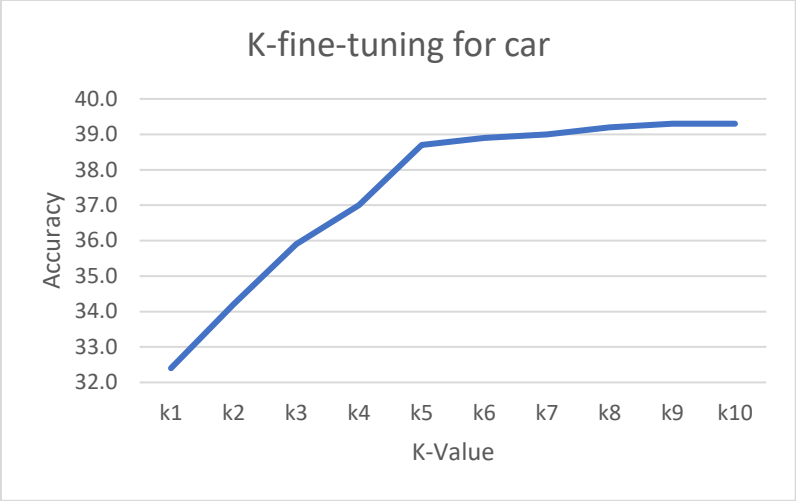

Figure 2. The optimal k-value for the car dataset using the elbow method is k = 6.

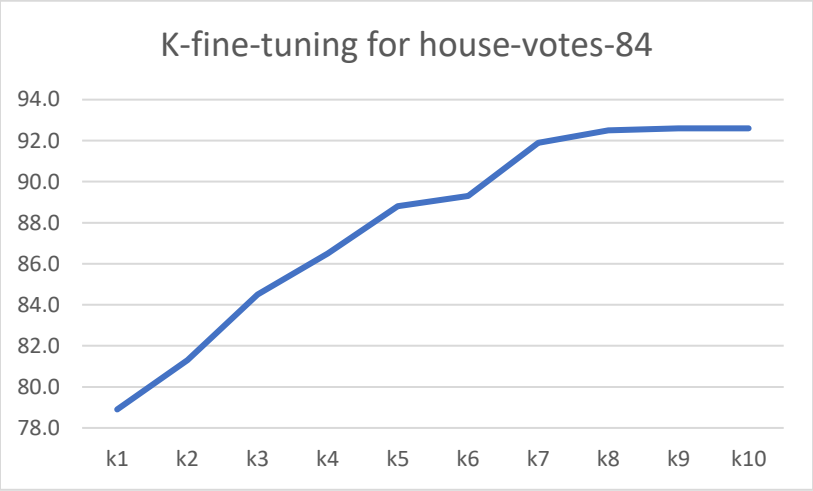

Figure 3. The optimal k-value for the house-votes-84 dataset using the elbow method is k = 8.

4.2 Fine-Tuning σ-value Results

A similar method was used for tuning sigma as was used for tuning for k. Each of the regression test sets underwent tuning of the Gaussian Kernel for sigma values [1, 5, 10], where 1 was the baseline value. The following visualizations shows that fine-tuning σ is an effective way to increase test metrics as the optimal σ - value had a positive effect on the MSE performance for each of the datasets. Shown below are results showing that $\sigma = 5$ increased MSE for each of the three regression datasets. See Figure 4, Figure 5, Figure 6, and Figure 7 for more in depth visual analysis for the results of fine-tuning σ.

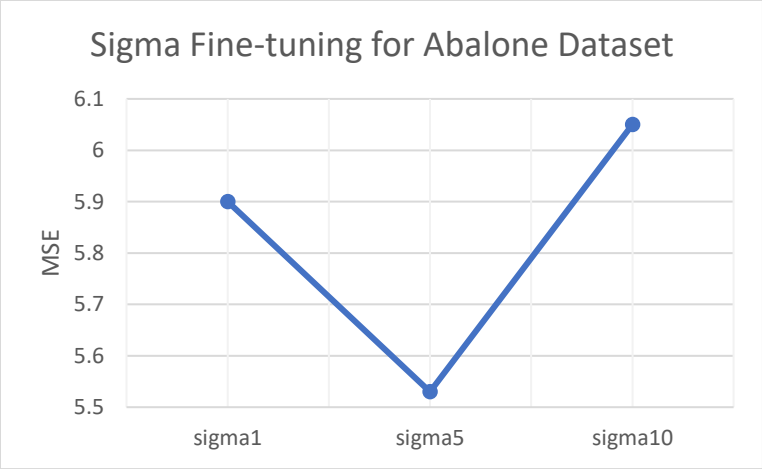

Figure 4. The optimal value for sigma for the Abalone dataset is about $\sigma = 5$

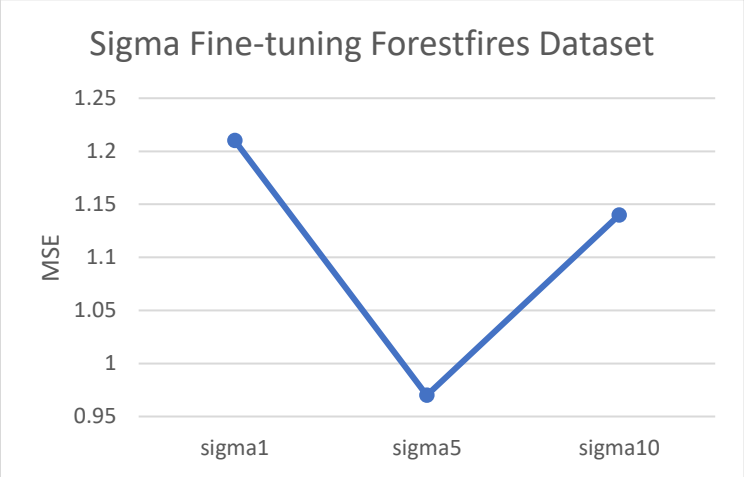

Figure 5. The optimal sigma for the Forestfires dataset is about $\sigma = 5$.

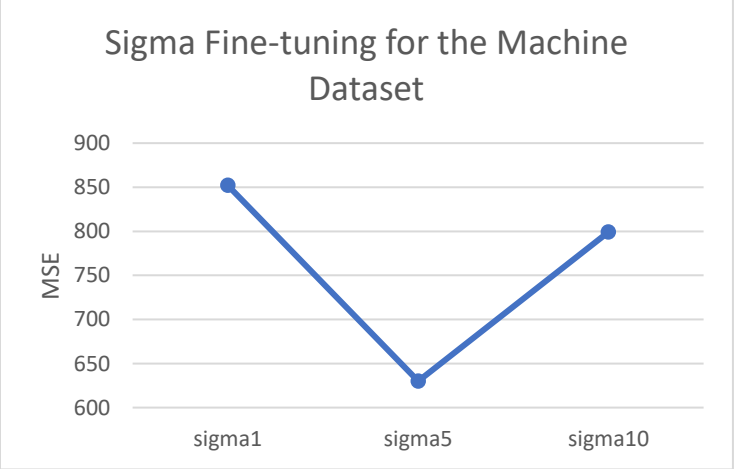

Figure 6. The optimal sigma value for the Machine dataset is about $\sigma = 5$.

4.3 C-KNN Results

The C-KNN did not produce the results that I had hypothesized above. To my surprise, the condensed method had a negative effect on performance for all datasets. In some cases, the accuracy dropped by 20 percent, while MSE went up by 3000 units for the machine dataset. The reason for the decrease in performance might be that these are all small datasets. Each dropped value may have a noticeable impact on the datasets, which does not allow the KNN model to generalize enough.

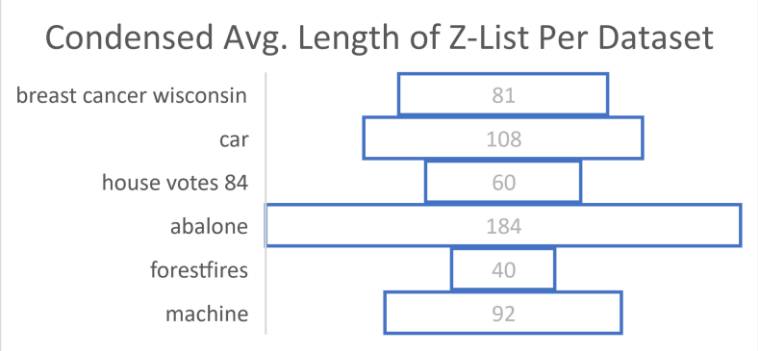

Figure 7. The bars in this plot represent the length of the Z-list for each of the datasets after execution of the condensed method.

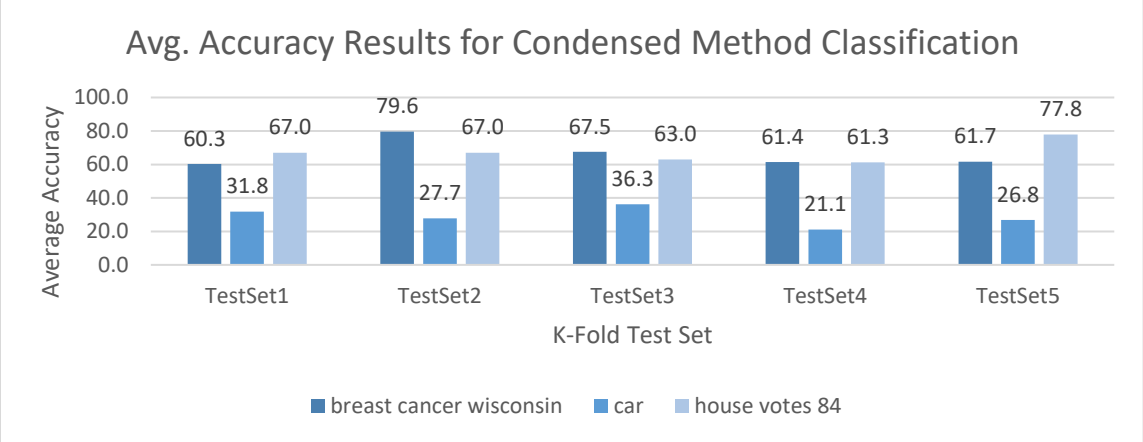

Figure 8. The accuracy for the classification datasets for each fold are shown above. It shows that the condensed method hurt the accuracy of all of the datasets.

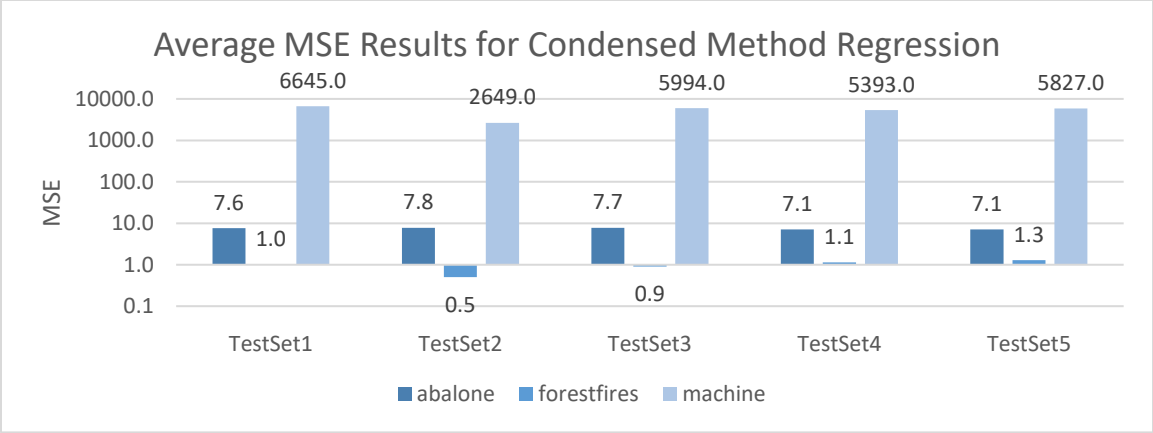

Figure 9. The MSE is shown for each of the regression dataset. An obvious feature here is that the Machine dataset has done worse. The other two will be more obvious when averaging all the datasets.

4.4 E-KNN Results

The E-KNN did not produce the results that I had hypothesized above. This method had a negative effect on performance for all the datasets. As in the condensed case, the accuracy dropped by up to 20 percent for the breast-cancer-wisconsin dataset, while MSE went up by 3000 units for the machine dataset. Similarly, a potential reason for the decrease in performance might be that these are all small datasets. Each dropped value may have a noticeable impact on the datasets, which does not allow the KNN model to generalize enough.

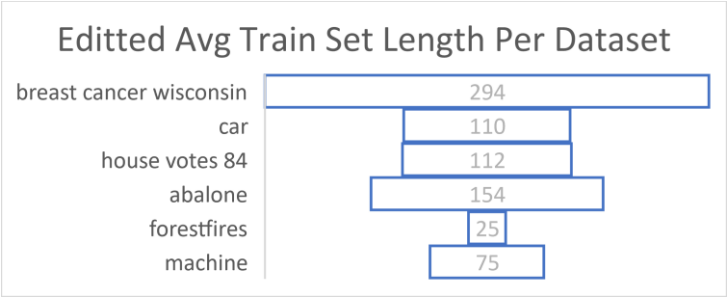

Figure 10. The bars in this plot represent the length of the train set for each dataset after execution of the edited method.

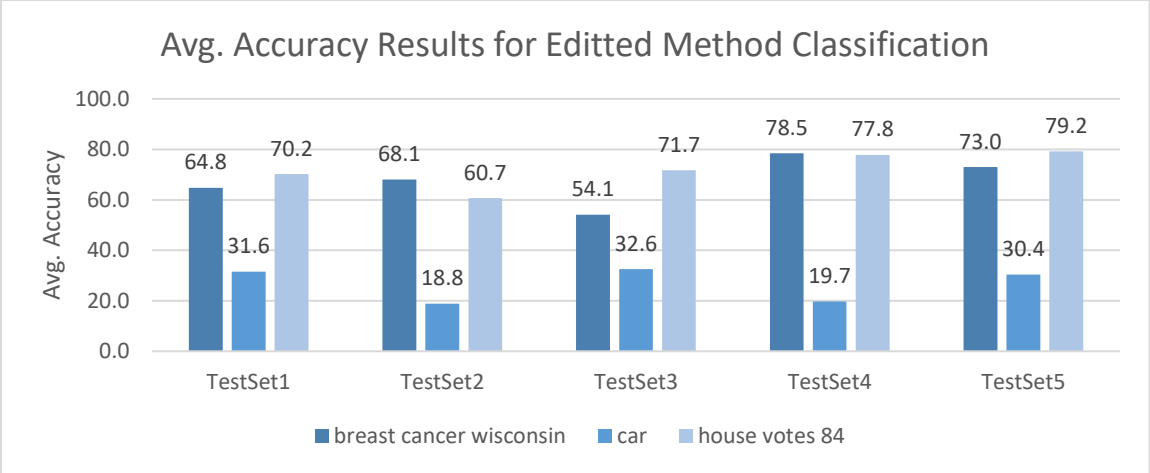

Figure 11. The accuracy for the classification datasets for each fold are shown above. It shows that the edited method also hurt the accuracy like the condensed method.

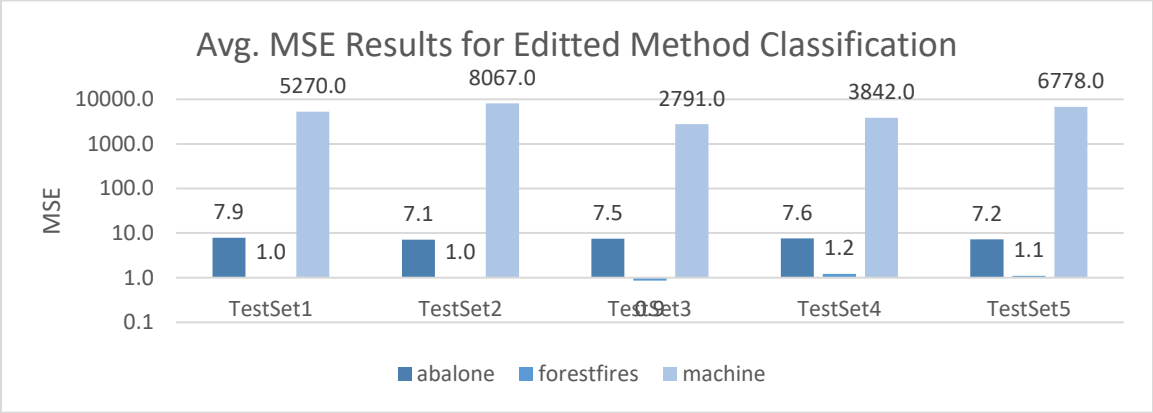

Figure 12. The MSE for the regression datasets for each fold are shown above. It shows that the edited method also hurt the datasets tested.

5 Conclusion

The models performed better while just using the optimal k-value. A possible cause for this is that the data that was thrown away may have been valuable data on account of the small size of the datasets and possibly high dataset variance. It should be mentioned that the abalone and forestfires datasets were not affected as much as the other four datasets by either the condensed or edited KNN methods.

This was an interesting project, as it showed us some tools to optimize the datasets such as fine-tuning the k and σ values helped each of the datasets. The same cannot be shown for the edited and condensed cases, however this does not prove that these methods are inefficient, but rather, they did not perform well using these datasets with the KNN method. It was also learned how to use the Gaussian kernel to weight predictions made on input values. Though not shown here, the Euclidean distance paired with the Gaussian kernel has a positive effect on inference performance metrics, especially when the σ is tuned. The pairing of the two were used on all the results seen in this document. All-in-all, I would say that this was a useful and challenging assignment.

References

Marko Bohenec, B. Z. (1997, June). Car Evaluation Database. Avignon, France.

McCormick, C. (2014, Febuary 2014). *Kernal Regression.* Retrieved from mccormickml.com: https://mccormickml.com/2014/02/26/kernel-regression/

Paulo Cortez, A. M. (2007, December). Forest Fires. Guimaraes, Portugal.

Phillip Ein-Dor, J. F. (1987, October). Relative CPU Performance Data. Tel Aviv, Israel.

Schlimmer, J. (1984). 1984 United States Congressional Voting Records Database. Washington D.C., USA.

Waugh, S. (1995). Extending and benchmarking Cascade-Correlation. Hobart, Tasmania, Australia.

Wolberg, W. H. (1992, July 15). Wisconsin Breast Cancer Database. Madison, Wisconsin, USA.

Appendix

The following is a brief description of the dataset used, as well as source information.

- Abalone is a regression dataset used to predict the age of a gastropod species by counting the number rings on the inner shell of the sample. There are 4177 samples with eight attributes to include, sex, length, diameter, height, whole weight, shucked weight, viscera weight, shell weight, ring number (Waugh, 1995). There are no missing values in this dataset.

- Breast-cancer-wisconsin is a dataset that includes 699 samples with 10 attributes and is used to classify either benign or malignant cancer types (Wolberg, 1992). This dataset is missing 16 values that are denoted by a "?."

- Car is a classification dataset used to evaluate hierarchy induction tools. There are 1728 samples with six attributes and four classes including, unacc, acc, good, and v-good (Marko Bohenec, 1997). This dataset has no missing values.

- Forestfires is a regression dataset used to predict the burn area of a forest fire based on 517 samples, and >12. The attributes include special coordinates X and Y of a park map, month, data, indices from the Fire Warning Information system, temperature, relative humidity, wind speed, rain, and burn area (Paulo Cortez, 2007). This dataset has no missing values.

- House-votes-84 is a classification dataset used to predict whether a congressman from the house of representatives belongs to the republican or democrat parties based whether they voted on 16 key votes: handicapped-infants, water-project-cost-sharing, adoption-of-the-budget-resolution, physician-fee-freeze, El-Salvador-aid, religious-groups-in-school, anti-satellite-test-ban, aid-to-nicaraguan-contras, mx-missile, immigration, synfuels-corporation-cutback, education spending, superfund-right-to-sue, crime, duty-free-exports, and export-administration-act-south-africa (Schlimmer, 1984). The attributes are either a "y" or a "no," and missing values are represented as a "?."

- Machine is a regression dataset that is used to predict relative CPU performance and includes 209 samples with 10 attributes. Attributes include, vendor name, model name, machine, cycle time, minimum main memory, maximum main memory, cache memory, minimum channels, maximum channels, published relative performance, and estimated relative performance (Phillip Ein-Dor, 1987). This dataset has no missing values.