UNIVERSITY OF MIAMI

ECE730 Final Project Report

Ruijie Yin
Division of Biostatistics
Department of Public Health Sciences
11/21/2018

1 INTRODUCTION

The technique of decision tree is widely used in machine learning, the goal is to create a non-parametric model that can predict the value of a target variable based on several input variables. Two types of decision trees are mainly used in machine learning, one is the classification tree when the predicted outcome is the class label to which the data belongs;the other is regression tree when the predicted outcome can be considered a real number. Trees are easy to understand and interpret, they can handle both numerical and categorical data and potentially performs well with large datasets; at the same time suffers from creating over-complex trees (overfitting) in which case mechanisms such as pruning are necessary to avoid this problem.This project aims to classify handwritten digits that are automatically scanned from envelopes using classification tree; by doing so in reality, we can further classify the envelopes with the same zip code and prepare for delivery automatically, which is more efficient than using human workers and can thus reduce human labor and production cost.

2 METHODS

2.1 DATA

The zipcode datasets for this project comes from the neural network group at ATT research labs; It contains normalized handwritten digits, that were automatically scanned from envelopes by the U.S. Postal Service. The original scanned digits are binary and of different sizes and orientations; the images used have been deslanted and size normalized, resulting in 16 x 16

grayscale images (Le Cun et al., 1990). Several simulation training datasets were also generated to investigate the performance of our classifier.

2.2 Variable Characteristics

Variables used in the real datasets are the digit id (0-9) and the 256 grayscale values; In the simulated training datasets, variables used are the class labels and all the features.

2.3 Modeling

Descriptive statistics and graphs were used to demonstrate the distribution of digit id in the real datasets.

2.3.1 Cross-validation

Cross-validation is used to select the best parameter in a model when the size of our dataset is limited. It is a re-sampling procedure and has a single parameter called k that refers to the number of groups that a given data sample is to be split into, which is often called the k-fold cross-validation. Thus, when a specific value for k is determined, it is then be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation. Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.It is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split. The general procedure is as follows:(1)Shuffle the dataset randomly.(2)Split the dataset into k groups,let $v \subset \{1,...,n\}$ be a random set with distribution P_v, let $L_{(v)}$ be the learning set restricted to v (the testing data) and $L_{-(v)}$ be the learning set restricted to $\{1,...,n\} \setminus v$ (the training data): for each unique group: fit a model on the training set and evaluate it on the testing set; calculate a cross-validated misclassification error for each of the model:

$$Err_{cv}(\hat{\mu}) = E_v[\frac{1}{|L_{(v)}|}\sum_{i \in L_{(v)}} Q(Y_i, \hat{\mu}(X_i, L_{-(v)}))] \tag{2.1}$$

where $\hat{\mu}(X) = \hat{\mu}(X, L)$ is an estimator for the unknown target function $\mu(X) = E(h(Y)|X)$ and $Q(Y, \hat{\mu}) \geq 0$ is the loss function. Then retain the parameter in the model giving the smallest misclassification error and finally use this parameter to refit a model using all the training dataset we have to obtain a final best model. Apparently, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. In this project, cross-validation is used to select the best maximum of depth in a classification tree (the length of the longest path from a root to a leaf).

2.3.2 COEFFICIENT OF VARIANCE

A coefficient of variation (CV) is a statistical measure of the dispersion of data points in a data series around the mean. It is calculated as:

$$CV = \frac{\sigma}{\mu} \tag{2.2}$$

where σ is the standard deviation and μ is the expected value for a feature, respectively. The coefficient of variation (CoV) represents the ratio of the standard deviation to the mean, and it is a useful statistic for comparing the degree of variation from one data series to another, even if the means are drastically different from one another. It is also used in feature selection, for example, only features with CoV values greater than some threshold can be selected into modelling. There is, however, no uniform or standard criterion to determine a threshold and choosing different thresholds may lead to different results; Besides, when any of the data are negative, CoV stops making sense, in this case, instead of the mean, we should use the mean of the absolute values and call the new measure ACoV for absolute coefficient of variation.

2.3.3 DECISION TREE

Decision trees can be used for both classification and regression problems, given the nature of the problem this project aim to solve, a classification tree is implemented. Classification trees are non-parametric methods for classifying a p-dimensional feature x into one of $K \geq 2$ possible categorical values $Y \subset C_1, C_2, ..., C_K$

$$\hat{c}(x) : X \rightarrow \{C_1, C_2, ..., C_K\} \tag{2.3}$$

Given the learning data $L_n = (X_i, Y_i)_{1 \leq i \leq n}$ we have, we suppress the dependence on the learning data, technically

$$\hat{C}(x) = \hat{C}(x, L_n) \tag{2.4}$$

It utilizes binary recursive partitioning (RPART) to grow a tree, the procedure is as follows: (1)start at the root node (the top of the tree); (2)The split $\hat{s}(x)$ of a feature x is found by searching all its unique values

$$s_1 < s_2 ... < s_m, 2 \leq m \leq n \tag{2.5}$$

define $\hat{s}(x)$ to be the split that most reduces tree impurity, for a feature x it splits a node h into left and right daughter nodes h_L and h_R, the Gini node impurity for h is defined as:

$$\hat{\delta}(h) = \sum_{k=1}^{K} \hat{p}_j(h)(1 - \hat{p}_j(h)) \tag{2.6}$$

where $\hat{p}_j(h)$ is the proportion of class k cases in h.The decrease in node impurity is

$$\hat{\delta}(h) - [\hat{P}(h_L)\hat{\delta}(h_L) + \hat{P}(h_R)\hat{\delta}(h_R)] \tag{2.7}$$

Maximizing the decrease in node impurity is equivalent to minimizing the Gini index:

$$\hat{\theta}(s,h) = \hat{P}(h_L)\hat{\delta}(h_L) + \hat{P}(h_R)\hat{\delta}(h_R) \tag{2.8}$$

where $\hat{P}(h_L) = \frac{|h_L|}{|h|}$ and $\hat{P}(h_R) = \frac{|h_R|}{|h|}$. When the best split point $\hat{s}(x^*)$ for the best feature x^* is determined, the split on x^* creates left and right daughters:

$$Left := \{X_i : x_i^* \leq \hat{s}(x^*)\} \tag{2.9}$$

$$Right := \{X_i : x_i^* \geq \hat{s}(x^*)\} \tag{2.10}$$

Continue recursively until the tree cannot be partitioned further, or until a stopping criterion is reached, resulting in the terminal nodes (ends of the tree). The loss function in misclassification error that is used in cross-validation is:

$$Q(Y, \hat{C}(X)) = 1_{\{\hat{C}_{(X)} \neq Y\}} \tag{2.11}$$

3 Results and Discussions

3.1 Results on simulated data

To investigate the effect of noise level (σ^2) in affecting misclassification error, three simulation datasets: dataset 1,dataset 2 and dataset 3 are generated. Each of the dataset has one response variable, the class label K (K=1,2,3) and three features x_1,x_2 and x_3.In all three datasets,assume the first two features x_1 and x_2 determine the class label and the remaining feature x_3 is irrelevant; each class label k has 100 data points so that each dataset contains a total of 300 data points; All features are generated from the Gaussian Distribution with the common mean μ_1,μ_2 and μ_3 for x_1,x_2 and x_3,respectively,:

$$\mu_1 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}, \mu_2 = \begin{pmatrix} 2 \\ 3 \\ 0 \end{pmatrix}, \mu_3 = \begin{pmatrix} 3 \\ 4 \\ 0 \end{pmatrix} \tag{3.1}$$

but with different covariance structures σ_1,σ_2 and σ_3 in the three datasets, for simplicity, we assume features are independent of each other:

$$\sigma_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \sigma_2 = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \sigma_3 = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 9 \end{bmatrix} \tag{3.2}$$

Classification trees with and without feature selection are fitted on these three datasets; When using feature selection, features with $ACoV < 1$ are selected; 10-fold cross-validation is used to determine the best maximum depth for each tree. Results are shown in the following table:

misclassification error	w.o. feature selection	w. feature selection
dataset 1	0.3533	0.3433
dataset 2	0.5067	0.5033
dataset 3	0.5767	0.5733

Table 3.1: Effect of different noise level (σ^2)

As we expected, an increasing noise level causes increasing misclassification error. That is, when the dispersion of the data points for features increases, it becomes more difficult to determine the best split point, thus will result in increasing the misclassification error.

For the effect of different number of features, three simulation datasets: dataset 1,dataset 4 and dataset 5 are generated, where dataset 1 is the same dataset used in the previous investigation; Dataset 4 has four features x_1,x_2, x_3 and x_4 while dataset 5 has five: x_1,x_2, x_3, x_4 and x_5; All three datasets have their first two features x_1 and x_2 determine the class label and the rest features are set to be non-informative. Each dataset contains a total of 300 data points, in which each class label k has 100 data points. The means used to generate x_1,x_2,x_3 and x_4 in dataset 4 from the Gaussian Distribution are $\mu_4 1$,$\mu_4 2$ and $\mu_4 3$, respectively:

$$\mu_{41} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \end{pmatrix}, \mu_{42} = \begin{pmatrix} 2 \\ 3 \\ 0 \\ 0 \end{pmatrix}, \mu_{43} = \begin{pmatrix} 3 \\ 4 \\ 0 \\ 0 \end{pmatrix} \tag{3.3}$$

, the means used to generate x_1,x_2,x_3,x_4 and x_5 in dataset 5 from the Gaussian Distribution are $\mu_5 1$,$\mu_5 2$ and $\mu_5 3$, respectively:

$$\mu_{51} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mu_{52} = \begin{pmatrix} 2 \\ 3 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mu_{53} = \begin{pmatrix} 3 \\ 4 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.4}$$

and different covariance structures σ_1,σ_4 and σ_5 for the features are used in the three datasets, respectively; again, assume all the features are independent of each other but have the same level of noise:

$$\sigma_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \sigma_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \sigma_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Results are shown in the following table:

misclassification error	w.o. feature selection	w. feature selection
dataset 1	0.3533	0.3433
dataset 4	0.4267	0.42
dataset 5	0.34	0.36

Table 3.2: Effect of different number of features

When the number of deterministic features remains the same, the misclassification error will first increase then decrease when increasing the number of irrelevant features. This is probably because once we have determined the best maximum depth of a classification tree using cross-validation, especially when the depth is relatively small compared to the total features under consideration, only a few features will eventually be used to grow the tree, even if we increase the total number of features, most of the features will remain unused.

To compare the effect of different numbers of deterministic features, we generated three simulation datasets: dataset 5, dataset 6 and dataset 7. Dataset 5 is the same as what we generated previously; In contrast to dataset 5, dataset 6 has one more deterministic feature and dataset 7 has two more. The means μ_{61}, μ_{62} and μ_{63} used to generate x_1,x_2,x_3,x_4 and x_5 in dataset 6 are:

$$\mu_{61} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 0 \end{pmatrix}, \mu_{62} = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 0 \\ 0 \end{pmatrix}, \mu_{63} = \begin{pmatrix} 3 \\ 4 \\ 5 \\ 0 \\ 0 \end{pmatrix} \tag{3.6}$$

respectively; While for dataset 7, the means μ_{71}, μ_{72} and μ_{73} are:

$$\mu_{71} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 0 \end{pmatrix}, \mu_{72} = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 0 \end{pmatrix}, \mu_{73} = \begin{pmatrix} 3 \\ 4 \\ 5 \\ 6 \\ 0 \end{pmatrix} \tag{3.7}$$

All these three simulation datasets use the same covariance structure for the five features:

$$\sigma = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

The results are displayed in the table below:

misclassification error	w.o. feature selection	w. feature selection
dataset 5	0.34	0.36
dataset 6	0.36	0.36
dataset 7	0.3533	0.34

Table 3.3: Effect of different number of deterministic features

The results indicates that the misclassification error first increases and then decreases as the number of deterministic features in the data increases; The reason is several deterministic features compete to be split in the begin as we increase the number of deterministic features, but once the best parameter is determined, only a few features will actually be used to grow the tree.

In order to find the impact of different training data sample sizes on growing a tree, we stimulated four different datasets: dataset 1, dataset 8, dataset 9 and dataset 10. The dataset 1 is exactly the same dataset we created in the previous and the only difference between the four datasets is their total sample size: they are 300, 450, 600 and 1500, respectively. The results are:

misclassification error	w.o. feature selection	w. feature selection
dataset 1	0.3533	0.3433
dataset 8	0.3911	0.3822
dataset 9	0.3783	0.3783
dataset 10	0.3767	0.3767

Table 3.4: Effect of different training data sample size.

The misclassification error tends to be stabilized when we increase the training data sample size; This indicates that the more data we have, the more accurate our model should be.

Finally, we investigated how different feature selection method affect the performance of our classifier. The dataset used in this case is the dataset 7 of which there are four deterministic features given that the total number of features is 5. The cut-off points of ACoV is set to be greater than 0.25, 0.29, 0.4 and 0.6, so that 4,3,2 and 1 features will be selected, respectively. The results are shown as following:

feature selection method	misclassification error
≥ 0.25	0.34
≥ 0.29	0.38
≥ 0.4	0.3633
≥ 0.6	0.45

Table 3.5: Effect of different feature selection methods.

When using all these four deterministic features in the model we are expected to find the model that can give us the smallest misclassification error since we are utilizing all the information we have, any elimination of deterministic features will increase the misclassification error.

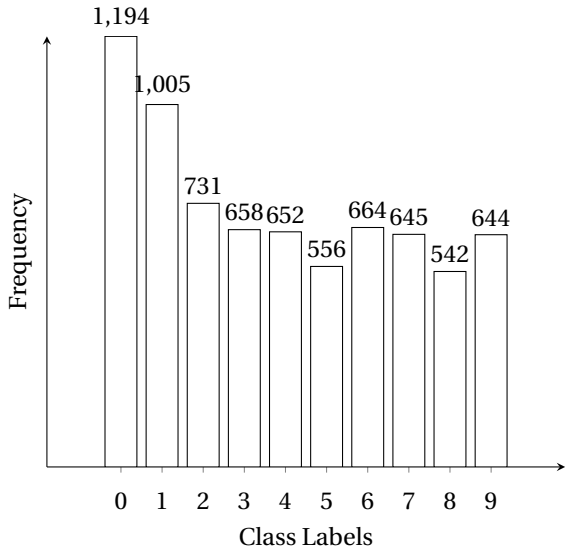
3.2 Results on real data

We applied our classification tree on the real zipcode training dataset.There are 10 class labels, from 0 to 9, a total of 10 digits, distributed as following:

Class Label	Proportion in the data
0	0.16
1	0.14
2	0.1
3	0.09
4	0.09
5	0.08
6	0.09
7	0.09
8	0.07
9	0.09

Table 3.6: Proportions of class labels in the training data.

Figure 3.1: The distribution of class labels in the training dataset.

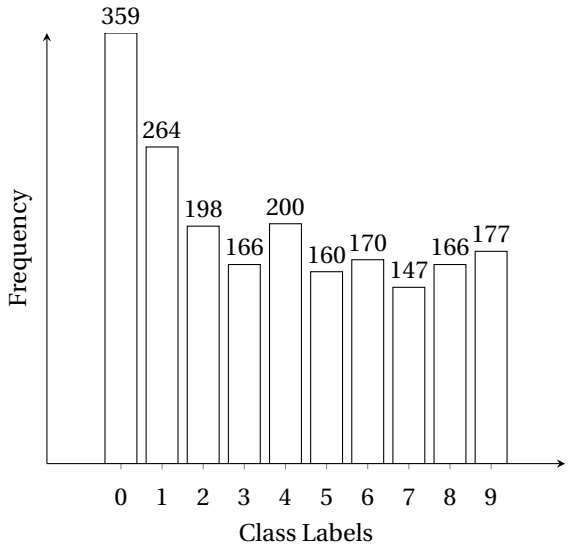


and we validated the model on the real zipcode testing dataset. The 10 class labels are distributed in the following way:

Class Label	Proportion in the data
0	0.18
1	0.13
2	0.1
3	0.08
4	0.1
5	0.08
6	0.08
7	0.07
8	0.08
9	0.09

Table 3.7: Proportions of class labels in the testing data.

Figure 3.2: The distribution of class labels in the testing dataset.



Thanks to the setting of zipcode in the United States, the 10 digits are not uniformly distributed, apparently the digit 0 and 1 have a much larger proportion than all other digits;

We fit a classification tree without feature selection on the training zipcode dataset, the results shows that the best maximum depth that gives a minimum misclassification error is 6 and the best number of splits is 13:

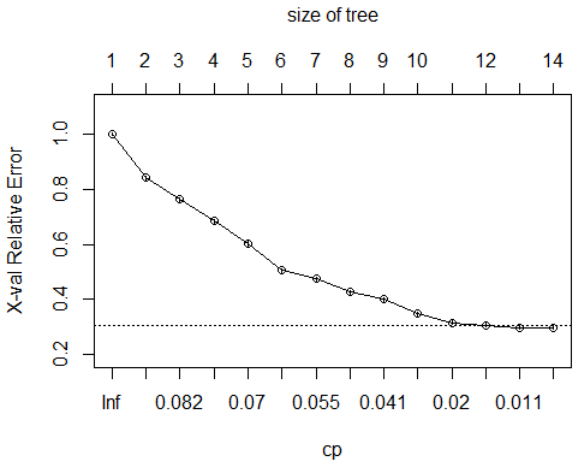


Figure 3.3: The classification tree without feature selection.

The plot of the tree dendrogram is shown as below:

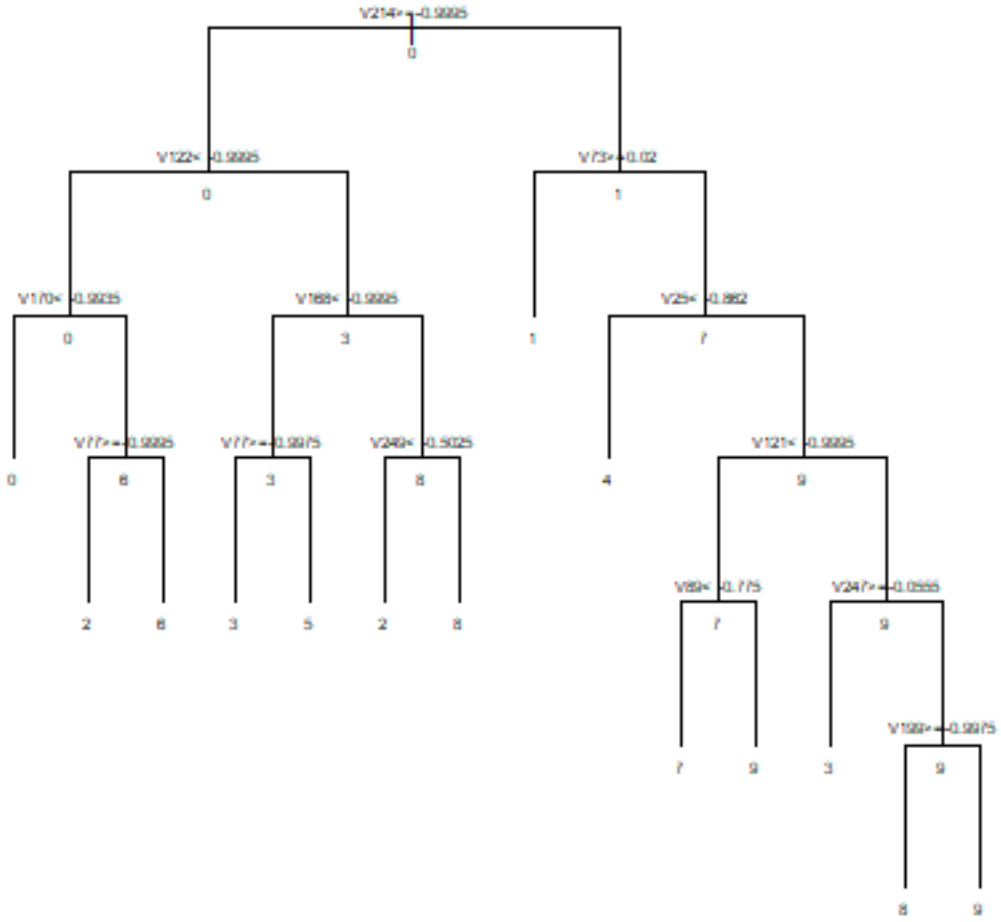


Figure 3.4: The classification tree without feature selection.

The corresponding misclassification error on the testing zipcode dataset is 0.2750374. We then fit a classification tree with feature selection, the cut-off value of ACoV is chosen to be 0.05, 0.25, 0.4 and 0.9; Different cut-off values and their corresponding misclassification error is summarized in the following table:

Cut-off value	number of features selected	misclassification error
≥ 0.05	182	0.2924763
≥ 0.25	156	0.2924763
≥ 0.4	131	0.2924763
≥ 0.9	73	0.3104136

Table 3.8: Effect of different feature selection methods.

For illustration, a graph of classification tree with feature selection (ACov>0.4) is shown as below,

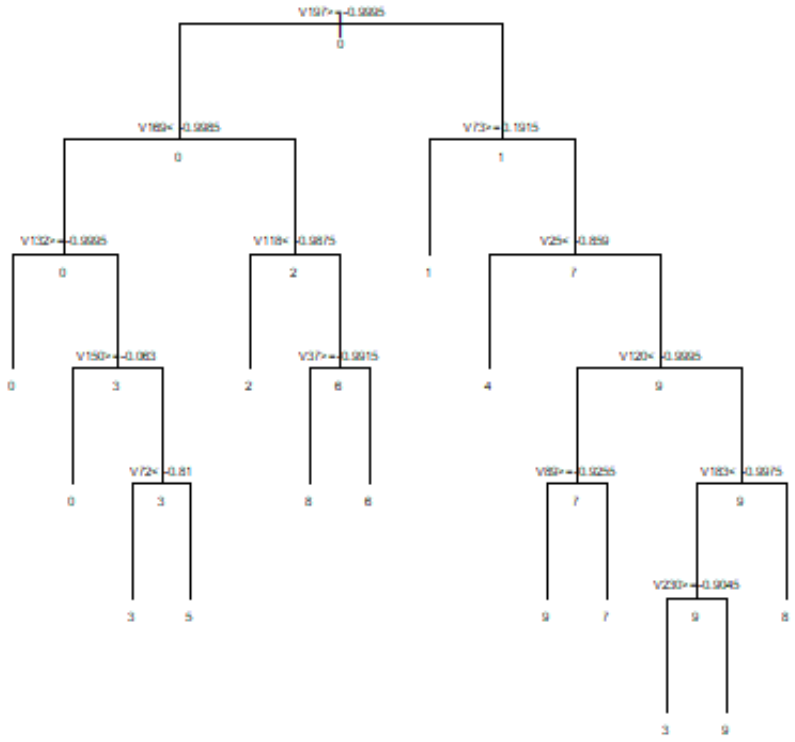


Figure 3.5: The classification tree without feature selection.

The results shows that with any different standards of selecting features, it increases the missclassification error than simply using all features along in our classification tree.

4 CONCLUSIONS

Based on the results obtained from the real zipcode dataset, it seems that feature selection cannot reduce the missclassification error in this case, any elimination of the feature will cause a higher rate of missclassification; we should further try with boosted trees, bagged trees and random forrest to see if they improve in reducing the missclassification error; The results obtained from the simulated dataset exhibited the instability of the classification tree as presented in the results and discussion section, as the misclassification error varies by the noise level (σ^2), the number of features, the number of relevant feature, the number of training data samples n and the feature selection method.

5 REFERENCE

1.Gelfand, S.B., Ravishankar, C.S. and Delp, E.J., 1989, November. An iterative growing and pruning algorithm for classification tree design. In Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE International Conference on (pp. 818-823). IEEE.
2.Groover, M.P., 2007. Automation, production systems, and computer-integrated manufacturing. Prentice Hall Press. Vancouver
3.Georgakopoulos, D., Hornick, M. and Sheth, A., 1995. An overview of workflow management: From process modeling to workflow automation infrastructure. Distributed and parallel Databases, 3(2), pp.119-153. Vancouver
4.Kohavi, R., 1995, August. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).
5.Efron, B. and Gong, G., 1983. A leisurely look at the bootstrap, the jackknife, and cross-validation. The American Statistician, 37(1), pp.36-48.
6.Abdi, H., 2010. Coefficient of variation. Encyclopedia of research design, 1, pp.169-171.
7.Lovie, P., 2005. Coefficient of variation. Encyclopedia of statistics in behavioral science.
8.Freund, Y. and Mason, L., 1999, June. The alternating decision tree learning algorithm. In icml (Vol. 99, pp. 124-133).