Fine Particulate Matter Concentration Level Prediction by using Tree-based Ensemble Classification Algorithms

Yin Zhao
School of Mathematical Sciences
Universiti Sains Malaysia (USM)
Penang, Malaysia

Yahya Abu Hasan
School of Mathematical Sciences
Universiti Sains Malaysia (USM)
Penang, Malaysia

***Abstract*—Pollutant forecasting is an important problem in the environmental sciences. Data mining is an approach to discover knowledge from large data. This paper tries to use data mining methods to forecast $PM_{2.5}$ concentration level, which is an important air pollutant. There are several tree-based classification algorithms available in data mining, such as CART, C4.5, Random Forest (RF) and C5.0. RF and C5.0 are popular ensemble methods, which are, RF builds on CART with Bagging and C5.0 builds on C4.5 with Boosting, respectively. This paper builds $PM_{2.5}$ concentration level predictive models based on RF and C5.0 by using R packages. The data set includes 2000-2011 period data in a new town of Hong Kong. The $PM_{2.5}$ concentration is divided into 2 levels, the critical points is 25μg/m^3 (24 hours mean). According to 100 times 10-fold cross validation, the best testing accuracy is from RF model, which is around 0.845~0.854.**

Keywords—Random Forest; C5.0; PM2.5 prediction; data mining.

I. Introduction

Air pollution is a major problem for some time. Various organic and inorganic pollutants from all aspects of human activities are added daily to the air. One of the most important pollutants is particulate matter. Particulate matter (PM) can be defined as a mixture of fine particles and droplets in the air and this can be characterized by their sizes. $PM_{2.5}$ refers to particulate matter whose size is 2.5 micrometers or smaller. Due to its effect on health, it is crucial to prevent the pollution getting worse in a long run. According to WHO's report, the mortality in cities with high levels of pollution exceeds that observed in relatively cleaner cities by 15–20% [1]. Forecasting of air quality is much needed in a short term so that necessary preventive action can be taken during episodes of air pollution. WHO's Air Quality Guideline (AQG) [2] says the mean of $PM_{2.5}$ concentration in 24-hour level should be less than 25μg/ m^3, although Hong Kong's proposed Air Quality Objectives (AQOs) [3] is 75μg/m^3 right now. Because the target data is from a new town in Hong Kong, which means there are lots of people living in this area, so it is need to be a stricter standard of air pollution in such area. As a result, we use 25μg/ m^3 based on 24 hours mean as our standard points. The number of particulate at a particular time is dependent on many environmental factors, especially the meteorological data and time serious factors.

Predictive models for $PM_{2.5}$ can vary from the simple to the complex; hence we have CART, C4.5, Artificial Neural Networks, Support Vector Machine among others. In this paper, we try to build models for predicting next day's $PM_{2.5}$ concentration level by using two popular tree-based classification algorithms, which are, Random Forest (RF) [4-5] and C5.0 [6-7]. CART and C4.5 are simple decision tree models because there is only one decision tree in each model. While RF and C5.0 are ensemble methods based on CART and C4.5, and each of them has a bunch of basic decision trees in the model. Some of the differences among these two algorithms are shown in Table 1.

TABLE I. BRIEF DIFFERENCE BETWEEN RF&C5.0

Algorithms	Number of Trees	Methods	Basic Classifier
RandomForest	Multiple	Bagging and Voting	CART
C5.0	Multiple	Boosting and Voting	C4.5

R [8] is an open source programming language and software environment for statistical computing and graphics. It is widely used for data analysis and statistical computing projects. In this paper, we will use some R packages as our analysis tools, namely "randomForest" package [9] and "C50" package [10]. Moreover, we also use some packages for plotting figures, such as "reshape2" package [11] and "ggplot2" package [12].

The structure of the paper is: Section 2 reviews some basic concept of tree-based classification methods, while Section 3 and 4 will describe the data and the experiments. The conclusion will be given in Section 5.

II. Methodology

A. Random Forest (RF)

RF is an effective prediction tool in data mining, which is based on CART. It employs the Bagging [13] method to produce a randomly sampled set of training data for each of the trees.

CART uses *Gini* index which is an impurity-based criterion that measures the divergences among the probability distributions of target attribute's values.

Definition 1 (*Gini* Index): Given a training set *S* and the target attribute takes on *k* different values, then the *Gini* index of *S* is defined as

$$Gini(S) = 1 - \sum_{i=1}^{k} p_i^{\,2}$$

Where p_i is the probability of *S* belonging to class *i*.

Definition 2 (*Gini* Gain): *Gini* Gain is the evaluation criterion for selecting the attribute *A* which is defined as

$$GiniGain(A,S) = Gini(S) - Gini(A,S)$$
$$= Gini(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} Gini(S_i)$$

Where S_i is the partition of *S* induced by the value of attribute *A*.

CART algorithm can deal with the case of features with nominal variables as well as continuous ranges.

Pruning a tree is the action to replace a whole sub-tree by a leaf. CART uses a pruning technique called "minimal cost-complexity pruning" which assuming that the bias in the re-substitution error of a tree increases linearly with the number of leaves. Formally, given a tree *T* and a real number $\alpha>0$ which is called the "complexity parameter", then the cost-complexity risk of *T* with respect to α is:

$$R_\alpha(T) = R(T) + \alpha \cdot |T|$$

where $|T|$ is the number of terminal nodes (i.e. leaves) and *R(T)* is the re-substitution risk estimate of *T*.

Bagging, which stands for "**b**ootstrap **agg**regat**ing**", is an ensemble classification method. It repeatedly samples from a data set with replacement according to a uniform probability distribution. Each sample has a probability $1-(1-1/N)^N$ of being selected, where *N* is the number of observations in the training set. If *N* is sufficiently large, this probability converges to $1-1/e \simeq 0.632$, that is, a bootstrap sample contains approximately 63% of the original training data, while other data is a natural good testing dataset which is known as OOB (Out of Bag [14]) in RF. Since every sample has an equal probability of being selected (i.e. $1/N$), bagging does not focus on any particular instance of the training data. Therefore, it is less sensitive to model overfitting when applied to noisy data.

RF constructs a series of tree-based learners. At each tree node, a random sample of *m* features is drawn, and only those *m* features are considered for splitting. Typically $m=\sqrt{p}$ (as default in R "randomForest" package), where *p* is the number of features. The essential difference between Bagging and RF is the latter not only selecting samples randomly but also the features being selected randomly. RF will not prune the trees during the whole growing procedure.

B. C5.0

C5.0 is an advanced decision tree algorithm developed based on C4.5. It includes all functionalities of C4.5 and applies some new technologies, the most important application among them is Boosting (i.e. AdaBoost [15]), which is a technique for generating and combining multiple classifiers to improve predictive accuracy.

C4.5 uses information gain ratio which is an impurity-based criterion that employs the entropy measure as an impurity measure.

Definition 3 (Information Entropy): Given a training set *S*, the target attribute takes on *k* different values, and then the entropy of *S* is defined as

$$Entropy(S) = -\sum_{i=1}^{k} p_i \log_2 p_i$$

Where p_i is the probability of *S* belonging to class *i*.

Definition 4 (Information Gain): The information gain of an attribute *A*, relative to the collection of examples *S*, is defined as

$$InfoGain(A,S) = Entropy(S) - Entropy(A,S)$$
$$= Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} Entropy(S_i)$$

Where S_i is the partition of *S* induced by the value of attribute *A*.

Definition 5 (Gain Ratio): The gain ratio "normalizes" the information gain as follows:

$$GainRatio(A,S) = \frac{InfoGain(A,S)}{SplitEntropy(A,S)}$$
$$= \frac{InfoGain(A,S)}{-\sum_{i=1}^{n} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}}$$

Similar to CART, C4.5 can also deal with both nominal and continuous variables.

Error Based Pruning (EBP) is the pruning method which is implemented in C4.5 algorithm. The idea behind EBP is to minimize the estimated number of errors that a tree would make on unseen data. The estimated number of errors of a tree is computed as the sum of the estimated number of errors of all its leaves.

AdaBoost stands for "**ada**ptive **boost**ing", it increases the weights of incorrectly classified examples and decreases the ones of those classified correctly.

C5.0 is much efficient than C4.5 also on the aspect of unordered rule sets. That is, when a case is classified, all applicable rules are found and voted. This improves both the interpretability of rule sets and their predictive accuracy.

III. Data Preparation

All of data for the 2000-2011 period were obtained from Hong Kong Environmental Protection Department (HKEPD) and Hong Kong Met-online. The air monitoring station is Tung Chung Air Monitoring Station (Latitude 22°17'19"N, Longitude 113°56'35"E) which is in a new town of Hong Kong, and the meteorological monitoring station is Hong Kong International Airport Weather Station (Latitude 22°18'34"N, Longitude 113°55'19"E) which is the nearest station from Tung Chung. As mentioned in Section 1, accurately predicting high $PM_{2.5}$ concentration is of most value from a public health standpoint, thus, the response variable has two classes, which are, "Low" indicating the daily mean concentration of $PM_{2.5}$ is below 25μg/m^3 and "High" representing above it. Figure 1 shows that the days of two levels in 2000-2011.

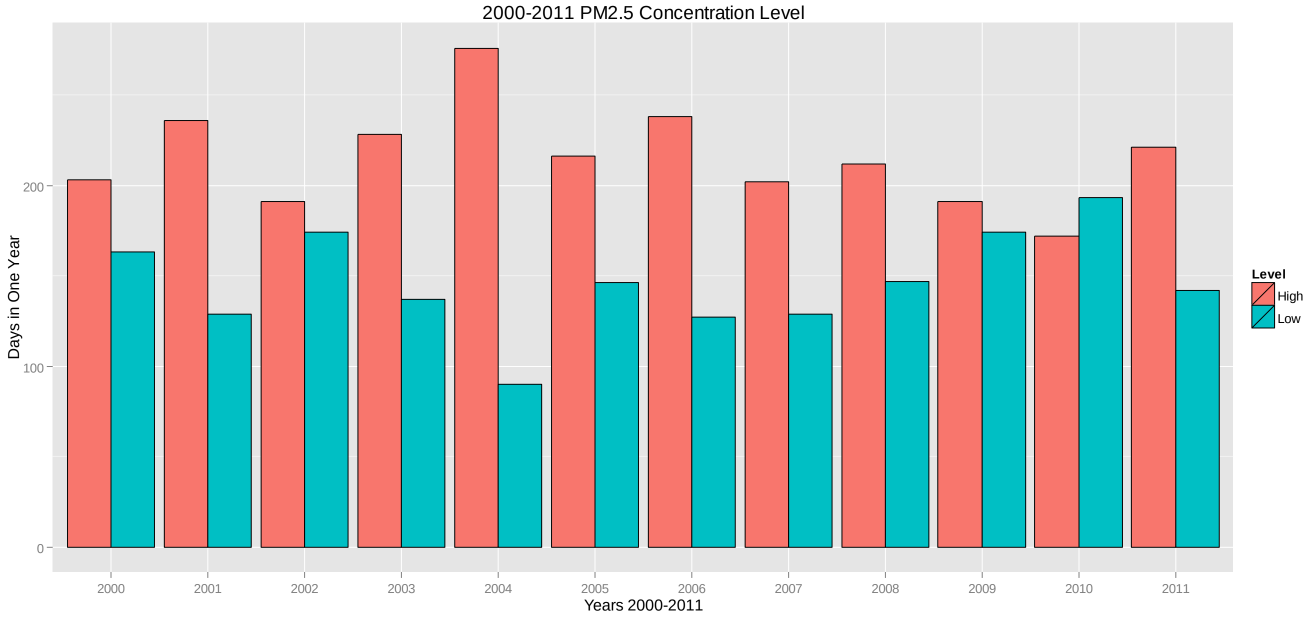

Fig.1. $PM_{2.5}$ concentration levels in 2000-2011

We learn that the air quality is the best in 2010 (i.e. it has the most "Low" days), while the worst is in 2004 among these 12 years. In summary, the percentage of "Low" and "High" level is around 40.0% and 59% during 12 years in this area, respectively (around 1% missing values). Thus, if a predictive model obtains the accuracy is less than 60%, which means it approximately equals the randomly guess, that would be failure. The purpose to use data mining method is to raise the accuracy, say, at least more than 60%.

We convert all hourly $PM_{2.5}$ data to daily mean values and the meteorological data is the original daily data. In addition, all of air data and meteorological data are numeric. We certainly cannot ignore the effects of seasonal changes and human activities; hence we add two time variables, namely the month (Figure 2) and the day of week (Figure 3).

Figure 2 clearly shows that $PM_{2.5}$ concentration reaches a low level from May to August, during which is the rainy season in Hong Kong. But the pollutant is serious from October to next January, especially in December and January. We should know that the rainfall may not be an important factor in the experiment as the response variable is the next day's $PM_{2.5}$, and it is easy to understand that rainy season includes variant meteorological factors. Figure 3 presents the trends of people's activities in some senses. We learn that the air pollution waves slightly during the week. The concentration levels are similar from Tuesday to Thursday, while the lowest level appears on Sunday. This situation can be related to Tung Chung is a living area in Hong Kong, which means the air is less influenced by factories or other pollution source (i.e. different from business area or industrial area).

At last, there are 4326 observations by deleting all NAs and 14 predictor variables (Table 2) and 1 response variable which is the next day's $PM_{2.5}$ concentration level.

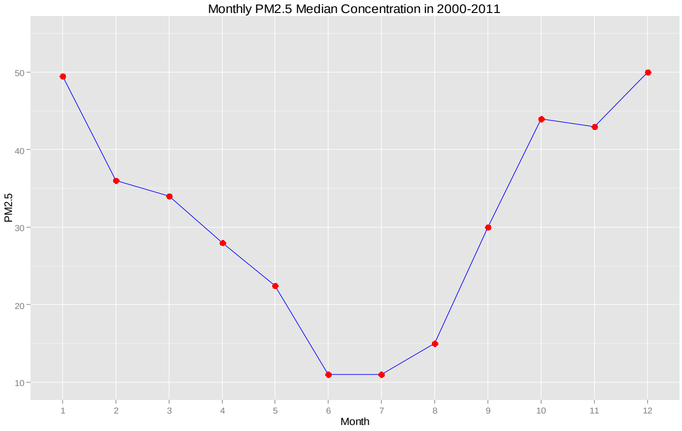

Fig.2. Monthly $PM_{2.5}$ Concentration in 2000-2011

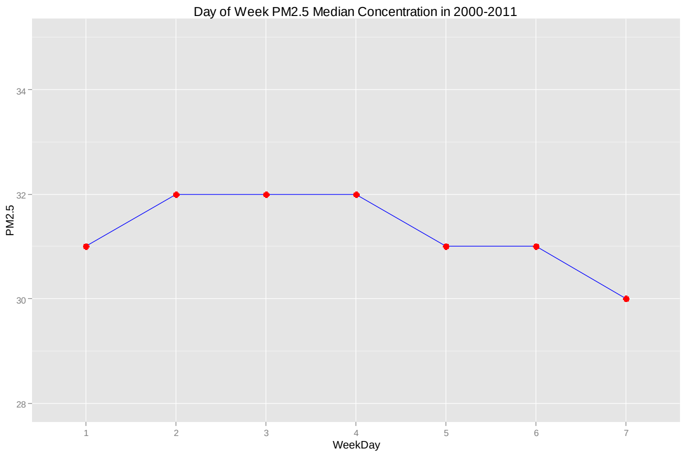

Fig.3. Daily $PM_{2.5}$ Concentration in 2000-2011

TABLE II. VARIABLE LIST

Notation	Description	Variable Class
MP	Mean Pressure	Numeric
AT1	Max Air Temperature	Numeric
AT2	Mean Air Temperature	Numeric
AT3	Min Air Temperature	Numeric
MDPT	Mean Dew Point Temperature	Numeric
RH1	Max Relative Humidity	Numeric
RH2	Mean Relative Humidity	Numeric
RH3	Min Relative Humidity	Numeric
TR	Total Rainfall	Numeric
PWD	Prevailing Wind Direction	Numeric
MWS	Mean Wind Speed	Numeric
PM2.5	$PM_{2.5}$ concentration	Numeric
MONTH	Month	Nominal
WEEK	Day of week	Nominal

IV. EXPERIMENTS

The experiments include three sections: the first and second experiment will test RF and C5.0, respectively. We try to train the model and select the proper parameters in each one. The third one will compare them by using 100 times 10-fold cross validation (10-fold CV) in order to understand which one is more accurate as well as stable.

A. Random Forest (RF)

We use R package "randomForest" to train and test the performance of RF model. There are two important parameters in RF model, that is, the number of splitting feathers (i.e. "*mtry*") and the number of trees (i.e. "*ntree*"). We try to select a proper number in order to obtain the best testing accuracy by 10-fold CV. Firstly, we set "*ntree*" from 1 to 500 and "*mtry*" from 2 to 5. The result is shown in Figure 4. We learn that when the number of splitting feathers is 2 or 3 is somewhat better than other two values as the accuracy of both are tightness. The best accuracy of each splitting number is shown in Table 3. According to this result we choose *mty* = 3 and *ntree* = 98 in the following experiments. Note that the default value in R package is $mtry = \sqrt{p}$ (as we mentioned in Section 2) and *ntree* = 500, generally speaking, we can use "*mtry*" as the default value and select "*ntree*" by using 10-fold CV. Alternatively, one can choose the function *tuneRF* for selecting parameters in RF model and the details can be checked in the help file of randomForest package. Figure 5 shows that the importance of variables in RF model, we can see the most important predictor is the previous $PM_{2.5}$, and then MDPT, MP, and MONTH. The criterion of this list is according to the mean decreasing *Gini* gain of each predictor. Why the variable WEEK is not important in RF model? A reasonable explanation is that WEEK waves slightly on each day, moreover, all the medians of $PM_{2.5}$ concentration are higher than $25\mu g/m^3$ (see Figure 3) which is the boundary between response variable.

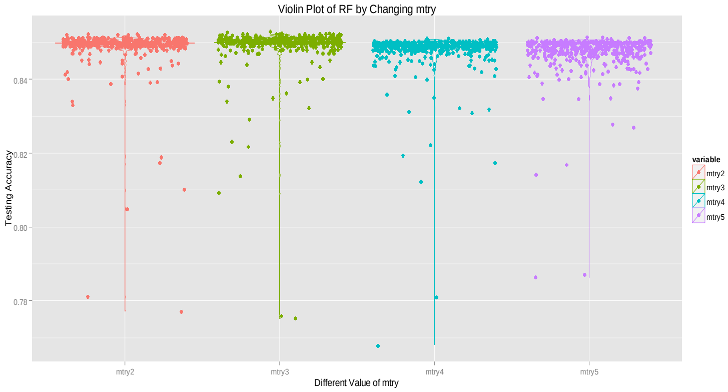

Fig.4. Accuracy on Different Number of Splitting Feathers

TABLE III. RESULT OF RF

mtry	*ntree*	Testing Accuracy
2	95	0.852
3	98	0.853
4	246	0.851
5	349	0.851

B. C5.0

We will use "C50" package for building C5.0 model in this paper. Similar as RF, we try to obtain the best number of trees at first. Note that the maximum number of trees in "C50" package is 100, which is much less than RF (i.e. *ntree* = 500). Some of the results are shown in Table 4. We find that the highest accuracy is at the 32nd tree, whose testing accuracy is 0.852. Figure 6 indicates the trends of accuracy by changing number of trees in RF and C5.0. The training accuracy of both models increases steadily and stays at a stable level at last. The testing accuracy waves little serious at the beginning, especially, C5.0 is much higher than RF when there are only a few trees. But both of them float in a moderate level later, RF is higher than C5.0 at this phase. Figure 7 shows the variables importance of C5.0 model. According to this result, we learn that the most important predictor is the previous $PM_{2.5}$, too. And MONTH, PWD are also important variables, but it is different from the result of RF. C5.0 calculates the percentage of splits associated with each predictor. We think this result is more accurate than RF algorithm, because *Gini* gain will be in favor of those variables having more values and thus offering more splits [16]. But C5.0 uses gain radio which avoids this problem. Variable WEEK is still not important in this model.

TABLE IV. RESULT OF C5.0

Trees	Training	Testing
1	0.868	0.843
2	0.868	0.843
3	0.877	0.842
……	……	……
31	0.943	0.847
32	0.945	0.852
33	0.945	0.849
……	……	……

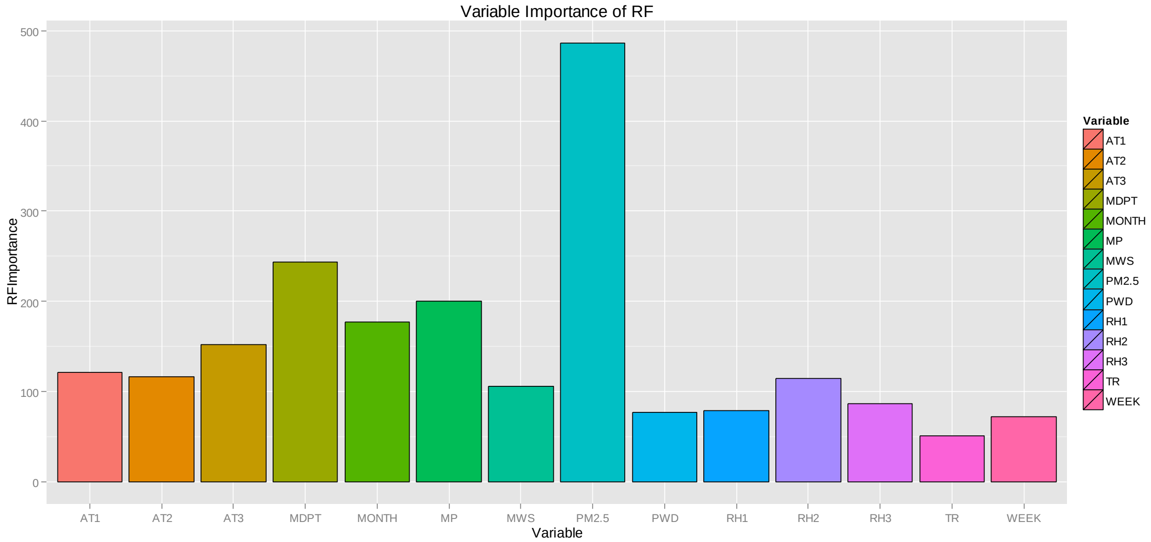

Fig.5. Variable importance of RF

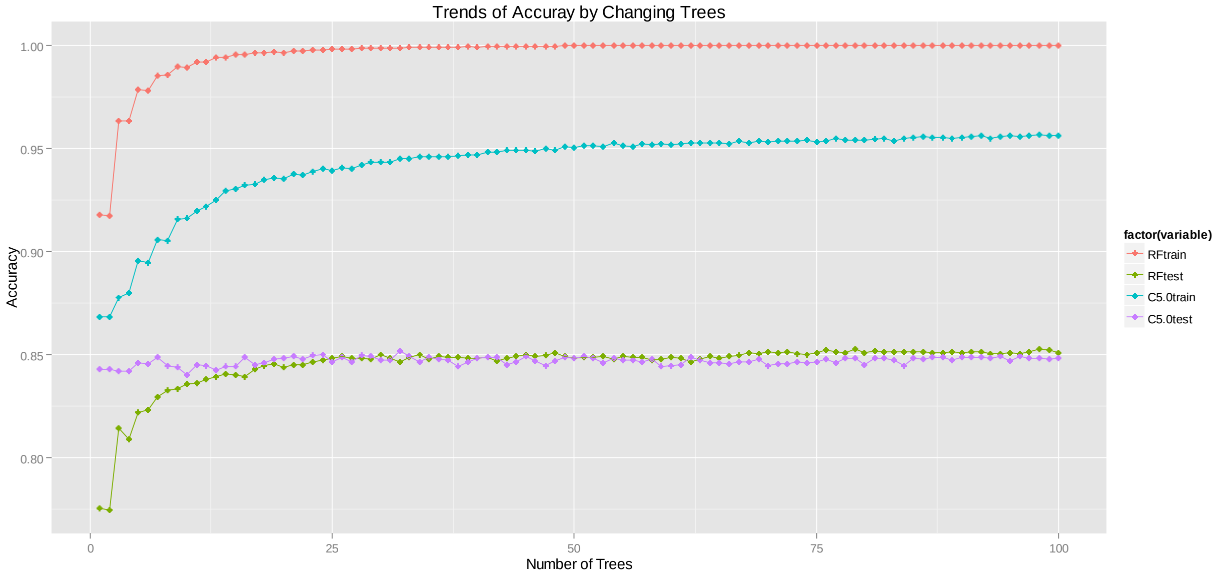

Fig.6. Trends of Testing Accuracy by Changing Number of Trees in RF & C5.0

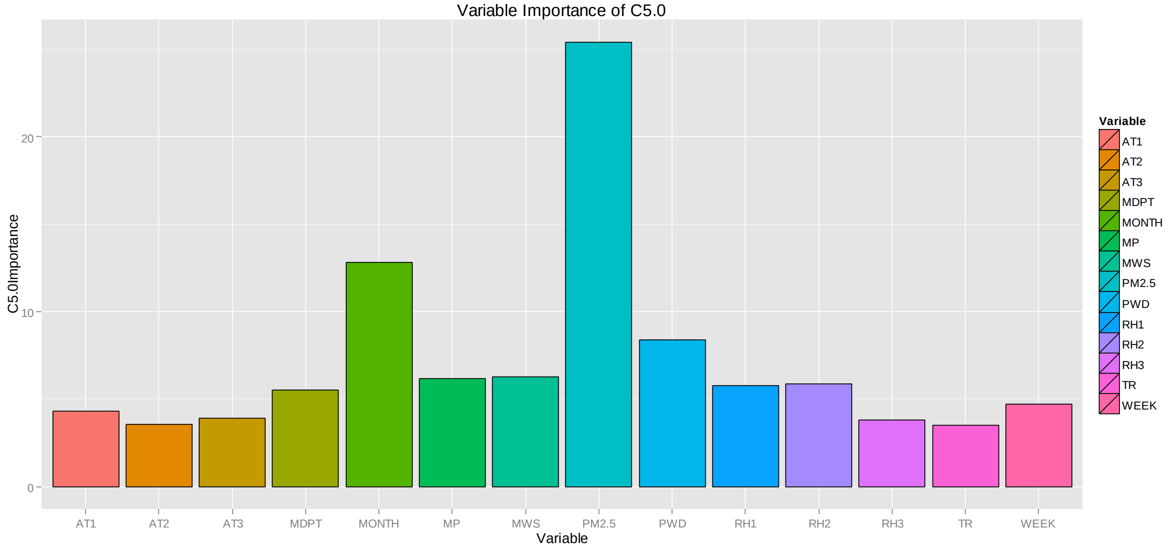

Fig.7. Variable importance of C5.0

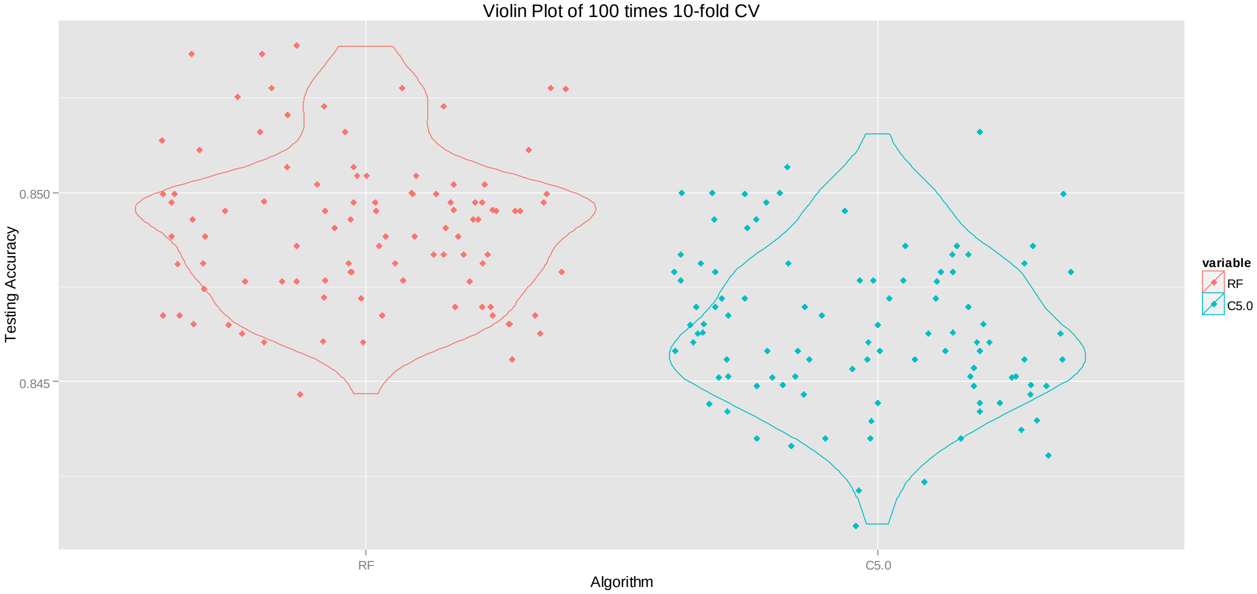

Fig.8. Violin Plot of 100 Times 10-fold CV

C. Comparison

We compare two algorithms by using 100 times 10-fold CV with the result shown in Table 5.

We learn that RF obtains the best result, and its accuracy is around 0.845~0.854. While C5.0 also gets a moderate accuracy, or say, only a little bit worse than RF. Another issue is the stability of these two algorithms during repeated times.

Figure 8 shows the violin plot of 100 times 10-fold CV. We can see that RF is more stable than C5.0 during 100 times and its accuracy is also better than C5.0.

TABLE V. COMPARISON BETWEEN RF&C5.0

	Maximum	**Minimum**	**Median**
RF	0.854	0.845	0.849
C5.0	0.852	0.841	0.846

V. CONCLUSION

In this paper, we build $PM_{2.5}$ concentration levels predictive models by using two popular data mining algorithms, which are RF and C5.0. The dataset, which is from a new town in Hong Kong, includes 4326 rows and 15 columns by deleting all missing values. Based on all experiments, we have our conclusions as below.

1) Selecting the best parameters in each model based on the testing accuracy by using 10-fold CV. For RF model, the number of trees is 98 and the number of splitting feathers is 3. For C5.0 model, the best number of trees is 32. We prefer to use default value of mtry in RF model and select ntree by 10-fold CV.

2) According to 100 times 10-fold CV, the best result is from RF which is around 0.845~0.854. It not only obtains the highest accuracy but also performs more stable than C5.0.

3) Another issue between them is the importance of variables, and we prefer the result of C5.0 as it is unbiased.

4) The advice of using RF or C5.0 in practice is to select the number of iterations at first. 10-fold CV is the selecting method in this paper, while researchers can repeat this process many times, for instance, 10 times 10-fold CV should be better than once. In summary, the selecting process has to maximum limit reducing the random error.

ACKNOWLEDGMENT

The authors wish to thank Hong Kong Environmental Protection Department (HKEPD) and Hong Kong Met-online allowing us to use all the data in this paper.

REFERENCES

[1] Air quality and health, http://www.who.int/mediacentre/factsheets/fs313/en/index.html

[2] WHO Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide (Global update 2005), World Health Organization, whqlibdoc.who.int/hq/2006/WHO_SDE_PHE_OEH_06.02_eng.pdf

[3] Proposed New AQOs for Hong Kong, http://www.epd.gov.hk/epd/english/environmentinhk/air/air_quality_objectives/files/proposed_newAQOs_eng.pdf

[4] Leo Breiman, "Random Forests", Machine Learning, Volume 45, Issue 1, pp. 5-32, 2001.

[5] Leo Breiman, "Statistical Modeling: The Two Cultures", Statistical Science, Volume 16, No. 3, pp. 199-231, 2001.

[6] J. R. Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann, Los Altos, 1993.

[7] C5.0: An Informal Tutorial, www.rulequest.com/see5-unix.html

[8] R-project: http://www.r-project.org/

[9] Andy Liaw, Matthew Wiener, "randomForest" package, Version 4.6.7, http://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[10] Max Kuhn, Steve Weston, Nathan Coulter, "C50" package, Version 0.1.0 14, http://cran.r-project.org/web/packages/C50/C50.pdf

[11] Hadley Wickham, "reshape2" package, Version 1.2.2, http://cran.r-project.org/web/packages/reshape2/reshape2.pdf

[12] Hadley Wickham, "ggplot2" package, Version 0.9.3.1, http://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf

[13] Leo Breiman, "Bagging Predictors", Machine Learning, volume 24 issue 2, pp. 123-140, 1996.

[14] Leo Breiman, "Out-of-bag estimation", http://www.stat.berkeley.edu/~breiman/OOBestimation.pdf

[15] Yoav Freund, Robert Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", European Conference on Computational Learning Theory, pp. 23-37, 1995.

[16] Leo Breiman, Jerome Friedman, Richard Olshen, Charles Stone, "Classification and Regression Trees", Wadsworth Int. Group, pp.42, 1984.