Estimation of PM2.5 concentration from air quality data in Tokyo using machine learning techniques

Graduate School of Humanities and Sciences, Program for Leading Graduate Schools
Ochanomizu University
Megumi KITAGAWA

1 Introduction

Air pollution is one of the most serious global issues because of its adverse effects on human health as well as environmental damage. In such a situation, public interest in the protection from health issues caused by air pollutants has enlarged. To tale any actions not to expand the damages from air pollution, prediction of the future trend in air pollution is needed. Currently, there is a forecast system called VENUS (Visual atmospheric ENvironmental Utility System), provided by National Institute for Environmental Studies, Japan. The information about PM2.5 concentration and O_3 in 3 days ahead is available on their website. They perform this estimation of air pollutants by numerical simulation of physical and chemical processes, which are based on weather forecast data and other environmental data. Motivated by such demand in society, there are more and more studies in machine learning application to construct prediction models.

As a successful example of machine learning models using air quality monitoring data, a PM10 prediction model based on support vector machine with regression showed acceptable performance in Spain [1]. Another investigation in urban air quality proposed a well-performed boosted regression tree model constructed from particle matter concentration data together with traffic data and meteorological data [5]. Particularly, a strategy to apply random forest for building models seems to be effective. Such a model to predict PM2.5 in the U.S. on a national scale was established in [2], where the authors also employed a significant technique in the predictor variable integration. Moreover, random forest algorithm was also adopted for the estimation of PM2.5 in China [3], which is used to obtain reliable historical PM2.5 exposure levels in epideminological studies. Recently, LightGBM approach emerged in [6], whose accuracy was greater than the performance of other machine learning algorithm based models.

In this paper, we developed LightGBM models to predict daily averaged PM2.5 concentration in 1 day ahead. Our target area is the entire Tokyo. We used air quality monitoring data, meteorological data, and social data. We proposed regional models to obtain a more accurate prediction. More concretely, the same number of models as the number of PM2.5 measurement system stations were created. Therefore, the procedure of hyperparameter tuning was carried out for each location independently. This research is aimed at connecting to the practical use of getting predicted future PM2.5 concentrations by adding the latest observed values to update the prediction model day by day.

2 Method

2.1 Data resource

Regarding the air pollution monitoring, the Japanese government set the environmental standard for PM2.5 concentration in the atmosphere in September 2009. The Tokyo Metropolitan

Government has begun to developing PM2.5 automatic measuring machines since 2010, and is currently monitoring atmospheric environment including PM2.5 concentrations in atmospheric environment at all measurement stations. The locations of the measurement stations whose measuring results are used to our study are shown in Figure 1. The hourly monitoring records of PM2.5 concentration, SPM concentrations, temperature, relative humidity, wind speed, and oxidant are downloaded from the Tokyo Metropolitan Government website.

More detailed meteorological observation statistical values such as daily averaged precipitation are available on the Japan Meteorological Agency website. We use the daily averaged values of precipitation, pressure, and the 24 hours sum of the daylight hours. The observation site of pressure is only the place 'Tokyo' among the meteorological observation stations, described in Figure 2.

The population estimate for Tokyo is based on the census population as of October 1, 2015, and is estimated by adding the number of changes in the population of the Basic Resident Register every month. We obtained the monthly population estimate by city and ward from the Tokyo Metropolitan Government website.

In this paper, for all of the air pollution measurements, the meteorological observations, and the population estimate, we focus on the data collected from October in 2013 to September in 2018.

2.2 Feature integration

We have selected the features divided in four types, shown in Table 1. Basically we took the daily average for each hourly observed values. For the precipitation and the daylight hours, we took the sum of each day. We also added a weekly mean of observed PM2.5 concentration. When negative values appeared in the data, then we dealt with them by replacing with 0.

In our scheme of the proper use of the prediction, we needed to arrange the values of PM2.5 concentration on a day, which combined with other information on the previous day as one record. Thus, the prediction would be performed using the data obtained until 1 day before.

Let us give a description of the predictor variables in detail. We have four different types of features.

1. Particle matter feature. The features of this type are based on the observed values of PM2.5 and SPM concentration. We took the weekly average on PM2.5, and the difference between the values in 1 day before and those of 2 days before for PM2.5 and SPM. Addition of the convolutional layer for PM2.5 plays a key role, that is a modern technique introduced in [2]. Its value at a monitoring site i is defined by the formula

$$\frac{\sum_{j \neq i} \frac{1}{d_{ij}^2} x_j}{\sum_{j \neq i} \frac{1}{d_{ij}^2}},$$

   where x_j is the observed PM2.5 concentration value and d_{ij} is the distance from the monitoring site i to another monitoring site j.

2. Meteorological feature. This feature type includes the meteorological conditions such as the temperature, the relative humidity, the oxidant, and the wind speed observed at the PM2.5 measurement stations in Figure 1 as well as the daily precipitation, pressure, and the daylight hours obtained at the weather observation sites in Figure 2.

3. Temporal feature. We used the calendar month parameter to characterize the temporal information.

4. Social feature. The monthly population estimate by city and ward were involved.

Type	Feature	Statement	unit
Particle matter feature	PM25_lag1	daily PM2.5 concentration of 1 day before	$\mu g/m^3$
	SPM_lag1	daily SPM concentration of 1 day before	$\mu g/m^3$
	PM25_conv_lag1	the convolutional layer for PM2.5 of 1 day before	$\mu g/m^3$
	PM_avg	the weekly average of PM2.5 concentration	$\mu g/m^3$
	diff_PM25	difference between daily PM2.5 concentration of 1 day before and one of 2 days before	$\mu g/m^3$
	diff_SPM	difference between daily SPM concentration of 1 day before and one of 2 days before	$\mu g/m^3$
	diff_PM25_conv	difference between the convolutional layer for PM2.5 of 1 day before and one of 2 days before	$\mu g/m^3$
Meteorological feature	Temperature_lag1	daily average of temperature of 1 day before	0.1 °C
	Humidity_lag1	daily average of relative humidity of 1 day before	0.1 %
	Ox_lag1	daily average of oxidant of 1 day before	ppb
	Precipitation_lag1	daily sum of precipitation of 1 day before	mm
	Pressure_lag1	daily average pressure of 1 day before	hPa
	Sunshine_lag1	the daylight hours of 1 day before	hours
	wind_lag1	daily average wind speed of 1 day before	0.1 m/s
	diff_Ox	difference between daily average oxidant of 1 day before and one of 2 days before	ppb
	diff_Precipitation	difference between daily precipitation of 1 day before and one of 2 days before	mm
	diff_Press	difference between daily pressure of 1 day before and one of 2 days before	hPa
	diff_Sunshine	difference between the daylight hours of 1 day before and one of 2 days before	hours
	diff_wind	difference between daily wind speed of 1 day before and one of 2 days before	0.1 m/s
Temporal feature	Month	calendar month	
Social feature	Population	the monthly population estimate by city and ward	

Table 1: Feature list

2.3 Solution statement

In this paper, we chose LightGBM for our prediction model, which is a machine learning framework of gradient boosting based on decision tree. The technique of gradient boosting has been highly attractive in the field of competitive data science. The basis of LightGBM is as follows.

- Decision Tree. It is a popular method of the supervised learning, which grows branches of dataset by a criterion determined by the features.
- Ensemble Leaning. Ensemble method builds one training model by a combination of multiple models.
- Gradient Boosting. Boosting method reflects the output from one step previous weak learner in the next training dataset. That is, it improves the ensemble performance by developing an weak learner to train the training sample which is the residuals of another weak learner. In each iteration of gradient boosting, the gradients are applied to evaluate the loss function with respect to the output of the model.

In the training process of gradient boosting, there are two methods for handling decision tree as follows.

- Level-wise: The tree grows by levels, which is traditionally used. We give a description in Figure 3.
- Leaf-wise: The tree grows by leaves as shown in Figure 4. It has a tendency to reduce the computational cost in the training process. Since it constructs more complicated trees, it raises the model accuracy and at the same time it tends to lead to overfitting.

LightGBM applies the leaf-wise method in training process of boosting so that it efficiently decreases the calculation cost.

One of the crucial techniques of LightGBM is the histogram-based algorithm in the process of the training of decision tree. This algorithm constructs discrete bins from continuous feature values. Then it uses the bins to create the feature histograms, so that it can effectively find best split points derived from the feature histograms. These procedures are beneficial to realize reducing the training costs including the memory consumption and computational cost on the split point determination. Particularly, in order to make the cost on the step for the histogram building smaller, it is necessary to reduce the size of dataset and the number of features. LightGBM is outstanding in the reducing since it is equipped with the successful strategies for dataset subsampling and weak features filtering [4].

2.4 Approach

Our aim is to predict the daily averaged PM2.5 concentration in 1 day ahead for each location. Therefore, we suppose to get the predicted values by a prediction model associated with the data gathered until the previous day. Moreover, in order to obtain more accurate prediction, we have created the models for each PM2.5 monitoring site.

We applied grid search technique for hyperparameter tuning. In particular, we considered the following hyperparameters of LightGBM.

1. n_estimators. The number of boosting iterations.
2. max_depth. The maximum value of the tree depth. Generally, this parameter should be restricted to tackle overfitting. It also should be adjusted by keeping a balance between max_depth and other hyperparameters to acquire more accurate model.
3. num_leaves. The maximum number of leaves in one tree. Since it controls the complexity of decision tree, it may largely affects on the model performance. Too large value causes overfitting and too small value causes underfitting. It is better to change together with max_depth.
4. learning_rate. The shrinkage rate. It controls the model not to lead to overfitting.

All statistical computing was done in Python, version 3.7.3, using scikit-learn (version 0.21.0) and lightgbm (version 2.3.0) packages.

2.5 Training and test dataset

We made a partition of training and test datasets by a specific date. More concretely, the period for the training dataset is from 2013/10/08 to 2017/09/30, while the period for the test dataset is from 2017/10/01 to 2018/09/30. In other words, the training data contains the records for 5 years and the test data contains for 1 year.

2.6 Evaluation function

The model performance was evaluated using two different functions such as Root Mean Squared Error (RMSE) and Coefficient of determination (R^2). For the predicted values p_i, RMSE and R^2 are defined by

$$\mathrm{RMSE} = \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - p_i)^2 \right)^{\frac{1}{2}}, \quad \mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - p_i)^2}{\sum_{i=1}^{n} (y_i - \mu_i)^2},$$

where y_i is an actual value, μ_i is mean of actual values, and n is the number of observations.

3 Result

The model performance for each training and test dataset is printed in Table 2. The column of 'Range' and 'Average' for all sites displays minimum, maximum values of the performance and the average for all locations respectively. For 'Overall', we gathered all predicted values from each location and then calculate the evaluation indicators with all observed values. We achieved that RMSE = 2.927 and R^2 = 0.840 on test dataset. Considering our challenging points, although it is natural to vary the scores in different locations, we have some unsatisfactory models whose RMSE and R^2 are maximum and minimum among all monitoring sites respectively. Comparing RMSE scores among all locations, the maximum value 2.503 of RMSE appeared Harumi Chuo-ku on training data, while on test data Yashio Shinagawa-ku model took the maximum value 3.559 of RMSE. Its difference is about 1.0, which is almost the same as the difference of average RMSE between training data and test data. On the other hand, since R^2 is an evaluation index which means better if the value is close to close to 1, the models with minimum R^2 should be improved. The largest value 0.914 of R^2 occurred at Takanawa Minato-ku in training data, however, the minimum in test data is 0.724 at Higashiome Ome-shi. These results give more than 0.18 of difference, so the difficulty is how to deal with this issue to diminish such a gap between training data and test data, that is, how to build models to fit not only to training data but also test data nearly equally.

Model	Performance statistics	Range	Average for all sites	Overall
	RMSE on train	1.303 – 2.503	1.914	1.949
	R^2 on train	0.914 – 0.966	0.935	0.936
LightGBM	RMSE on test	2.534 – 3.558	2.922	2.927
	R^2 on test	0.724 – 0.901	0.830	0.840

Table 2: Performance of the ML models

We provide a scatter plot of actual values and predicted values for daily averaged PM2.5 concentrations in Figure 5. We can see that the predicted value has a positive linear relationship with the actual value regardless of the training or test dataset. For test dataset, the model is likely to estimate lower than actual values, especially in the range of larger values. On the other hand, for training dataset, larger estimation than actual values occurs around 0 of actual values.

We show the ranking of the feature importance calculated by taking mean for all monitoring sites in Figure 6. The features generated by PM2.5 concentrations such as the values in 1 day before, the weekly average, and the convolutional layer for PM2.5 of 1 day before have

significantly large contributions to our model. The first 10 features contain four PM2.5 related features and five different kinds of meteorological features. This fact verifies the advantage of using features of particle matter type and meteorological type.

Acknowledgements

We would like to thank Yuka Sato and Sayaka Takayama for their support and discussion. The author is particularly grateful for the assistance given by Takuya Okazaki. We would like to express the deepest appreciation to Professor Fadoua Ghourabi and Professor Xavier Dahan for giving us constructive comments and warm encouragement. We would also like to thank Program for Leading Graduate Schools, Ochanomizu University for their financial support.

References

[1] P. J. García Nieto, F. Sánchez Lasheras, E. García-Gonzalo, and F. J. de Cos Juez. Estimation of pm10 concentration from air quality data in the vicinity of a major steelworks site in the metropolitan area of avilés (northern spain) using machine learning techniques. *Stochastic Environmental Research and Risk Assessment*, 32(11):3287–3298, Nov 2018.

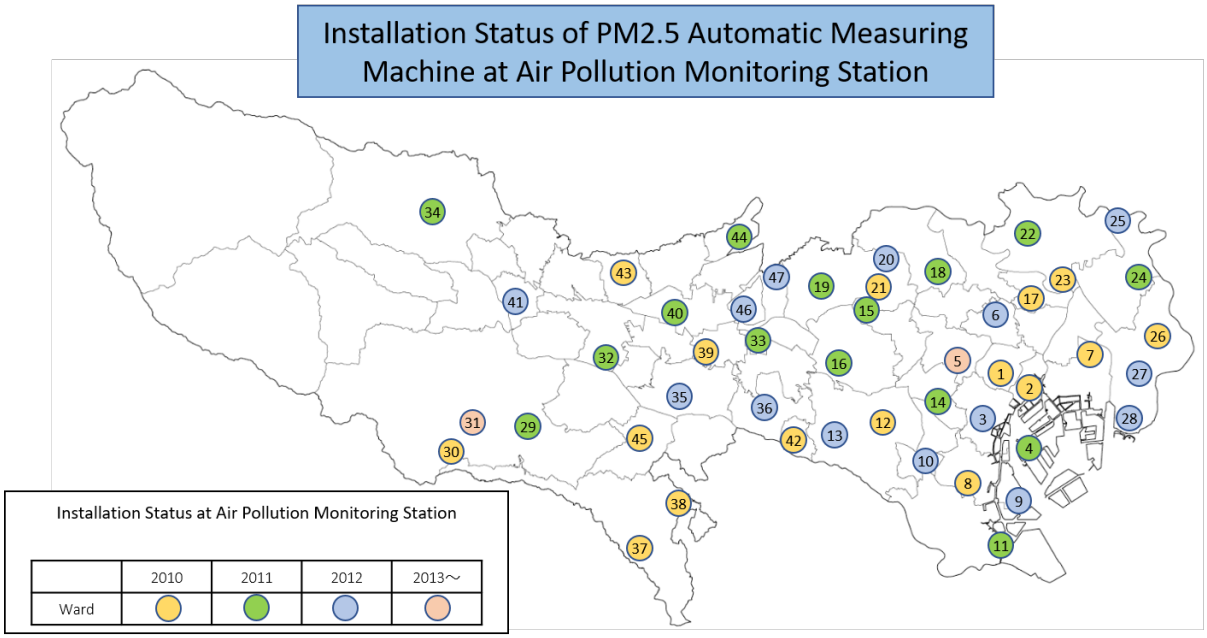
[2] Xuefei Hu, Jessica H. Belle, Xia Meng, Avani Wildani, Lance A. Waller, Matthew J. Strickland, and Yang Liu. Estimating pm2.5 concentrations in the conterminous united states using the random forest approach. *Environmental Science & Technology*, 51(12):6936–6944, 06 2017.

[3] Keyong Huang, Qingyang Xiao, Xia Meng, Guannan Geng, Yujie Wang, Alexei Lyapustin, Dongfeng Gu, and Yang Liu. Predicting monthly high-resolution pm2.5 concentrations with random forest model in the north china plain. *Environmental Pollution*, 242:675 – 683, 2018.

[4] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017.

[5] A. Suleiman, M.R. Tight, and A.D. Quinn. Applying machine learning methods in managing urban concentrations of traffic-related particulate matter (pm10 and pm2.5). *Atmospheric Pollution Research*, 10(1):134 – 144, 2019.

[6] Y. Zhang, Y. Wang, M. Gao, Q. Ma, J. Zhao, R. Zhang, Q. Wang, and L. Huang. A predictive data feature exploration-based air quality prediction approach. *IEEE Access*, 7:30732–30743, 2019.



Measuring Station	Measuring Station	Measuring Station	Measuring Station	Measuring Station
(1) Kandatsukasamachi, Chiyoda-ku	(11) Higashikoujiya, Ota-ku	(21) Nerima, Nerima-ku	(31) Dairakujimachi, Hachioji	(41) Honcho, Fussa
(2) Harumi, Chuo-ku	(12) Setagaya, Setagaya-ku	(22) Nishiarai, Adachi-ku	(32) Izumimachi, Tachikawa	(42) Nakaizumi, Komae
(3) Takanawa, Minato-ku	(13) Seijo, Setagaya-ku	(23) Ayase, Adachi-ku	(33) Sekimae, Musashino	(43) Narahashi, Higashiyamato
(4) Daiba, Minato-ku	(14) Utagawacho, Shibuya-ku	(24) Kamakura, Katsushika-ku	(34) Higashioume, Ome	(44) Kamikiyoto, Kiyose
(5) Naitomachi, Shinjuku-ku	(15) Wakamiya, Nakano-ku	(25) Mizumotokouen, Katsushika-ku	(35) Miyanishicho, Fuchu	(45) Atago, Tama
(6) Honkomagome, Bunkyo-ku	(16) Kugayama, Suginami-ku	(26) Shishibone, Edogawa-ku	(36) Jindaijiminamimachi, Chofu	(46) Tanashicho, Nishitokyo
(7) Oshima, Koto-ku	(17) Minamisenju, Suginami-ku	(27) Haruecho, Edogawa-ku	(37) Nakamachi, Machida	(47) Shimohoya, Nishitokyo
(8) Toyomachi, Shinagawa-ku	(18) Honmachi, Itabashi-ku	(28) Minamikasai, Edogawa-ku	(38) Nougayamachi, Machida	
(9) Yashio, Shinagawa-ku	(19) Shakujiimachi, Nerima-ku	(29) Katakuracho, Hachioji	(39) Honmachi, Koganei	
(10) Himonya, Meguro-ku	(20) Kitamachi, Nerima-ku	(30) Tatemachi, Hachioji	(40) Ogawamachi, Kodaira	

Figure 1: Atmospheric environment measurement stations in Tokyo.

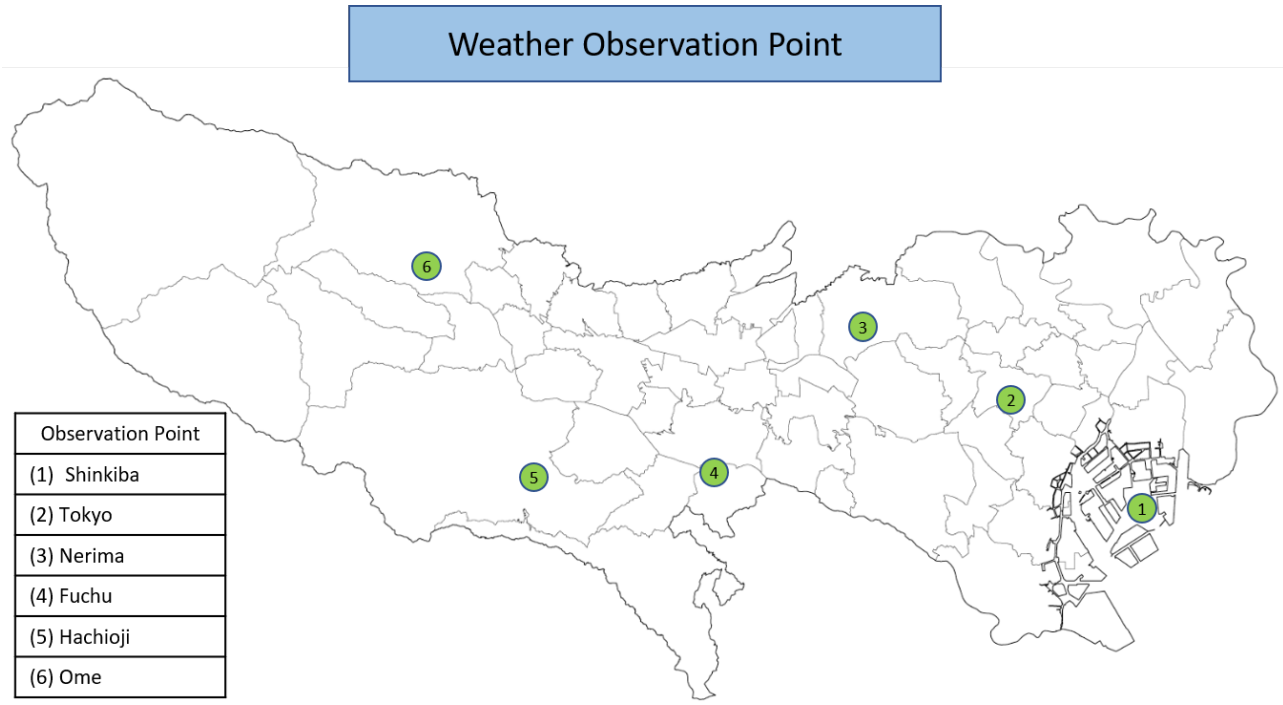


Figure 2: Weather observation sites in Tokyo.

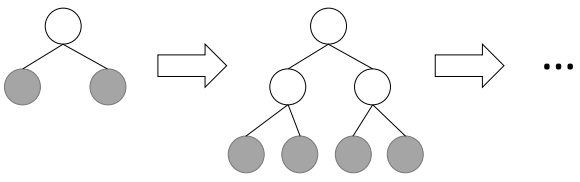


Figure 3: Level-wise decision tree training.

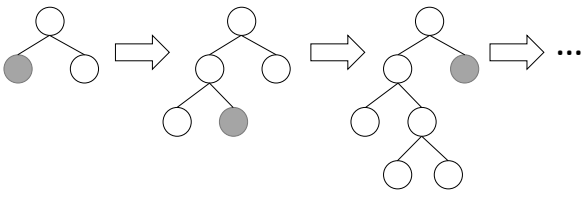


Figure 4: Leaf-wise decision tree training.

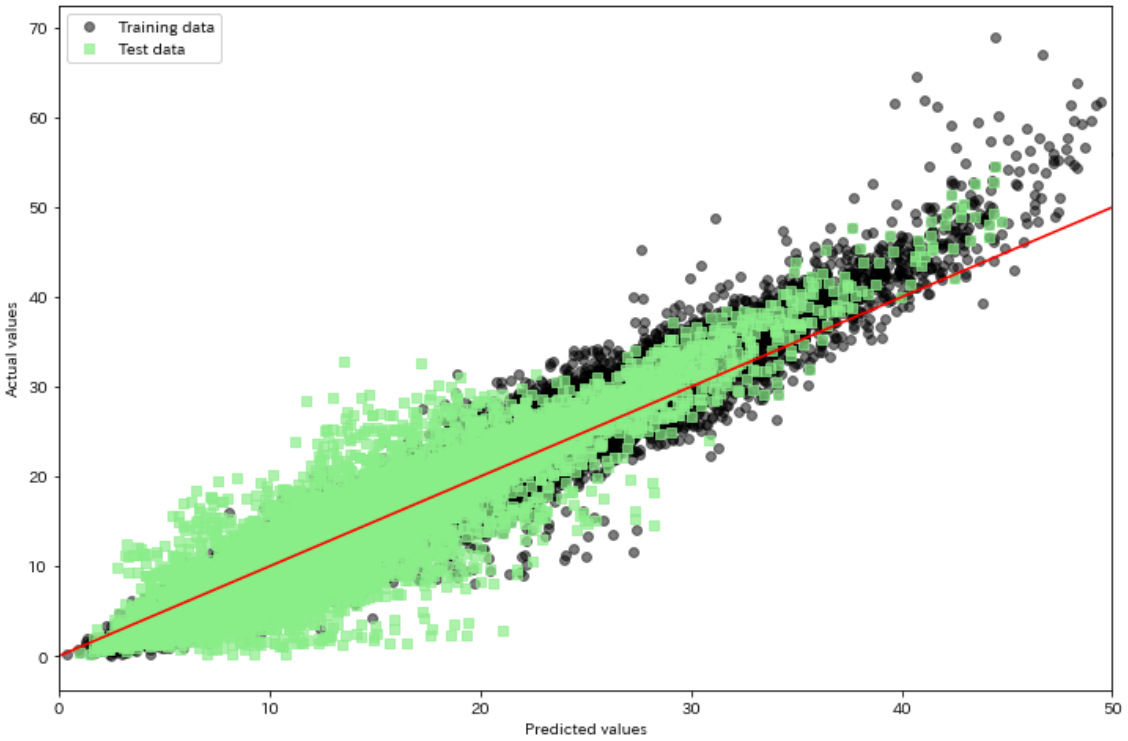


Figure 5: Scatter plot of actual values and predicted values.

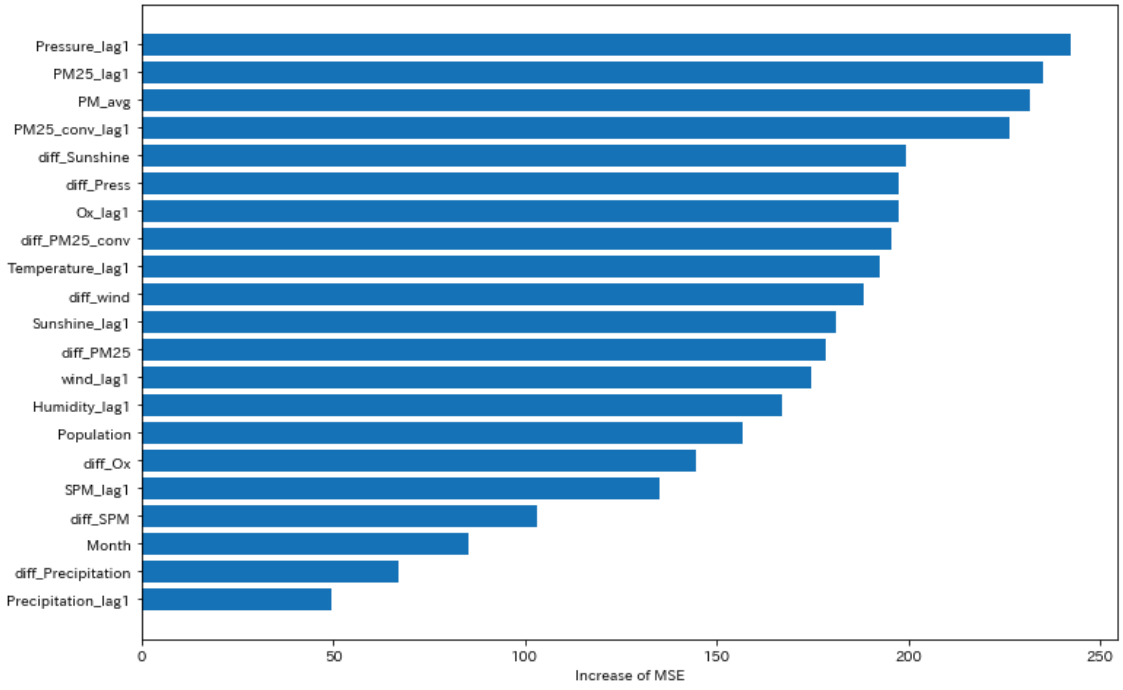


Figure 6: Feature importance.