Prediction of PM2.5 Concentration

Shuhui Guo

Contents

1 Introduction **2**

2 Data **2**

2.1 Data Source 2

2.2 Data Preparation and Description 2

3 Modeling **3**

3.1 Lasso Regression 3

3.2 Random Forest 5

3.3 LSTM Neural Network 6

4 Model Comparison and Conclusion **7**

References **7**

1 Introduction

Air quality and air pollution are getting more attentions nowadays. The particular matter, PM2.5, is one of the main pollutants. Studies show that exposure to high PM2.5 can cause various physical diseases. Therefore, it is important to forecast the PM2.5 concentration and take actions to prevent severe air pollution problems.

In this study, three machine learning algorithms on the PM2.5 dataset are implemented to give more accurate prediction results for the PM2.5 concentration. Hopefully this study will give some suggestions to this field.

2 Data

2.1 Data Source

Data analysis in this report is based on the Beijing PM2.5 dataset on UCI Machine Learning Repository. The raw dataset contains 43824 observations and 12 features from January 2010 to December 2014. The goal is to predict the PM2.5 concentration.

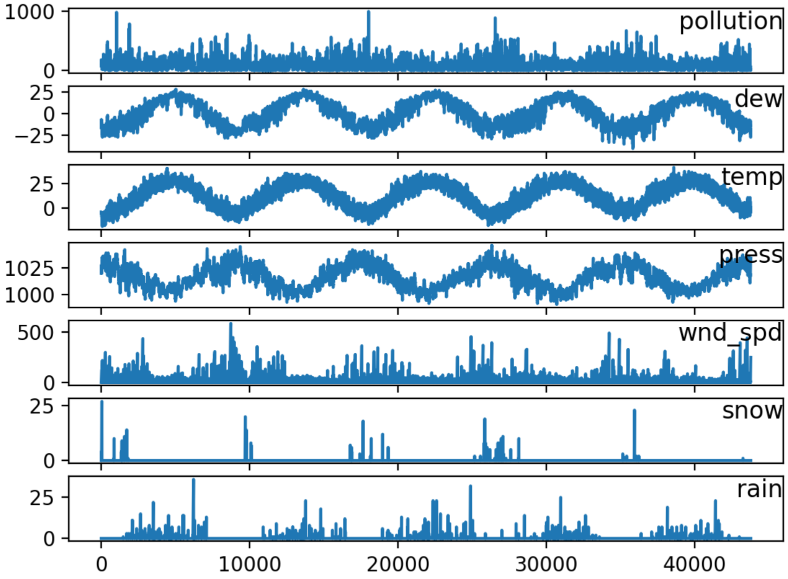
2.2 Data Preparation and Description

There are 2067 missing values in **pollution**. After removing the missing values, the total number of observations is 41757. For the features, there are 10 numerical variables and 1 categorical variable. For model fitting and realistic prediction, the variables **year**, **month**, **day**, **hour** should not be put in models, but can be seen as time labels. Finally, the dataset used for analysis includes 41757 observations, 7 input variables and 1 target variable.

The target variable is **pollution**. The input variables are shown as below:

	variable	data type	explanation
1	dew	numerical	Dew Point
2	temp	numerical	Temperature
3	press	numerical	Pressure
4	wnd dir	categorical	Combined wind direction
5	wnd spd	numerical	Combined wind speed
6	snow	numerical	Cumulated hours of snow
7	rain	numerical	Cumulated hours of rain

To get an understanding of the data, the trends of the numerical variables are visualized as below.



Based on this plot, we can see the time series characteristics and seasonal trends are obvious for each variable.

3 Modeling

To build models, I first divide data into training and testing datasets. Since the variables have time series characteristics, randomly splitting and shuffling the dataset is not a good idea. To maintain the trends, the data recorded from January 2010 to December 2013 are divided into training dataset and the data recorded from January 2014 to December 2014 are divided into testing dataset.

Then I will try Lasso Regression, Random Forest and LSTM Neural Network to do prediction.

3.1 Lasso Regression

The Lasso Regression is the regression model with lasso penalty. The lasso penalty could be written as:

$$\hat{\beta} = \text{arg min}_{\beta}\frac{1}{2}||y - X\beta||^2 + \lambda||\beta||_1$$

where λ is the penalty level which could be selected by cross-validation. By this model, the varaibles used for prediction are selected with the less colinearity issue and the prediction can be more precise.

Since the data has time series characteristics, the general cross-validation method cannot be properly used because the variables have autocorrelations by time and the natural orders of data are important. Thus, the 10-fold cross-validation method applied in this study is:

Step 1: Split the training dataset to 10 consecutive time folds.

Step 2: Train on fold 1 and test on fold 2.

Step 3: Train on fold 1, 2 and test on fold 3.

Step 4: Train on fold 1, 2, 3 and test on fold 4.

Step 5: Train on fold 1, 2, 3, 4 and test on fold 5.

Step 6: Train on fold 1, 2, 3, 4, 5 and test on fold 6.

Step 7: Train on fold 1, 2, 3, 4, 5, 6 and test on fold 7.

Step 8: Train on fold 1, 2, 3, 4, 5, 6, 7 and test on fold 8.

Step 9: Train on fold 1, 2, 3, 4, 5, 6, 7, 8 and test on fold 9.

Step 10: Train on fold 1, 2, 3, 4, 5, 6, 7, 8, 9 and test on fold 10.

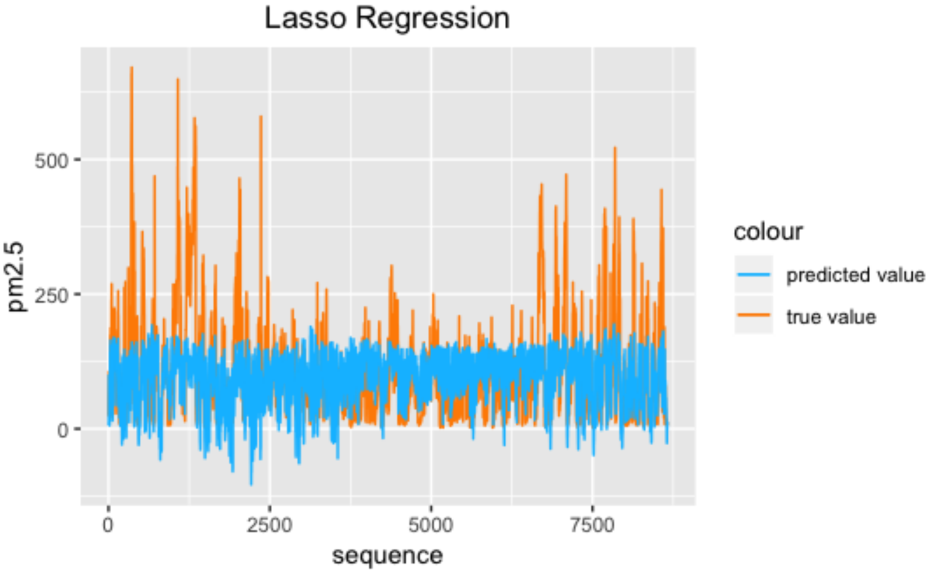
Step 11: Calculate the average of the mean squared error of the 9 test folds.

After the above procedure, the λ giving the lowest error is selected as 0.05. Then the model with this λ is fitted. The selected variables and corresponding coeffitients are as below:

variable	coefficient	variable	coefficient
Intercept	1951.065	dew	4.151
temp	-6.227	press	-1.746
wnd dir-cv	8.064	wnd dir-NE	-18.386
wnd dir-NW	-21.254	wnd dir-SE	11.586
wnd spd	-0.187	snow	-2.793
rain	-6.067		

Take the variables **temp** and **snow** as example. For every one unit increase in **temp** (temperature), the average of pm2.5 concentration decreases 6.227. This might be because when the temperature increases, people will use less heat and gas, which releases a mass of pollutants to the air. For every one unit increase in **snow** (cumulated hours of snow), the average of pm2.5 concentration decreases 2.793. This might be because the precipitation can reduce the pollutants floating in the air.

Using this model to do the prediction for the testing dataset, the comparison of testing data and prediction values is plotted as below:

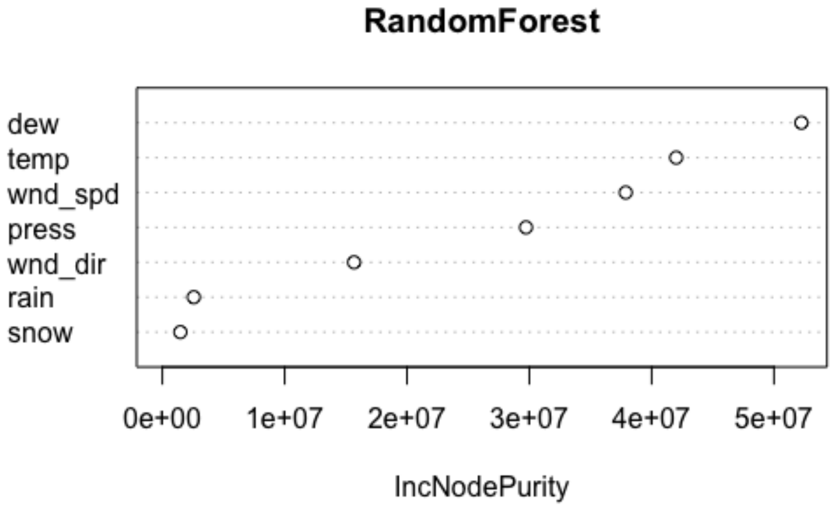


In this plot, the trend of testing data is fitted by prediction values, but not very well. The root-mean-squared error is 82.136.

3.2 Random Forest

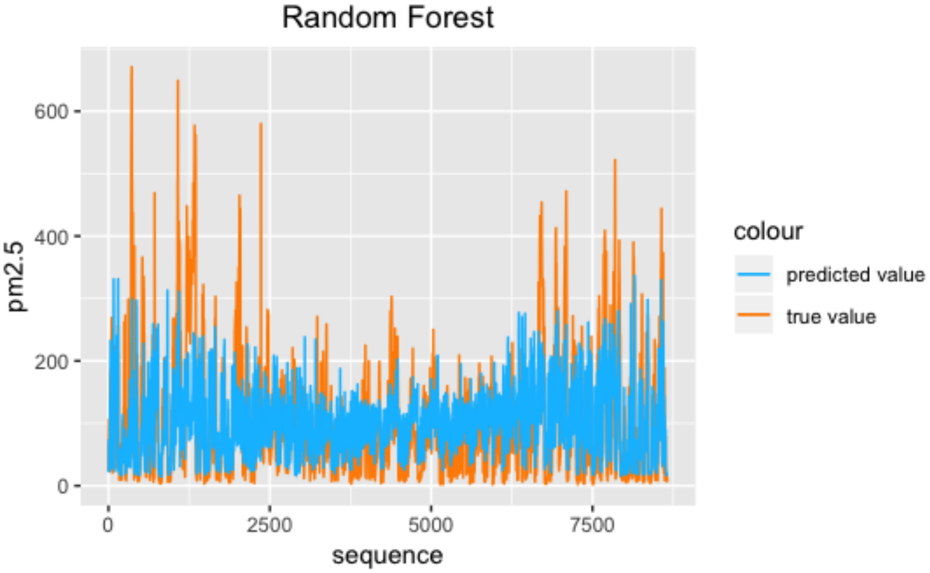
Random Forest is a flexible model to handle both categorical and numerical variables in a natural way. Also, it is robust to outliers. So in this part, Random Forest will be used to predict the pm2.5 concentration.

After trying different parameters, the parameters giving the lowest prediction error are selected to be $ntree = 500$, $mtry = p/3 = 7/3$, $nodesize = 5$. The variable importance can be shown as below:



Based on this plot, **dew** and **temp** are the most important predictors in this model. It matches the real condition because the pm2.5 concentration is mostly effected by the dew point temperature and the sensible temperature. **wnd spd**, **press**, and **wnd dir** are also important. The wind and pressure are related to the locations pm2.5 pollutants float so that they are important for the concentration. **rain** and **snow** are the least important. It might because precipitations are temporary features which affect pm2.5 concentration.

Using this model to do the prediction for the testing dataset, the comparison of testing data and prediction values is plotted as below:



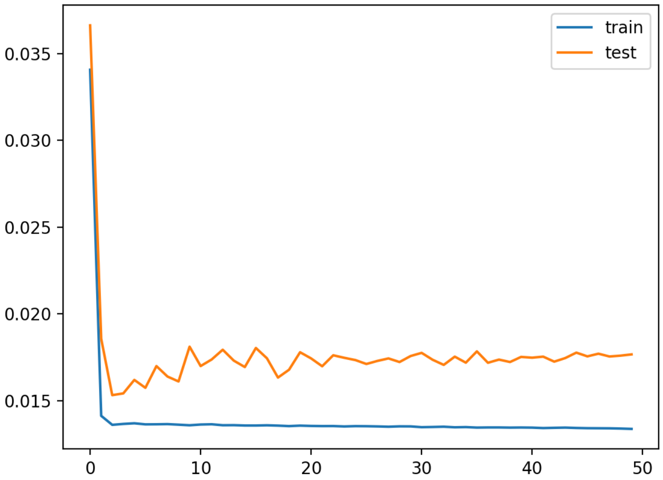
In this plot, the trend of testing data is well fitted by prediction values. The root-mean-squared error is 76.953.

3.3 LSTM Neural Network

The Long Short-Term Memory (LSTM) recurrent neural network can be used to maintain the temporal information and process the data as an actual sequence, which is proper for the time series data. In this part, an LSTM model is developed in the Keras deep learning library to predict the pm2.5 concentration.

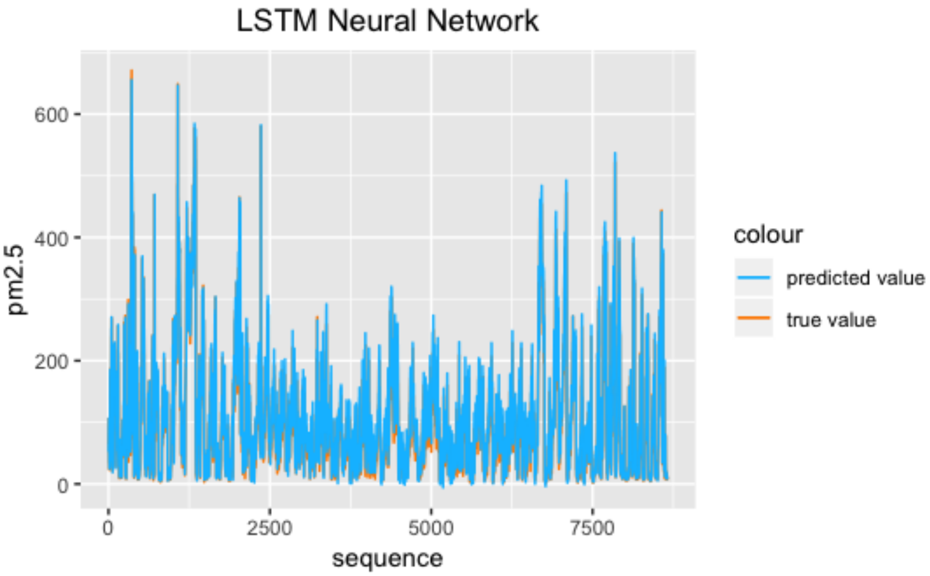
Before building this model, all features are normalized. Then the time series dataset is framed as a supervised learning dataset. After this conversion, the data with $t-1$ is set as X, and the data with t is set as y.

The LSTM neural network has a hidden layer and an output layer. There are 50 units in the hidden layer. The model is trained with ADAM optimizer, and fit for 50 epochs with a batch size of 72. The Mean Absolute Error (MAE) loss function is applied in this model and the train and test loss after each epoch are plotted as below:



Based on the above plot, we can see the testing loss becomes stable after 20 epochs.

Using this model to do the prediction for the testing dataset, the comparison of testing data and prediction values is plotted as below:



In this plot, the trend of testing data is fitted by prediction values very well. The root-mean-squared error is 25.254.

4 Model Comparison and Conclusion

So far, three models have been fitted to make predictions. The performance measurements for the three models is shown as below:

measurement	Lasso Regression	Random Forest	LSTM Neural Network
RMSE	82.136	76.953	25.254

Based on these results, LSTM Neural Network has the lowest RMSE, which indicates LSTM Neural Network is the best model for prediction. Nevertheless, LSTM Neural Network cannot reveal the relationships between predictors and the response. Therefore, if the correlations between predictors and response need to be explored, Lasso Regression and Random Forest can be applied.

References

Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H. and Chen, S. X. (2015). Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating. Proceedings of the Royal Society A, 471, 20150257.

Pochetti, F. (2014, September 16). Pythonic Cross Validation on Time Series. Retrieved from http://francescopochetti.com/pythonic-cross-validation-time-series-pandas-scikit-learn/

Brownlee, J. (2017, May 8). How to Convert a Time Series to a Supervised Learning Problem in Python. Retrieved from https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/

Kennedy, G. (n.d.). Beijing Air Quality: Statistical analysis using R. Retrieved from http://www.garethkennedy.net/AQIStats.html