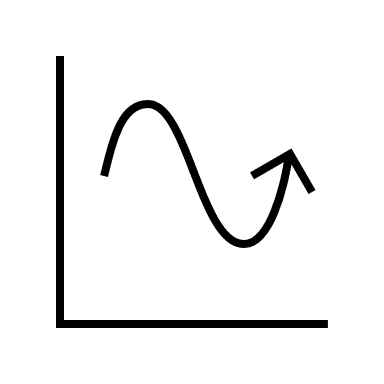
******Prediction Models for PM2.5 Using Time and Associated Weather Conditions in Azure ML Studio and Using the ARIMA Time Series Model**

**Lorena Cannon ͣ\***

**ͣ**Departments of Geography and Anthropology, University of Colorado, Denver

**\*** [lorena.cannon@ucdenver.edu](mailto:lorena.cannon@ucdenver.edu)

Received: May 14, 2020

**Abstract** Using a large dataset from UCI ML Repository of 5 years of hourly PM2.5 concentrations in Beijing with associated time and meteorological measurements, this analysis performed and evaluated a univariate ARIMA time series prediction model and various multivariate regression models in Azure ML Studio to predict future PM2.5 measurements. Results indicate that a multivariate decision forest regression model is the most accurate, with the ARIMA model very close behind in predicting future PM2.5 levels. Discussion regarding the usefulness in collecting and considering different time and associated weather data trends is compared with the different univariate and multivariate models.

**Keywords: PM2.5, particulate matter 2.5, big data, Azure ML Studio, machine learning, ARIMA model, python**

1. **Introduction and Background**

Worldwide, since the industrial revolution, pollution levels have increased in many urban areas. A standard measure of pollution and dangerous air qualities is the particulate matter 2.5 (PM2.5), which is a concentration measurement of suspended particles in the air measuring 2.5 micrometers or less in diameter. High PM2.5 levels are linked with many adverse health conditions, such as chronic respiratory diseases, heart disease, lung disease, and birth defects. The ability to predict future levels of PM2.5 could be helpful for designating times or days that would be particularly dangerous for vulnerable persons to go outside. It could also be helpful for designing future policy changes which aim to reduce PM2.5 levels, as patterns or correlations with other factors could contribute to those reductions. The goal of this analysis is to use a big dataset to explore relationships, design and perform various predictive models for PM2.5 levels using associated weather conditions and/or the time of collection, and to compare the models’ effectiveness.

The dataset is called “Beijing PM2.5 Data” and is downloaded from the University of California Irvine Machine Learning Repository, located at <https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data>. It was donated by Song Xi Chen, csx@gsm.pku.edu.cn, from the Guanghua School of Management, Center for Statistical Science at Peking University on January 19, 2017. This dataset is comprised of hourly particulate matter 2.5 (PM2.5) measurements from the US Embassy in Beijing, along with associated meteorological data from the Beijing Capital International Airport, from Jan 1st, 2010 to Dec 31st, 2014. The distance between the embassy and the airport is about 13 miles, according to Google Maps.

The dataset has a total of 43,824 observations, each with 13 attributes: no. (row number), year, month, day, hour, pm2.5 (PM2.5 concentration (ug/m^3)), DEWP (Dew Point (â„ƒ)), TEMP (Temperature (â„ƒ)), PRES (Pressure (hPa)), cbwd (Combined wind direction), Iws (Cumulated wind speed (m/s)), Is (Cumulated hours of snow), and Ir (Cumulated hours of rain). All data is reported in integers, except for combined wind direction which is a string (NW, SE, etc.), and missing data is entered as ‘NaN.’ This dataset is multivariate and a time-series.

1. **Methods and Analysis**

I used a python script in a Jupyter Notebook on the Microsoft Azure Notebooks platform to clean the data, calculate data statistics, and visualize and explore the data. The original csv table was uploaded into my project folder for this analysis phase. All codes and results from this analysis phase are compiled in a supplementary ipynb file attached to this report. ARIMA Modeling and testing was also performed in this Jupyter Notebook. For multivariate predictive modeling and testing, I used the Azure Machine Learning Studio. I uploaded a cleaned dataset csv table onto this platform for this phase of modeling and testing.

**2.1 Preprocessing and Statistics**

In order to run quantitative analyses on the data I first cleaned the dataset to remove all rows containing missing PM2.5 measurements. This reduced the dataset size by 2,067 observations, about 4.2% of the original data, yielding a total of 41,757 valid observations.

The first predictive model, the ARIMA time series, is a univariate model. I created a univariate dataset from the cleaned dataset to only include the PM2.5 column. I calculated descriptive statistics on this column using a python script. The mean PM2.5 value is 98.613215 ug/m^3, with a standard deviation of 92.050387 ug/m^3. The minimum value is 0, the maximum value is 994, and the 25th percentile, 50th percentile and 75th percentile are 29, 72 and 137, respectively. There are 581 unique PM2.5 values. See *Figure 1* for a boxplot and bar chart of the PM2.5 column, visualized using Azure ML Studio. Notice the highly skewed distribution.

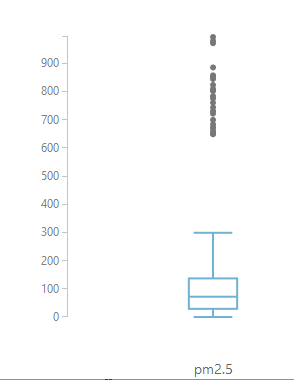
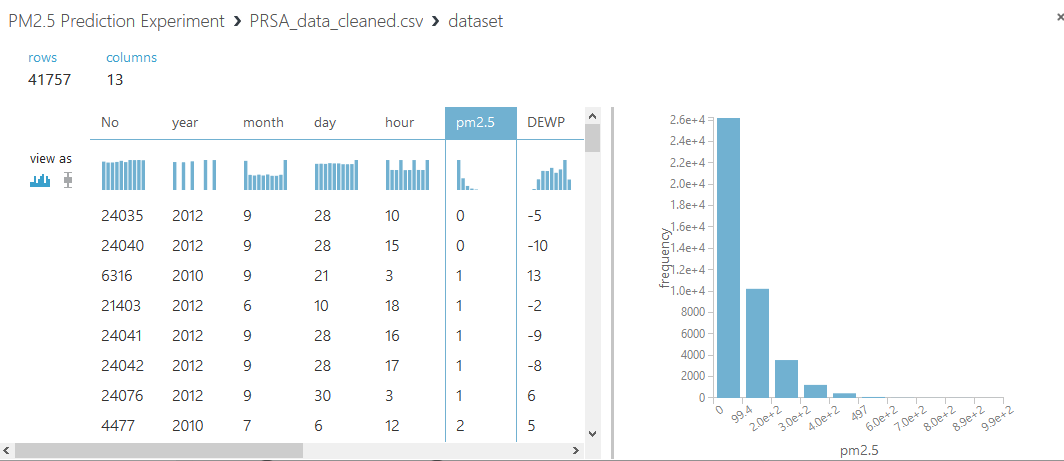


Figure 1: Boxplot and bar chart of PM2.5 data, visualized using Azure ML Studio.

.

**2.2 Visualization and Exploration**

I used a python script to visualize the univariate dataset across time. See *Figure 2* for a graph of this dataset.

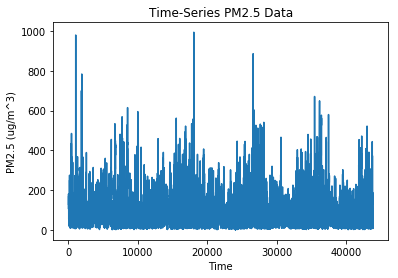


Figure 2: A graph of PM2.5 values by time

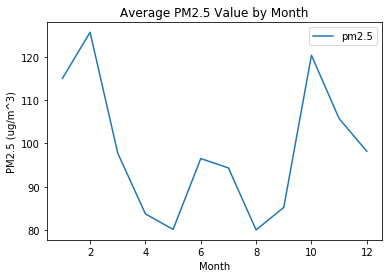
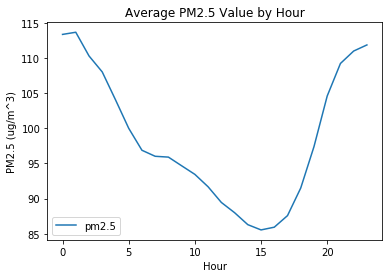
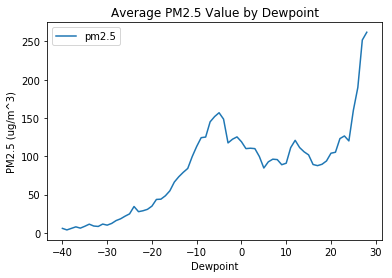
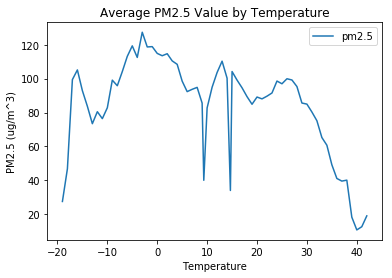
 I noticed from *Figure 2* that there may be interesting trends according to the collection time, as the data spans five years. Using a python script, I summarized the PM2.5 data by month collected, and separately by hour collected. I then performed an average PM2.5 statistic according to each summary and graphed them accordingly. See *Figure 3* for these two graphs.

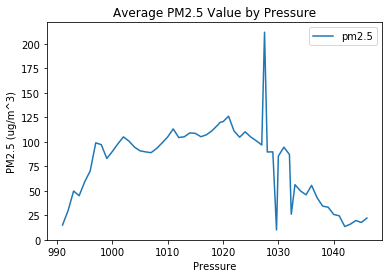
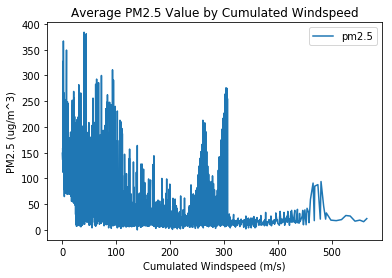
Figure 3: Summarized average PM2.5 values (left) by month and (right) by hour.

To explore the relationships that the meteorological data might have with the PM2.5 data, I visualized four other summarized statistics: dewpoint, temperature, pressure, and cumulated windspeed. See *Figure 4* for those graphs.



(b)

(a)

****

(d)

(c)

Figure 4: Summarized average PM2.5 values by (a) dewpoint, (b) temperature, (c) pressure, and (d) cumulated windspeed.

* 1. **Modeling**

The first prediction model performed was the ARIMA time series model. I ran this on the univariate PM2.5 dataset in Azure Notebooks using a python code provided by Jason Brownlee (2019). I then moved into Azure ML Studio to design and test various regression models using the multivariate dataset.

* + 1. **ARIMA Model**

ARIMA is an acronym which stands for Autoregression Integrated Moving Average. The statistical model is trained on a history of values which inform the model to predict future values. The input parameters include the number of lag observations considered, the number of times an observation is differenced with the previous observation(s), and the size of the moving average window. In my Jupyter Notebook, I utilized the full ARIMA model with lags of 1, a difference level of 1, and a moving average window of 1, as written by Brownlee (2019). I trained the model on the full dataset minus the final 100 days. I then evaluated the model with a root mean squared error (RMSE) and plotted the expected and predicted values on the same graph to visualize the results.

* + 1. **Azure ML Studio Models**

I began my Azure Machine Learning Studio regression models by running Pearson correlation coefficients across all columns. This was meant to inform my choices of columns to use for prediction of the PM2.5 values. The absolute values of all coefficients run across the PM2.5 column were less than 0.25. The strongest relationship with PM2.5 was the Cumulated wind speed column at r=-0.247784, which is still not a strong relationship. I decided to try a few different column combinations for use in my regression models to experiment for the best combination. I used the RMSE statistic to compare different regression models, accepting the model with the lowest RMSE as the best model.

I started with some basic linear regression models, with varying columns used for prediction and altering the proportion of data used for training and testing. I kept the default parameters for this model, including an ordinary least square solution method, an L2 regularization weight of 0.001, and no random number seed.

Next I compared the best linear regression model with a Bayesian linear regression model, retaining the best combination of columns for prediction and percentage of training data which was demonstrated by the best linear regression model. The default parameter used in the Bayesian linear regression model is a regularization weight of 1. I also added the following three regression models for comparison: neural network regression, boosted decision tree regression, and decision forest regression, all with their default parameters. Default parameters for neural network regression include a single parameter trainer mode, the hidden layer specified as fully-connected case, 100 hidden nodes, a learning rate of 0.005, 100 learning iterations, initial learning weights diameter set to 0.1, a momentum of 0, a min-max normalizer, and no random number seed. Default parameters for boosted decision tree regression include a single parameter trainer mode, maximum number of leaves per tree at 20, minimum number of training instances required to form a leaf at 10, a learning rate of 0.2, 100 total trees constructed, and no random number seed. Default parameters for decision forest regression include a bagging resampling method, a single parameter trainer mode, 8 decision trees, a maximum depth of trees at 32, 128 random splits, and the minimum samples per leaf node at 1.

Based on the results of the various regression models, the decision forest regression proved to be the best model. I then compared a few different decision forest models with each other, varying the values of number of decision trees and maximum depth of trees, keeping other parameters at their default values, to try to find the best decision tree model for the data.

Finally I used the tune model hyperparameters module on a decision forest regression model to try a few combinations of parameters to see if there were ideal numbers I hadn’t yet tried. I used random sweep with a maximum of 5 runs, a random seed of 0, and used RMSE for my comparison metric. For this decision forest regression model I used the bagging resampling method with a parameter range trainer mode. My number of decision trees parameters were 100, 125, and 150. The maximum depth of trees parameters were 1, 16, and 40. The number random splits per node parameters were 1, 128, and 1024. The minimum number of samples per leaf node parameters were 1, 4 and 16, and I allowed unknown values for categorical features. With the results from this tune model run, I tried one last decision forest regression model with new parameter levels to try to decrease the RMSE as much as possible. See *Figure 5* for an image of my final Azure ML Studio workspace.

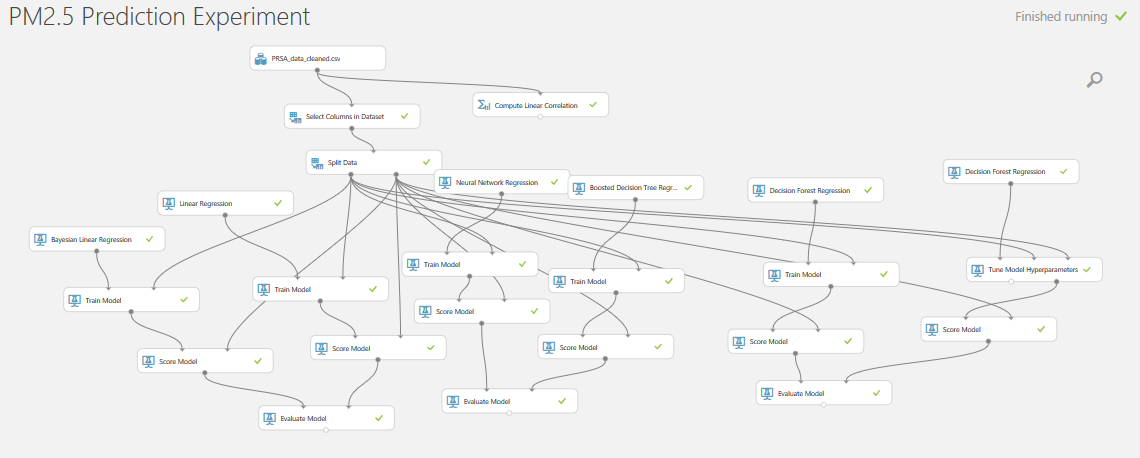


Figure 5: My final Azure ML Studio workspace setup.

1. **Results and Discussion**

According to the summarized PM2.5 by month and hour (see *Figure 3*), the peak PM2.5 measurements occur overwhelmingly in the wintertime, between October and February, and at night, between 8PM and 5AM. Considering these trends together makes sense, as winter nights would be the coldest times in Beijing. Beginning in the 1950s policy changes in Northern China provided winter heating in the form of coal-fired boilers for cities north of the Huai River. Coal is a major contributor to pollution and particularly to PM2.5 levels. Researchers Liang et al. (2015), who provided the associated report with this dataset, concluded from their analysis of the data that during the winter heating months, coal-heating was responsible for a third of the PM2.5 levels. Researchers Wang, Li and Haque (2019) studied particulate matter levels in multiple northern cities in China and demonstrated that levels have measurably decreased since heightened environmental protections in the form of heating policy changes have been put into place, particularly in recent years.

**3.1 ARIMA Model**

The RMSE from the univariate ARIMA model for predicting the final 100 hour PM2.5 levels was 33.037, which is quite accurate given the whole dataset’s standard deviation of 92.05 ug/m^3. See *Figure 6* for a graph of expected and predicted values plotted together from this model.

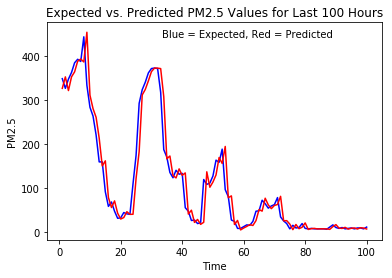


Figure 6: Graph of expected and ARIMA model predicted PM2.5 values plotted together. Blue line is expected values, red line is predicted values.

This model was tested on only 100 values out of a total of 41,757 PM2.5 values, which is a training set of 99.7605% of the data. In order to make it a model with 90% training data, I would need to test it for the final 4,176 hourly PM2.5 measurements, which was not possible in my chosen platform due to timeout errors for the model. Predicting the final 100 days took about two hours to complete in Azure Notebooks. Given the graph in *Figure 6* and the RMSE, I am pleased with the results of this ARIMA model in predicting future PM2.5 measurements from this dataset. I believe it is comparable with the multivariate Azure ML Studio regression models, although the very small percentage of testing data in this ARIMA model should be kept in mind when comparing between these univariate and multivariate models.

**3.2 Azure ML Studio Models**

The Pearson correlation coefficients run for each numeric column with the PM2.5 column did not prove to be very helpful in choosing columns to predict the PM2.5 values. See *Table 1* for the coefficient results.

|  |  |
| --- | --- |
| **COLUMN CORRELATED AGAINST PM2.5** | **PEARSON CORRELATION COEFFICIENT** |
| row number | -0.017706 |
| year | -0.01469 |
| month | -0.024069 |
| day | 0.082788 |
| hour | -0.023116 |
| dewpoint | 0.171423 |
| temperature | -0.090534 |
| pressure | -0.047282 |
| cumulated wind speed | -0.247784 |
| cumulated hours of snow | 0.019266 |
| cumulated hours of rain | -0.051369 |

Table 1: Pearson correlation coefficients for each numerical column in the dataset as run against the PM2.5 column.

The correlation results are quite poor. It is strange that the row number, which is a unique value for each observation, has a larger correlation with PM2.5 than the year, although this does indicate that there is not an outlier year with much higher or lower PM2.5 values. Likewise, the day has a larger correlation than the month, pressure, cumulated hours of snow, and cumulated hours of rain. This is surprising, as the day of the month should not be informative of the PM2.5 level. Also, when exploring the data, it looked as though the month would be a good column to include in predicting PM2.5 levels, since the winter months had much higher average PM2.5 levels than the summer months. This could be due to just a few very far outliers, considering the highly skewed nature of this dataset. Finally, it looks as though dewpoint, cumulated wind speed, and temperature are the most informative columns for predicting PM2.5 levels, according to these Pearson correlation coefficients.

|  |  |  |
| --- | --- | --- |
| **COLUMNS USED FOR PREDICTION OF PM2.5** | **TRAINING**  **DATA (%)** | **RMSE** |
| dewpoint, temperature, and cumulated wind speed | 90 | 79.925 |
| month, hour, dewpoint, temperature, pressure, cumulated wind speed, cumulated hours of snow, cumulated hours of rain, combined wind direction | 75 | 78.650 |
| month, hour, dewpoint, temperature, pressure, cumulated wind speed, cumulated hours of snow, cumulated hours of rain, combined wind direction | 90 | 77.276 |
| day, hour, dewpoint, temperature, pressure, cumulated wind speed, cumulated hours of rain, combined wind direction | 90 | 76.928 |

Given these results, I started my regression prediction models off with a few different combinations of columns used for prediction of PM2.5. The first models I ran were simple linear regression models. See *Table 2* for parameter combinations and RMSE results from these models.

Table 2: Parameters used and RMSE results for simple linear regression models.

Clearly, the linear regression models are not very good predictor models for PM2.5. All RMSE values are in the 70s, which is not impressive considering the dataset’s standard deviation of 92.05. Following the linear regression results, I used the input parameters of the best model, same columns and 90% training data, to compare with other regression models. See comparisons of RMSEs from all regression models in *Table 3*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MODEL** | Linear Regression | Neural Network Regression | Bayesian Linear Regression | Boosted Decision Tree Regression | Decision Forest Regression |
| **RMSE** | 76.928 | 99.308 | 77.397 | 50.004 | 40.315 |

Table 3: RMSE results compared for all regression models using default parameter values.

Both the boosted decision tree and the decision forest models performed much better than the linear regressions, which does make sense since the relationships of PM2.5 with other columns are not linear. I was surprised at how poorly the neural network regression performed, which tells me that this dataset is probably not well suited to this regression model. Given these results, I pursued testing with decision forest regressions only. I tweaked the number of decision trees and maximum depth of decision trees parameters to find improved models. See *Table 4* for these models’ parameters and RMSE results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NUMBER OF DECISION TREES** | 8 | 20 | 50 | 100 | 100 |
| **MAXIMUM DEPTH OF TREES** | 32 | 50 | 32 | 50 | 32 |
| **RMSE** | 40.315 | 38.414 | 37.804 | 37.651 | 37.549 |

Table 4: Parameters and RMSEs for various decision forest regression models.

RMSE results improved with increased number of decision trees, but not with increased maximum depth of trees. The default value of 32 maximum depth paired with 100 decision trees yielded a slightly lower RMSE than the same parameters with 50 maximum depth of trees. To further fine-tune my decision forest regression model, I performed a tune model hyperparameters module, to test out various combinations of parameter values. See *Table 5* for the parameter combinations and RMSE results from this module, compared with the previous best model and a final run with a combination of parameter values which are a compromise of the hyperparameter values and the previous best model parameters.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Previous best model** | **Best hyperparameter module model** | **Compromise model** |
| **NUMBER OF DECISION TREES** | 100 | 138 | 138 |
| **MAX. DEPTH OF TREES** | 32 | 26 | 32 |
| **NUMBER OF RANDOM SPLITS PER NODE** | 128 | 1012 | 1024 |
| **MINIMUM NUMBER OF SAMPLES PER LEAF NODE** | 1 | 5 | 5 |
| **RMSE** | 37.549 | 39.886 | 40.041 |

Table 5: Parameters and RMSE results for final decision forest regression models.

The tune model hyperparameters module found the best combination of parameters to have an RMSE of 39.886, which is higher than the best decision forest regression model I found myself. This is probably due to the limited number of runs set to 5 maximum. This setting is due to my own time and computing power restraints. It would be interesting to increase this setting in a future study to find parameters which yield an RMSE of less than 37. In trying to design a decision forest regression model with a mix of parameters from my own model and the hyperparameter module model, the RMSE was even higher than the hyperparameters model. This is probably due to an overfitting of the model, where increased parameters lead to less accurate predictions.

1. **Conclusion**

Both the multivariate decision forest regression model in Azure ML Studio and the univariate ARIMA model in python performed very well as prediction models for PM2.5 values. It is difficult to compare the univariate with multivariate models simply due to the differences in percentage of training data used. Without being able to decrease the training to testing data in Azure Notebook, I ran a test of my best decision forest model in Azure ML Studio using 99.7605% training data, to see how the model’s RMSE would compare with the ARIMA model using an equal percentage of training data. The result was RMSE = 31.333 compared with the ARIMA model’s RMSE = 33.037. This does help to compare the models on a more even ground but using such a high percentage of training data is not generally a good rule for prediction models. With training data being equal, it seems that the multivariate decision forest regression model is slightly better at predicting PM2.5 values than the univariate ARIMA time-series model. However, they are quite close in accuracy, and the ARIMA model requires far less data for predictions.

For researchers wishing to design prediction models for PM2.5 values, it seems that collecting univariate measurements is nearly as informative as multivariate datasets of associated time and weather conditions. Although we do see discrete spikes of PM2.5 levels during the winter nights in Beijing, these seem to be extreme outliers which do not meaningfully inform prediction models of PM2.5 levels. These conclusions are in line with previous research which found that non-linear models with noise reduction, such as the removal of extreme data outliers, and primarily univariate time-series models are best at predicting PM2.5 levels, with the possibility of weather conditions slightly improving models (Pérez, Trier and Reyes 2000). Other research has proposed using associated meteorological data to statistically adjust the PM2.5 means and percentiles before comparing PM2.5 levels across time (Liang et al. 2015). It has also been proposed that the geographical configuration of a city, meaning the mountain ranges or basins related to the city’s position, dictate the degree and specifics of how wind direction and strength mitigate PM2.5 levels (Liang et al. 2016). This lends credence to the idea of normalizing PM2.5 levels by associated weather factors before comparing PM2.5 measurements across geographic areas.

For these reasons, I would first recommend a decision forest model with as many associated time and weather conditions as possible for prediction columns on large datasets, but without this availability for large collection, a univariate time series regression model such as ARIMA works nearly as accurately for predictions. If I were to conduct this experiment again, I would improve data cleaning efforts beforehand by removing extreme outlier values and possibly statistically adjusting PM2.5 levels according to certain meteorological and/or seasonal elements. These preprocessing steps could improve model performance and make the models more universally relevant for other datasets.

**Supplemental Materials**

**S1 File. Cannon\_BigData\_FinalProject\_PM2.5\_Beijing.ipynb.** This file is a Jupyter Notebook containing python codes used to explore the dataset, along with the results of those codes.

**Acknowledgements**

I would like to thank my Professor, Tony Wang, who taught this Big Data class (GEOG4990) of which this paper is my final project. Thanks to Jason Brownlee for providing the ARIMA model python code and explanation. I would also like to thank the UC Irvine Machine Learning Repository for providing open access datasets, and Song Xi Chen for donating the Beijing PM2.5 Dataset.

**References**

Brownlee, Jason. 2019. “How to Make Manual Predictions for ARIMA Models with Python.” August 28, 2019. <https://machinelearningmastery.com/make-manual-predictions-arima-models-python/>.

Liang, Xuan, Shuo Li, Shuyi Zhang, Hui Huang, and Song Xi Chen. 2016. “PM 2.5 Data Reliability, Consistency, and Air Quality Assessment in Five Chinese Cities: CONSISTENCY IN CHINA’S PM 2.5 DATA.” Journal of Geophysical Research: Atmospheres 121 (17): 10,220-10,236. <https://doi.org/10.1002/2016JD024877>.

Liang, Xuan, Tao Zou, Bin Guo, Shuo Li, Haozhe Zhang, Shuyi Zhang, Hui Huang, and Song Xi Chen. 2015. “Assessing Beijing’s PM 2.5 Pollution: Severity, Weather Impact, APEC and Winter Heating.” Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 471 (2182): 20150257. <https://doi.org/10.1098/rspa.2015.0257>.

Pérez, Patricio, Alex Trier, and Jorge Reyes. 2000. “Prediction of PM2. 5 concentrations several hours in advance using neural networks in Santiago, Chile.” Atmospheric Environment 34 (8), 1189-1196.

Wang, Shulin, Yongtao Li, and Mahfuzul Haque. 2019. “Evidence on the Impact of Winter Heating Policy on Air Pollution and Its Dynamic Changes in North China.” Sustainability 11 (10): 2728. <https://doi.org/10.3390/su11102728>.