# A Study of Models For Air Quality Prediction Based on PM 2.5

**Tanvi Thakur, Japnit Singh Sethi, Shreyas Bhat**
Virginia Polytechnic Institute and State University
tanvithakur94@vt.edu, japss96@vt.edu, bshreyas@vt.edu

## 1 Introduction

Air quality deterioration has been one of the growing challenges of this decade. WHO reports estimate that 90 percent of the population don't have access to clean air. Major cities of the world like Beijing, New Delhi , Manila among others have high levels of smog and smoke both indoors and outdoors. The detrimental effects of air pollution are a threat to our health and climate. This is evident when we look at spike in number of deaths due to health conditions aggravated due to bad air quality. According to surveys, nearly seven million premature deaths are caused every year due to conditions like heart disease, stroke, lung cancer and acute respiratory infections. This results in a need to monitor and analyse the air quality to help make informed decisions in advance. The current growth in Machine Learning allows us to leverage its predictive abilities to efficiently estimate the air quality by training a model on a past data using a variety of algorithmic approaches.

In this project we assess the air quality based on the concentration of one category of pollutants, Particulate matter of diameter 2.5 micrometers or lesser or PM2.5. This problem can be broken down into two tasks: regression to predict the levels of PM2.5 and classification to label the air quality based on the predicted levels of the pollutant. For the first tasks, we compare the abilities of different regression models and time series forecasting, ARIMA model to predict levels of PM2.5 and analyse how close the predicted data is to test dataset. This performance is evaluated using Mean Absolute Error and Root Mean Squared Error. For the classification task, we evaluate the performance of a set of classifiers at solving the multi-class classification problem posed here using a variety of performance metrics: accuracy, F-score, average precision, precision/recall break-even point.

## 2 Related Work

The project draws its inference from STATLOG study which dates back to the early 90s. But with the advancement of machine learning algorithms, there are many new models coming up. In 1997, Cooper et al. conducted research on medical data evaluating the performance of few models using accuracy and ROC. Barai et. al and Feng et.al predicted the air quality using neural networks using the concentration of PM2.5 in air. Azami et. al incorporated a different approach for finding out relationships between particulate matter and meteorological variables. Wang et. al conducted the experiment for Beijing city in particular. Sun et. al made his predictions using the hidden Markov Model and discussed the 24-hour average PM 2.5 concentration in Northern California. Shaharuddin et. al worked on determining the relations between meteorological factors and particulate matter using wavelet transforms. Diaz Robles et. al used ARIMA and neural networks for predicting the amount of PM2.5 especially in urban areas. But, all these papers had a limited extent of using just a few models. For our project, we have used several models to determine and analyse the air quality index using regression, time-series forecasting and classification.

# 3 Methodology

### Overview

As discussed the main goal of the project is to predict the air quality of a given city based on meteorological features like temperature, humidity, pressure etc. This involves a 2 step problem, first to predict the level of pollutants based on the inputs then asses the air quality based on the levels. For this test case, we have conducted the air quality analysis for the one city, Beijing, considering the data for the year of 2014. Before we can solve the problem, we first pre-process this data to eliminate unnecessary features and reorganize as needed for the task. The data then reflects relevant meteorological features and air quality information out of which the training set is the dataset from January to November and the test set is considered as data for the month of December.

We then approach the problem by first building the regression models listed in the sections that follow to predict the pollutant levels on an hourly basis based on the meteorological data. The models are then evaluated based on their prediction of the hourly values of PM2.5 for the test data. The classifications task is then run to classify the air quality into 6 classes based on severity. Here we assess the performance of different classifications models for this task based on their ability to label the air quality based on the levels of PM2.5. The evaluation metrics used for this assessment are Accuracy, Precision, Recall and F-1 scores.

### Datasets and Pre-processing

The datasets used in this project are:

1. Historical data containing concentration of PM2.5 on an hourly basis used for time series prediction.
2. Meteorological data containing the factors such as average temperature, relative humidity percentage, atmospheric pressure at sea level, atmospheric pressure, fog factor and winds that play an important role in determining the air quality index.

In order to mould the dataset according to our requirements, we carried out data pre-processing by carrying out multivariate analysis on the features available. Thereby eliminating irrelevant features like direction of wind, PM2.5 units, cloud cover. For columns wherein we had missing values, we used mean of that feature value to fill up the null values. Certain rows had values "10 or more", so we have considered all those values as 10. Lastly, normalization and merging of historical and meteorological data was carried out. The next task was to come up with a database so that the project could be used dynamically for any parameter or city. In order to ensure that the project has more generality, we used MongoDB (schema less database).

**Correlation between features**:

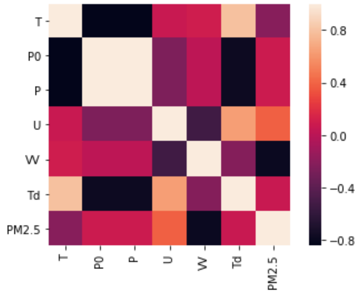


Figure 1: Correlation

Correlation determines the extent to which two variables are related. If the value of correlation is positive, it means that the two variables are directly proportional to each other. Negative value signifies that the two variables are inversely proportional. If the value is zero, it means there exists no relationship. Figure 1 shows the correlation heat map for all the 6 features considered for our analysis. We can infer that P0 and P are highly correlated. Hence, we have not considered P0 in our experiment. Similarly, the diagonal correlation value is 1 signifying all the features are correlated to each other as expected.

**Performance Metrics**:

**Mean Square Error**: It measures the quadratic mean of the differences between the actual values and the values predicted by the models. It can be calculated as:

$$\sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2}$$

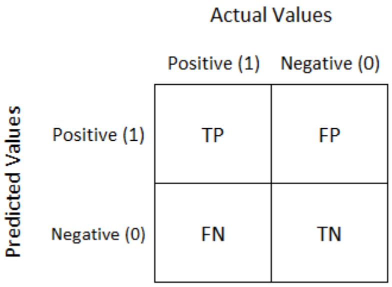
We have chosen this as one of our metrics since it has the benefit of penalizing very huge errors by taking square of the errors.

**Mean Absolute Error**: It is simply the average of the absolute differences between the true value and the predicted value provided all the individual differences have equal weight. It can be calculated as:

$$\sqrt{\frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y_j}|}$$

We chose mean absolute error as another metric since it assigns equal weightage to each difference.

**Confusion Matrix**: In this matrix, each column tells the actual class and each row tells the instances is a predicted class. as shown in the figure below



| | | Actual Values | |
|---|---|---|---|
| | | Positive (1) | Negative (0) |
| Predicted Values | Positive (1) | TP | FP |
| | Negative (0) | FN | TN |

True Positive(TP) - When the prediction matches the actual value and both the values are yes.
True negatives(TN) - When the prediction matches the actual value and both the values are no.
False positives(FP) - When the predicted value is yes while the actual value is no.
False negative(FN) - When the predicted value is no and the actual value is yes.

**Precision**: It is defined as ratio of true positives and total number of true and false positives. Mathematically, we can calculate as:

$$Precision = \frac{TP}{TP + FP}$$

**Recall**: It is defined as the ratio of true positives and total number of true positives and false negatives. It can be calculated as:

$$Recall = \frac{TP}{TP + FN}$$

**F-Score**: It is simple the harmonic mean of both recall and precision and can be calculated as:

$$F - Score = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

The models built with their performance metric results and observations are listed in the subsequent sections.

### 3.1 Logistic Regression

This algorithm predicts the probability of a categorical dependent binary variable that can be either 0 (no, failure) or 1 (yes, success). Also, the marginal probabilities of the training data are well preserved. In this case, we have used the L2 norm for penalization and stochastic gradient descent solver for predicting the value for the given dataset. Figure 2 shows the predictions of this model against test data.

### 3.2 Decision Tree

The main idea behind this algorithm is that it breaks the dataset into smaller chunks and then constructs a tree consisting of root, decision and terminal nodes. For making predictions, the values of root attributes are compared with that of the record's attribute. The test case for some attribute is represented by the nodes.

We have used the maximum depth of the tree as 5 since having a too deep tree may result into overfitting. To measure the quality of split, we have used Mean Square Error(MSE) as the measuring metrics in order to minimize the L2 loss using the mean of each terminal node. Figure 3 shows the predictions of this model against test data.

### 3.3 KNN

This non-parametric algorithm uses 'feature similarity' to predict the value of new data points such that the new data point is given a value on the basis of how much it resembles the points in the training set. We have taken 10 as the number of neighbours since taking a higher value of K will help in avoiding overfitting. Euclidean distance metric and uniform weights for all points in a neighborhood to generate the model. Figure 4 shows the predictions of this model against test data.

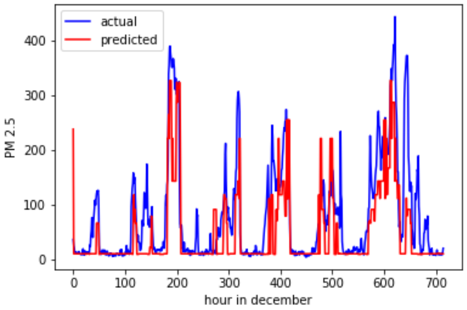


Figure 2: Logistic Regression

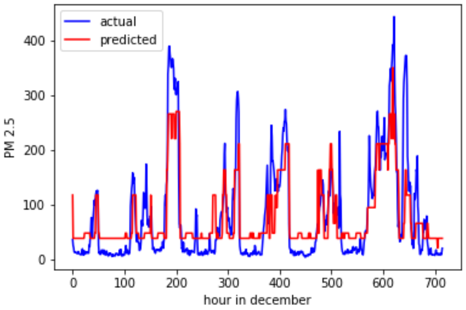


Figure 3: Decision Tree

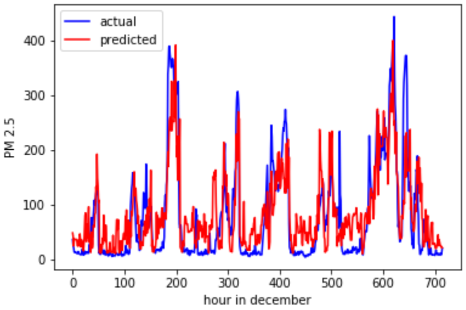


Figure 4: KNN

## 3.4 SVR

### 3.4.1 Linear and Non Linear(Radial basis)

This non-parametric algorithm is a combination of Support Vector Machine and Regression and generally works on the principle of maximal margin and separating hyperplane. The ultimate goal is to ensure that the error does not go beyond a certain threshold. In this case, we have tried to predict the values based on linear and nonlinear (Radial Basis Function) kernel functions. On visualizing, we can clearly see that the SVR with RBF outperforms linear SVR. One of the possible reasons is that RBF SVR creates non-linear combinations of features which can be very well aligned to our dataset which is also non-linear in nature.

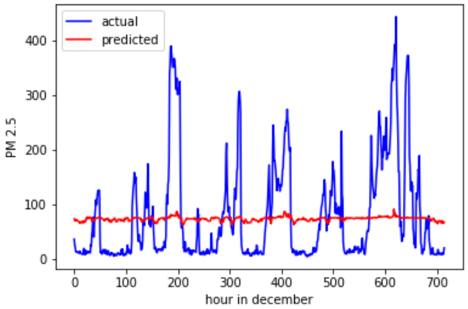


Figure 5: SVR- Linear

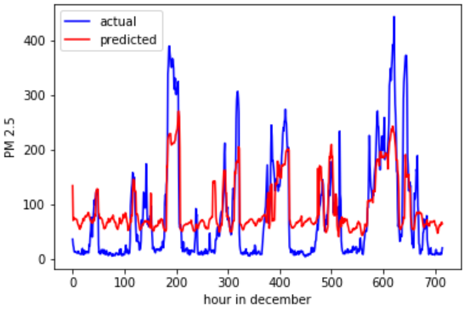


Figure 6: SVR- Radial Basis

Figure 5 and Figure 6 show the predictions of SVR with linear and RBF kernels against test data.

## 3.5 Multi Layer Perceptron

Multi Layer Perceptron is a feedforward neural network. It normally consists of 3 layers - an input, hidden and an output layer to predict the output. The main idea behind this algorithm is that it models on the basis of correlation between input and output. The training generally is done by adjusting the parameters so that the error is minimized. Each node (excluding input nodes) is a neuron which uses a non linear activation function incorporating backpropagation methodology. In this case, we have used 2 hidden layers(since learning with more layers is better but it comes at the cost of more training time) with 10 neurons (to avoid overfitting and to prevent increase in training time) for each hidden layer and ReLU as the activation function to construct the model. Figure 7 shows the predictions of this model against test data.

## 3.6 Random Forests

In this algorithm, a random sample of data points are drawn with replacement called bootstrapping. At testing times, predictions are made by averaging the predictions of each decision tree. The main idea behind training each tree on different samples is done so that the entire forest has low variance but without increasing the bias. In this case, we use 10 as the number of trees. Increasing the number of trees might give better results however after a certain point the performance deteriorates but the computation time increases for learning these extra trees.Figure 8 shows the predictions of this model against test data.

## 3.7 Gaussian Naive Bayes

Naive Bayes selects the outcome having the highest probability assuming all features are independent.We can then use Maximum A Posteriori (MAP) estimation to estimate the outcome. The main advantage is low computational cost since its maximum likelihood training needs linear time.Figure 9 shows the predictions of this model against test data.

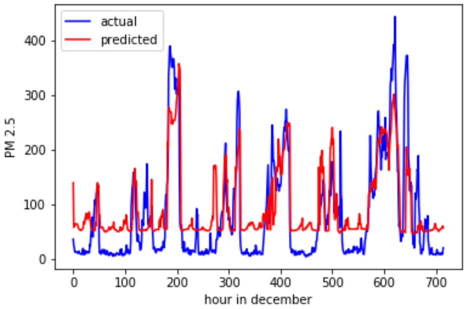


Figure 7: MLP

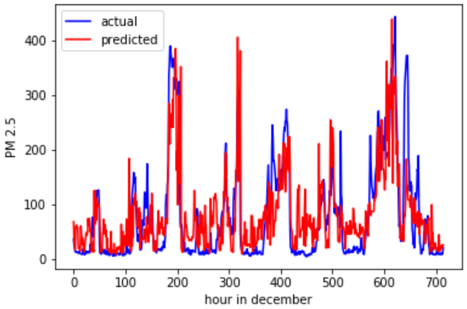


Figure 8: Random Forests

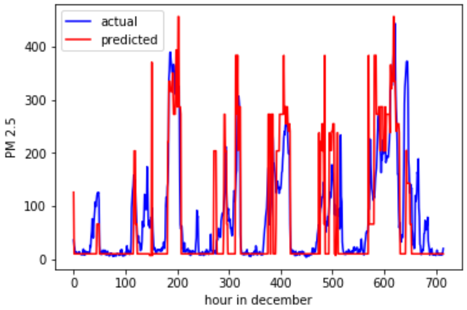


Figure 9: Naive Bayes

### 3.8 Regression Task and Observations

Table 1 shows the values of Mean Square and Mean Absolute Error for each regression model considered. We can clearly see that the Decision Trees perform the best followed by MLP, KNN,SVR, Random Forest, Logistic Regression. Guassian Naive Bayes is the worst performer. Similarly, if we consider mean absolute error, we understand that Decision trees perform the best followed by SVR, MLP, logistic regression, KNN, Guassian Naive Bayes. Random Forest performs the worst. However, if we consider the graphs plotted for each model, we can come to a conclusion that KNN, Random forest have the most accurate predictions followed by logistic regression and gaussianNB. The performance is not satisfactory for Decision Tree, MLP and SVRs. In the next section we explore these performance metrics for the ARIMA model.

Table 1: Regression Observations

| Regressor | RMSE | MAE |
|---|---|---|
| **Logistic Regression** | 68 | 41 |
| **Decision Tree** | 53 | 39 |
| **KNN** | 56 | 42 |
| **SVR** | 57 | 41 |
| **MLP** | 55 | 41 |
| **Random Forest** | 61 | 50 |
| **Gaussian Naive Bayes** | 72 | 43 |

### 3.9 ARIMA - Time Series Analysis

In general, A time series is a sequence where a metric is recorded over regular time intervals. In our case the PM2.5 values used have been recorded and observed on an hourly basis. Since this data by

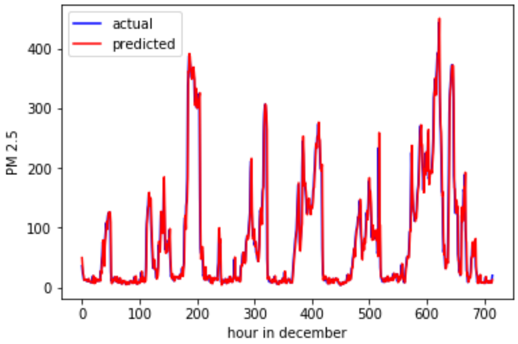


Figure 10: ARIMA - Time Series Analysis

nature is non-seasonal but exhibits a time-bound(hourly) pattern, it is an ideal candidate to forecast using the ARIMA or Auto Regressive Integrated Moving Average model . As the name suggests this model is essentially a ‘ Auto Regressive’ linear model which implies that it uses its own lags as predictors. So we have the term ‘p’ corresponding to the order of these lags. However as we know Linear models work best when the predictors are not correlated and are independent of each other. Our time series has to be made stationary. The common practice is using differencing (subtracting an observation from an observation at the previous time step), this is the ‘Integrated’ component.Now we get another term corresponding to the order of differencing ‘d’. Finally, we have the ‘Moving Average’ aspect of the model. The term corresponding to this is ‘q’.
We can therefore describe the ARIMA model as follows:

**Predicted Value (Y) = Constant($\alpha$) + Linear combination Lags of Y (upto p lags) + Linear Combination of Lagged forecast errors (upto q lags)**

**Before we start the time series is made stationary by differencing , where we use the term ‘d’ to ensure the right amount of differencing.**

The best fit for our ARIMA model is obtained with p = 5 , d=1 and q=0, which implies that our model is using the ‘ARI’ components without the moving average component of the model. The results for the predictions made using this model are as shown in Figure 10. And the error metric obtained are: Mean Absolute Error = 5 and Mean Square Error = 8
As we can see the lowest MAE and RMSE are observed for this model, yielding the best predictions. This is mainly because of the ‘time series sequence’ nature of the prediction parameter PM2.5 pollutants in air. Furthermore, Time series forecasting in this fashion is very efficient for the following reasons : (i) A time series irrespective of whether it is stationary or not can be analyzed;(ii) Its ability to extract the underlying serial correlations and trends in the data while minimizing the noise, as its based on historical data development. Therefore, the ARIMA model is the best choice for this task.

### 3.10 Classification task and observations

The second task of predicting the air quality as proposed is Classification. Table 2 shows the categorization considered according to severity and the corresponding air quality rating. For our dataset we set up a multi-class classification with 6 different classes that makes the assumption that each sample is assigned to one and only one label. Higher the PM 2.5 value higher is the severity of the situation and thus corresponds to a higher rating.

We have evaluated the classifiers listed in Table 3 based on the metrics as listed accuracy Precision, Recall and F-score in order to select the best classifier.

We found that Gaussian Naive Bayes which is an eager learner classifier (construct a classification model based on given training data before receiving data for classification) produces the best evaluation scores as it has the highest value for the chosen performance metrics. Further,it is scalable to

Table 2: Multiclass Classification

| PM 2.5 | Severity | Rating |
|---|---|---|
| 0-30 | Good | 1 |
| 31-60 | Satisfactory | 2 |
| 61-90 | Moderately polluted | 3 |
| 91-120 | Poor | 4 |
| 121-250 | Very Poor | 5 |
| 250+ | Severe | 6 |

Table 3: Calculation of Accuracy, Precision, Recall and F-score

| Classifier | Accuracy | Precision | Recall | F-Score |
|---|---|---|---|---|
| MLP | 0.74 | 0.73 | 0.78 | 0.75 |
| DT | 0.67 | 0.71 | 0.55 | 0.62 |
| KNN | 0.72 | 0.76 | 0.60 | 0.67 |
| Random Forest | 0.71 | 0.78 | 0.57 | 0.66 |
| Logistic Regression | 0.54 | 0.65 | 0.17 | 0.27 |
| Gaussian NB | 0.75 | 0.67 | 0.98 | 0.80 |

larger datasets and is also efficient in that it is a linear process compared to an expensive iterative approximation like other classifiers. Another promising classifier for this task, as we can see, is the Multi Layer Perceptron classifier with the second highest value for the selected criteria. Here we have used a Neural Network for classification with 3 layers. The architecture used is 2 hidden layers, on each with 10 neurons , and an output layer. The results noted above are obtained with sigmoid activation function.

## 4 Conclusion

From this experiment the most important takeaway is that Machine Learning models can be used effectively in any task by first understanding the nature of the task and the underlying statistical nature of the data. The air quality data used here is non-linear and time-bound in nature. Hence we see that the use of Time Series analysis using the ARIMA model works best for prediction of continuous values( levels of PM2.5). Among the regressors, promising results are observed with KNN and Random Forest. MLP, Decision Tree, SVR perform poorly for this task. In the classification task, we see that GaussianNB and MLP classifiers show promising results. While we have pre-processed the data to narrow down the analysis to one category of pollutants, this approach can be expanded further to include other pollutants and factors involved in analyzing the comprehensive Air Quality Index.

## References

[1] Xiang, Xu. (2019). Forecasting air pollution PM2.5 in Beijing using weather data and multiple kernel learning. Journal of Forecasting. 10.1002/for.2599.

[2] Shaharuddin M, Mohd A, Mohd J, Othman N, Karim A, Sopian K (2008) Application of wavelet transform on airborne suspended particulate matter and meteorological temporal variations. WSEAS Trans Environ Dev 4(2):89–98

[3] Kumar, Ujjwal Jain, Vijay. (2010). ARIMA forecasting of ambient air pollutants (O3, NO, NO2 and CO). Stochastic Environmental Research and Risk Assessment. 24. 751-760. 10.1007/s00477-009-0361-8.

[4] Barai S.V., Dikshit A.K., Sharma S. (2007) Neural Network Models for Air Quality Prediction: A Comparative Study. In: Saad A., Dahal K., Sarfraz M., Roy R. (eds) Soft Computing in Industrial Applications. Advances in Soft Computing, vol 39. Springer, Berlin, Heidelberg

[5] Xiao Feng, Qi Li, Yajie Zhu, Junxiong Hou, Lingyan Jin, Jingjie Wang, Artificial neural networks forecasting of PM2.5 pollution using air mass trajectory based geographic model and wavelet transformation, Atmospheric Environment, Volume 107, 2015, Pages 118-128, ISSN 1352-2310

[6] Wang JF, Hu MG, Xu CD, Christakos G, Zhao Y (2013) Estimation of Citywide Air Pollution in Beijing. PLOS ONE 8(1): e53400

[7] Patricio Pérez, Alex Trier, Jorge Reyes, Prediction of PM2.5 concentrations several hours in advance using neural networks in Santiago, Chile, Atmospheric Environment, Volume 34, Issue 8, 2000, Pages 1189-1196, ISSN 1352-2310

[8] Bhalgat, Pooja Bhoite, Sachin Pitare, Sejal. (2019). Air Quality Prediction using Machine Learning Algorithms. International Journal of Computer Applications Technology and Research. 8. 10.7753/IJCATR0809.1006

[9]Wei Sun, Hao Zhang, Ahmet Palazoglu, Angadh Singh, Weidong Zhang, Shiwei Liu, Prediction of 24-hour-average PM2.5 concentrations using a hidden Markov model with different emission distributions in No