PM2.5 Pollutant Concentration Prediction from Meteorological Data EE 660

Project Type: Individual

Nivedita Suresh, nsuresh@usc.edu

12/01/2018

1. Abstract

The objective of this project is to create a Regression model that can predict, with high accuracy, the concentration levels of the PM2.5 particle in the air given meteorological data in that region(Beijing). The dataset, taken from the UCI machine learning repository is based on a research paper about the relation between Meteorological factors like wind speed, direction, temperature, pressure etc. and PM2.5 concentration. The goal of this project is to use Machine learning tools to try and find this relation and predict the PM2.5 concentration with good performance metric. The data is processed to account for missing values and categorical data. Label Encoding and One Hot Encoding is used to process categorical data and the missing values are removed. The key features are selected from the normalized data using Principal Component Analysis and this data is the input for regression models. Linear Regression, Ridge Regression, Lasso, Random Forest regressor, Stochastic gradient descent and Support Vector regressor are the regression models tried in this project. We divide dataset into Pre-training, training, validation and testing and perform parameter tuning using cross-validation on training data and model selection using pre-training(preliminary) and validation(final) set.

2. Introduction

2.1. Problem Type, Statement and Goals

This project creates a Regression model to estimate the levels of PM2.5 pollutant in the atmosphere. The dataset, obtained from the UCI Machine learning repository consists of meteorological data including temperature, pressure, wind speed, wind direction etc. and the time of recording. The goal is to find a regression model that can predict with high accuracy the PM2.5 pollutant concentration in the air.

The problem is not trivial as it tries to employ Machine learning algorithms to estimate the correlation between meteorological factors and the pollutant levels. This gives us an understanding of what factors must be controlled for cleaner air.

The data contains categorical features. The time (hour, month, day and year) influence the pollutant level and are considered as categorical features. The Wind direction is another categorical feature. These categorical values must be preprocessed before extracting the features. The data also consists of missing labels which must be dealt with. Thus, it requires significant preprocessing. Also, we do not know much about the actual model or nature of this relation. Thus, several regression models need to be tried before deciding which is a good model.

2.2. Literature Review (Optional)

None.

2.3. Prior and Related Work (Mandatory)

None.

2.4. Overview of Approach

For this project several regression models were used to evaluate the best one that gives the lowest error. Linear Regression, Ridge Regression, Lasso Regression, Random Forest Regressor, Support Vector Regressor and Stochastic Gradient Descent Regressor were among the models used. To evaluate the performance of the models two evaluation metrics were employed in this project, Mean Absolute Error(MAE) and R2 score or coefficient of determination.

3. Implementation

3.1. Data Set

As mentioned previously, the dataset is taken from the UCI machine learning repository. It consists of categorical data and real-valued data. The label is the PM2.5 pollutant level which is a real value. There are 11 features in this data set.

FEATURES	YEAR	MONTH	DAY	HOUR	DEWP	TEMP
FEATURE DESCRIPTION	Year of data recorded	Month of data recorded	Day of the month	Hour during which the data was recorded	Dew Point	Temperature
TYPE	Categorical	Categorical	Categorical	Categorical	Real	Real
RANGE	2010,2011,2012, 2013, 2014	{1,2…12}	{1,2…31}	{1,2…24}	(-40,28)	(-19,42)

FEATURES	PRES	CBWD	LWS	LS	LR	PW2.5
FEATURE DESCRIPTION	Air Pressure measured in the region	Wind Direction	Wind Speed	Cumulative hours of snow	Cumulative hours of rain	Output, the pollutant concentration in air
TYPE	Real	Categorical	Real	Integer Real	Integer Real	Real
RANGE	(991,1046)	SE, SW, NE, NW, CV (CV: - Calm and variable)	(0.45,565.49)	[0,27]	[0,36]	(0,1000)

Table1. Features in Dataset and Description

There are 6 numeric features. Four of them are continuous real numbers and the other two take integer values. The meteorological features Temperature, Air Pressure, Dew point and Wind Speed take continuous real values. The cumulative hours of snow and rain take only integer values. While processing the data we

consider real and integer features to be the similar and both are processed in the same way.

There are five categorical features. There are the time features like year, month, day and hour and then there is wind direction which takes five possible values. These features must be encoded to integer numbers and one-hot encoded to process them.

The output is the PM2.5 concentration level which is a continuous and real data. This is what we are trying to predict in the regression problem.

In conclusion, the dataset has 43824 data points, with 11 features and a single real output.

3.2. Preprocessing, Feature Extraction, Dimensionality Adjustment

The dataset contains missing output values, categorical values and sparse and imbalanced feature values. It required preprocessing and feature extraction before applying regression models.

Preprocessing:

Since the output was missing (instead of the data), the missing values could not be imputed and instead the data points containing the missing labels were removed from the dataset. After this process, the number of data points reduced from 43824 to 41758.

The data contained the output in its column, so this needed to be separated before extracting the features.

Normalization was performed on all feature column that had real values. The mean feature value was subtracted from each feature and the standard deviation of the feature was used to scale these features. Standardization is important as it gives it makes all the feature values into the same range. If we do not do this in the preprocessing step, then feature values with higher magnitude like Air Pressure will be given more importance.

$$z = \frac{x - \mu}{\sigma}$$

The performance of the model with and without normalization was compared using pre-training dataset and it was determined that normalization reduced the mean absolute errors.

	Without Standardization	With Standardization
Mean Absolute Error	46.4853	46.4715

Table 2. Comparison of Unnormalized and Normalized data

Feature Extraction:

To handle categorical data, all columns containing categorical values were first encoded to get integer values. Since the first four categorical features (year, month, day and hour) were already integers, this process was required only for the last categorical feature (wind direction). The wind directions are categorical strings which had to be converted to integers using Label Encoding.

These encoded features were then converted into one-hot encoding format. This is important for categorical features were the order is unimportant and the distance of each categorical feature value from one other is zero (taking dot product of the one-hot encoded feature vectors will give us zero).

After performing the number of feature increase from 11 features to 82 features. Principal Component Analysis is used to reduce the dimensionality of the feature space to prevent overfitting. Here the number of components to reduce to was determined by performing using the pre-training dataset. We find that when the number of components chosen is 77, we get the lowest error.

Number of components	Mean Absolute Error
1	60.30529
2	58.18572
3	58.0536
:	:
76	49.03653
77	46.47501
78	47.45553
79	48.05048

Table 3. Comparison of errors due to number of components in PCA

3.3. Dataset Methodology

The dataset containing 43824 data points was first processed to remove all missing outputs. We get a reduced dataset containing 41758 data points. After doing so, the feature extraction technique of label encoding, and one-hot encoding of categorical features was applied. This is because this process is similar for all datasets (pre-training, training, validation, test) and do not depend on other data points in the set.

After this the data was split into pre-training and training_1 data set. The pre-training data contains 4176 points and the training_1 set contains 37581 data points.
The training_1 dataset was then split into training_2 and test data set with training_2 containing 28185 points and test containing 9396 data points.
Since there is sufficient number of data points, the training_2 set was split into training and validation with training set of 22548 points and validation set of 5637 points.
The pre-training set was also split into pre-training-train and pre-training-test of 3758 and 418 points respectively.
All the remaining pre-processing and feature extraction techniques like normalization and PCA were applied after the splits. This is very important as both normalization and PCA must be fitted to the train set and the test set must be transformed accordingly. It cannot be applied before the train-test split because the test set data will also influence the results of normalization and PCA dimensionality reduction which will contradict the rules we need to follow for applying machine learning algorithms to data.
The process is described in the diagram below.

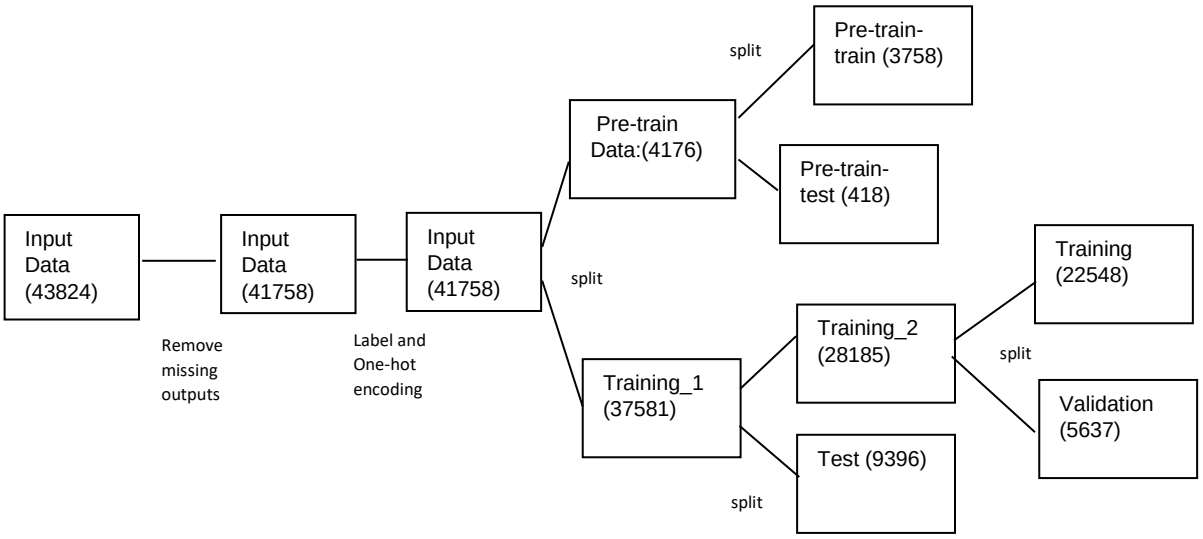


Fig 1. Flow chart describing Dataset methodology

The Pre-training set was used to determine what pre-processing and feature extraction technique to use for the given data. The Pre-training train was used to fit the data and the evaluation was performed on Pre-training test. As mentioned earlier, this was used to select normalization as a pre-processing technique and PCA as a dimensionality reduction technique. The Pre-training was also used to determine what classifiers to use for training. We applied a set of classifiers; Linear regression, Ridge Regression, Lasso, Random Forest Regressor, Support Vector Regressor and Stochastic Gradient Descent Regressor to determine which regression models to use for the actual training. This narrows the number of hypothesis in the hypothesis set by reducing the models we need to train for.

All the information obtained from the Pre-training dataset was used in the training set. The optimal Pre-processing and feature extraction techniques were used, and the reduced list of model hypothesis was taken to train the model on the training data. Cross validation on the training set was used to select the best hyper parameter for each model. The models with two or more parameters required nested cross-validation to select the best hyper parameter. The optimal hyper parameter obtained for each model was then used to train the entire training dataset and the validation set was used to evaluate the performance and select the best model from them,

Thus, for training, we have a hypothesis set containing 4 models to train. We use 5-fold cross validation on each model. Thus the number of data points will be #(training data)/5, 4510 for cross-validation-validation and 18038 for cross-validation-train. For each model, we have a hypothesis set containing the possible model parameters to use in training. The left-out set in cross validation was used to evaluate the performance of the model parameter and the parameter giving the lowest error was used as the optimal parameter. This is repeated for all the models to determine the best model parameters. Also, while training with a certain model parameter, we have another hypothesis set of all possible model training coefficients for regression model

which is infinitely large. E.g. For Linear Regression the set of all possible weights W, is infinite.

Now, we still have the hypothesis set of different models to choose from. We employ the validation set to make this decision. This is a cleaner method as we do not use validation set anywhere in the training process to select the model parameter. The validation is just used to select the best hypothesis out of the four models that give the lowest validation error.

The final model along with its optimal model parameters is used to train the whole training data set and then used on the test dataset to determine the test error. The test set is only used once at the end. It is not used to make any assumptions about the model or hypothesis.

This dataset methodology does not violate any assumptions in machine learning.

3.4. Training Process

The first step in the Training process was to come up with a hypothesis set. The Pre-training set was used for this purpose. The Pre-training set was used to test 6 regression models to determine the three best ones to use for training.

Pretraining:

Thus, the initial model hypothesis set H= {Linear Regression, Ridge Regression, Lasso Regression, Random Forest Regressor, Support Vector Regressor and Stochastic Gradient Descent Regressor}

Ridge regression and Lasso regression models were selected because there are too many features in the model and both Ridge and Lasso models help in selecting a sparser weight vector. This is very useful because in our dataset we have many features a lot of which may not even be significant. Thus, using Ridge and Lasso will help in assigning importance to only significant features.

Linear Regression is the simplest regression model and easily helps in understanding the basic correlation between different features and the output.

Random Forest Regressor is a popular regression model and usually gives very low errors because of the ability to determine a complex model to define the relationship between the input features and the output.

Support Vector Regressor was also selected in the model hypothesis because of the ability to select complex kernels that may help in understanding the relation between input features and output.

Stochastic Gradient descent is an iterative way of getting the function by traversing along the gradient of the loss function. This helps in minimizing the error at each step of the iteration.

After trying all six models on pretraining-train and evaluating the performance on pretraining-test, it was determined that Ridge Regression, Lasso Regression and Random Forest Regressor were the best models. The results are discussed in detail in the section below.

Training and Cross-validation:

The hypothesis set for training is the best three hypotheses selected using the pretraining set. H = {Ridge Regression, Lasso Regression, Random Forest Regressor}.

The validation set is used to select the best hypothesis from the above hypothesis set. Therefore, dimensionality of hypothesis set for validation error estimate is three.

However, before this we need to select the best model parameters for each of the three models. For this we employ cross-validation. The number of data points for cross validation-training set and cross-validation-validation set were discussed in the previous section, 18038 and 4510 respectively.

We consider each model separately:

1. Ridge Regression:

   Ridge regression is basically Linear regression with Gaussian prior. Thus, the objective function is:

   $$J(w) = RSS + \alpha \|w\|_2^{\,2}$$

   $$RSS = \text{Mean-squared error} = \frac{1}{N}\left(\sum_{i=1}^{N}(y_i - w^T x_i)\right)$$

   The Sklearn model uses the model parameter alpha, where alpha = α

   This regression model forces the model to select weight vector w to be sparse by minimizing the magnitude of weight vector $\|w\|_2^{\,2}$.

   For training, we use cross-validation on training set with different values of α, the model parameter.

   We take alpha_val = [0.0001,0.001,0.01,0.1,1,10,15,50,80,100]

   H_{model} = {h(0.0001), h(0.001)…h(100)} #hypothesis =10

   Thus, the hypothesis set consists of 10 elements (10 model parameters) and the leave-out set of cross validation is used to choose the best hypothesis out of 10 possible ones.

   We use 5-fold cross validation and average the errors obtained due to each fold to get the mean train and validation error.

   For training, we still need to determine the weight vectors. Since the feature dimension is 77, the VC dimension of the training hypothesis set for each model parameter is less than 77.

   $$D_{vc}(\text{alpha}) \leq 77$$

   Result:

   The 5-cross-validation is performed, and the mean Mean Absolute Error is plotted against the model parameter. We see that the graph has a minimum error at alpha =50. This indicates that this is the best model parameter.

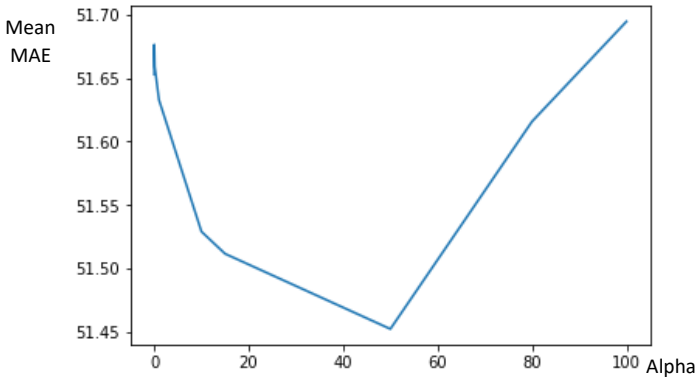


Fig 2. Mean MAE vs Alpha for Ridge Regression

Alpha = 50. Thus using cross-validation on training set we determined the optimal alpha for Ridge regression.

2. Lasso Regression:

The Lasso regression is also like linear regression and uses l1 regularizer or Laplacian prior. The objective function is given by:

$$J(w) = RSS + \alpha\|w\|$$

$$RSS = \text{Mean-squared error} = \frac{1}{N}\left(\sum_{i=1}^{N}(y_i - w^T x_i)\right)$$

We see that this model takes the l1 norm of the weights and thus reduces the sparsity of the weight vector more than ridge regression.

Again, for cross-validation, we use different values of alpha for the model parameter.

alpha_val = [0.0001,0.001,0.01,0.1,1]

H_{model} = {h(0.0001), h(0.001)…h(1)}. #hypothesis =5

There are 5 hypotheses (one for each model parameters) and the leave-out set of cross validation is used to choose the best hypothesis out of the 5 possible options.

We use 5-fold cross validation and average the errors obtained due to each fold to get the mean train and validation error.

The training set requires selecting the weight vectors from an infinite hypothesis set of possible weight values. However, the VC dimension of the set is finite because of the finite features we have.

Thus,

$$D_{vc}(\text{alpha}) \leq 77$$

Result:

After performing 5-fold cross validation, we get the optimal value of alpha for Lasso regression. We see that this value is alpha =0.1.

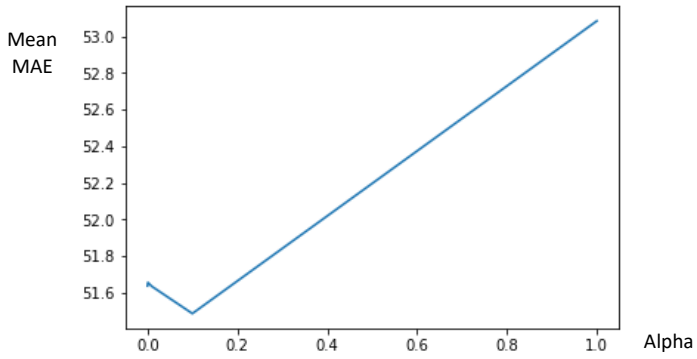


Fig 3. Mean MAE vs Alpha for Lasso Regression

The graph reaches a minimum at Alpha =0.1. This is the optimal model parameter for Lasso regression.

3. Random Forest Regressor:

The Random Forest Regressor divides the training data and the features and perform training on multiple trees. For regression the output is obtained by taking the output given by each tree in the random forest. The multiple number of trees in the random forest ensure that the bias of the tree is reduced, and the resulting output is close to the true value.

Thus, we split the data with sampling for each tree. This process is known as Bootstrapping. The features are also split for each tree. This is very important and really improves the accuracy of the output predicted. Reduced features for each tree reduces the correlation of the output predicted by each tree. Thus, variance of the output will be reduced. Thus, Random forest performs bagging with an attempt to reduce the bias and variance of output prediction.

However, Random Forests have a high tendency to overfit. To prevent this, we must prune the tree by controlling the depth of each tree or reducing the maximum features used in each tree. This requires us to estimate the best model by varying the depth of the tree and the number of trees in the random forest model.

Number of trees = [10,50,80,100]

Depth = [4, 50, None]

Again, we use cross-validation to determine the best model parameter. The number of hypothesis is larger now because we iterate over all possible number of trees and all possible tree depths.

#hypothesis = 4*3 =12

The hypothesis set:

$H_{RandomForest}$ = {h(ntree=10,dep =4), h(ntree=10,dep =50)…h(ntree=100,dep =None)}

The leave-out set in cross-validation must choose the best hypothesis from 12 possible hypotheses.

The Random forest for a model setting also must choose the best decision rule from all possible options. Even though the number of possible hypothesis is infinite, the VC dimension is finite. To find the optimal threshold for each feature, we go through all distinct j dimension for each data points in a region R_m. In the worst-case scenario, all data points have unique feature value j. Therefore, VC dimension is the number of data points for each region. The number of regions (M) depends on the depth.

$$D_{vc} \leq N\_datapoints*M$$

M = number of regions =depth

$$D_{vc} \leq number\_of\_trees *N\_datapoints\_per\_tree*depth$$

Result:

After performing cross-validation, we see that the least MAE is obtained when the number of trees is 80 and the depth of each tree is 50.

Number of trees =80

Depth =50

3.5. Model Selection and Comparison of Results

Preliminary Model Selection:

There were six models in the beginning and the best three models were selected using Pre-training dataset as described earlier. The preliminary model selection was performed without any parameter tuning, using the default parameter values of the model. This is done to just get an idea of which models can be improved by further training and which models will results in no improvement.

Model	Train Mean Absolute Error	Train R2 Score	Test Mean Absolute Error	Test R2 Score
Linear Regression	59.9535	0.287	59.8	0.2437
Ridge Regression	53.59	0.403	53.3	0.3838
Lasso Regression	56.01	0.3359	55.01	0.329
Random Forest Regressor	20.446	0.889	50.247	0.3811
Support Vector Regressor	61.05	-0.009	57.78	0.0322
Stochastic Gradient Descent	56.216	0.3279	53.989	0.3198

Result:

We see that the Linear Regression model and Support Vector Regressor model both perform the worst in the preliminary evaluation. Linear Regression is a very basic model and its performance cannot be improved as the model does not have any hyperparameters that it could tune. Thus, we can safely eliminate Linear Regression model.

Support Vector Regressor does not do well on the training, giving negative R2 scores. There is a possibility that by tuning the right kernels, we might improve our results, but it would still be lower than better models like Random Forest Regressor. Therefore, we eliminate SVR as well.

Both Ridge Regression and Random Forest perform well and can be improved even further by tuning the model parameters.

Between Lasso regression and Stochastic Gradient descent, we see that Lasso performs well on training and gives better R2 score on test. On the other hand, SGD gives a lower Mean Absolute Error. However, Lasso can be effectively tuned by varying the alpha parameter of the model and would result in sparser weight vectors, possibly improving model performance. While SGD regressor can be improved only by increasing the number of iterations or the convergence rate. Therefore, we decide to eliminate SGD regressor leaving us with three models; Ridge Regression, Random Forest Regressor and Lasso Regression.

Final Model Selection:

After this, training was performed on each of the three models using cross-validation approach to select the hyperparameters for each of this model. This was explained in the previous section. The optimal value for each model is:

Ridge Regression: Alpha=50
Lasso Regression: Alpha =0.1
Random Forest Regressor: N_estimator =80, Depth = 50

We are left with three models with the optimal parameter values known. To evaluate which of the three models perform the best, we evaluate the model performance on the validation set and use Mean Absolute error and R2 score as the evaluation metric.

Model	Train Mean Absolute Error	Train R2 Score	Validation Mean Absolute Error	Validation R2 Score
Ridge Regression	51.076	0.4085	52.16	0.41011
Lasso Regression	51.06	0.4073	52.158	0.4087
Random Forest Regressor	13.798	0.9499	37.253	0.66

Analysis of Results:

From the above table we can see that all three models give better R2 scores and lower errors than what was estimated during Pre-training. This is because of parameter tuning. This is especially true for Random forest were the preliminary validation error was 50.27 and the final error on validation set is 37.253. The R2 scores are also much higher.

Among the three models, we see that Random Forest Regressor is the best model. Both Ridge and Lasso regression gives similar Error and R2 score.

This is because Random Forest is more capable of representing complex models. Ridge and Lasso regression assume the target function to be linear which may not be the best representation of this model. The model used to represent the data is quite complex which is best represented by a Random Forest regressor.

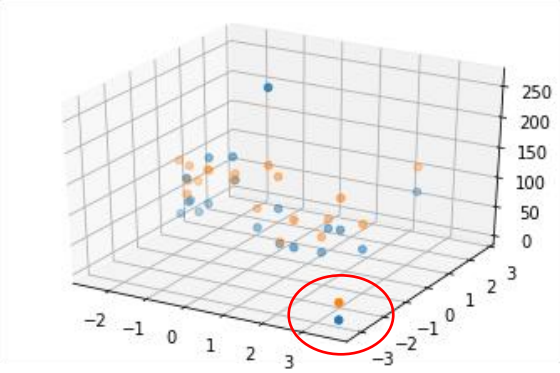
Best Model: Random Forest Regressor with N_estimator = 80, Depth = 50.

We will use this model to evaluate the test dataset.

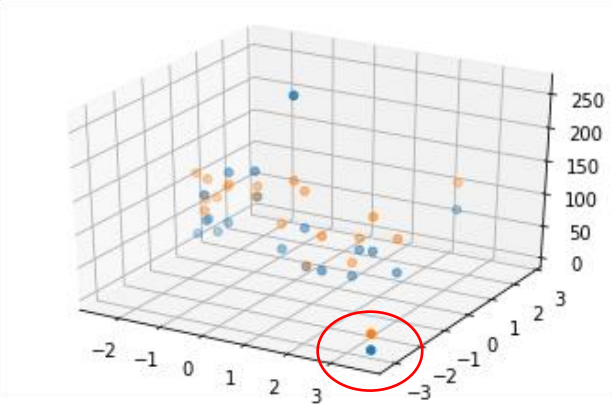
Plot:

Below are three plots that represent the actual and predicted PM2.5 values for the three models; Ridge regression, Lasso regression and Random Forest regressor. Here we consider only 20 data points, for easily visualizing the plot. The x and y axis represent the most 2 significant features in the data. This is estimated by performing Principal Component Analysis to get two components. The blue points represent the true output value, while the orange colored points represent the predicted PM2.5 values.

Ridge Regression:



Lasso Regression:



Random Forest regression:

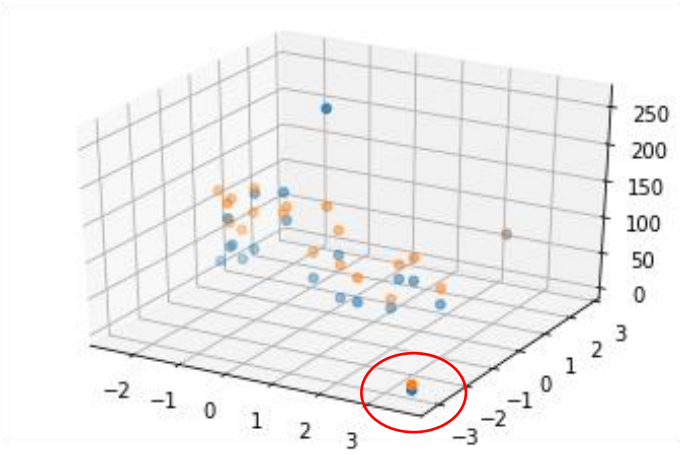


Fig 4. 3D plot of predicted output in terms of 2 salient features

From the plot, Random Forest gives the predicted values closest to the true values. The point in the red circular region indicates this.

The results obtained are as expected. Due to the ability of Random Forest to express complex models it gives the best output prediction and lowest error values compared to Ridge and Lasso Regression.

4. Final Results and Interpretation

The best model, as seen from the above results is Random Forest Regression.

As stated above the optimal parameters are:

Best Model: Random Forest Regressor with N_estimator = 80, Depth = 50.

This is then evaluated on the test set to determine the Final Results of our Regression model.

Model	Train Mean Absolute Error	Train R2 Score	Test Mean Absolute Error	Test R2 Score
Random Forest Regressor	12.809	0.95577	34.314	0.6915

This is the final result of the model. We get a Test error of 34.314 and R2 score of 0.6915.

Out of Sample Performance:

Since the test set contains only a single hypothesis, the out of sample error can be estimated with number of hypothesis M=1.

$$\text{Generalization error} = \sqrt{\frac{1}{2Ntest}\ln\left(\frac{2M}{\delta}\right)}$$

Since Ntest = 9396

Assuming we take δ = 0.05, if we want to predict with probability >= 1- δ =95%

$$\text{Generalization error} = \sqrt{\frac{1}{2*9396}\ln\left(\frac{2}{0.05}\right)} = 0.014$$

Eout <= Ein + 0.014, with probability >=95%

Since we have Ein = 34.314, we will get

Eout<= 34.328

Interpretation:

The Mean Absolute Error is around 34.328. For regression models we, unlike classification, we evaluate performance by estimating the distance of predicted points from the actual point. This metric can be anything from Mean squared error to Mean absolute error. However, the magnitude of the error will depend largely on the magnitude of the data itself. If the magnitude of the data is large, then error is likely to be higher. Therefore, we compare the errors of different models to see which one gives the lowest error and use this as a best model for the data.

Different kinds of models were used to find out the best one. Linear, Ridge and Lasso Regression are all linear models with different Priors. This means that the target function is linear in the features but have different importance for the weights. Ridge and Lasso increase sparsity. We know that many features of the data are categorical and gives us highly sparse features many of which may not be important. Thus, these two methods must outperform Linear regression. This is found to be true.

Since we do not know the underlying model of the data, we must try different model functions to see the best fit. Clearly, the linear models all give the same error and not much improvement is seen even after parameter tuning. This implies that we need to find better models to

represent data. This is why Random Forest performs so well. We have multiple trees in random forest each trying to fit different functions to the data. The aggregate of all model outputs is taken to predict the output. This gives much lower error than assuming any one model function.

One unexpected result obtained was for pretraining using SVR. The SVR also gives freedom to the model to select a complex function that will fit the data. However, it resulted in higher error than linear models.

We might try other methods that could improve the performance such as combine two or more models to determine the output. For instance, if we combined outputs of linear regression and SVR we might get better results than using SVR alone.

5. Contributions of each team member

Individual Project.

6. Summary and conclusions

In conclusion, this project applies key machine learning theories to better fit a dataset where we have no prior information about the underlying model. Normalization of features is key in regression models, to get a good error estimate. If we have features that are not normalized, we might get incorrect error estimate. PCA is also useful when we want to use only the most significant features in our model. Pre-training data is found to be quite useful in doing a preliminary analysis of the data without peeking into our training data. It helped in selecting the pre-processing techniques, feature extraction techniques and help us narrow our hypothesis sets by reducing the number of models to train on. This save us a lot of computation time.

Several regression models were tried and Random forest Regressor was found to be very useful for complex models.

An interesting thing to do next would be to somehow use the missing outputs present in the original data. In this project, these outputs were just removed from the dataset. This might give us useful information about the model. We might perform a Semi-supervised approach to determine the missing outputs and use these to determine the model of our data.

The data also contains time (year, day, month etc.) as features. Instead of considering them as categorical data, another approach would be to use them as a time series and extract key features like mean, standard deviation etc. of this time series data. This could give us more information about the model to use for regression.

7. 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.
- https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data#
- https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression
- Machine Learning: A Probabilistic Perspective by Kevin P. Murphy
- Learning from Data by Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin