Analysis And Prediction Of Delhi Climate Using Machine Learning

Gaurav Lodhi
MT19063

Kaamran Khan
MT19064

Vedant Desai
MT19074

I. Introduction

Delhi is infamous for air pollution and erratic weather conditions like fog, hail, smog, etc. It causes huge damage to life and resources in significant sectors like agriculture, transport system, infrastructure, etc. And the climatic conditions are only going to get worse in the near future. Thus, there is an urgent need for accurate climate analysis and prediction. The traditional weather forecasting systems have traditionally relied on physical simulated models which are resource intensive, complex and require lot of computation power. We take a datacentric approach using machine learning algorithms to derive the solutions which can be run on less resource intensive and low-cost computing system. So, the main goal of our project is to provide quick and accurate predictions which can be used in our day-to-day life for (i) classification of extreme weather conditions (ii) regressive forecasting of AQI (Air Quality Index).

II. Related Work

Since we have created a custom dataset by combining 2 different datasets, no related work will be found on this dataset. But a considerable amount of work has been done separately for Delhi weather data and Delhi pollution data. Some of them are :- "Emission estimates and trends (1990–2000) for megacity Delhi and implications." Gurjar, B.R., Aardenne, J.A. van, Lelieveld, J., Mohan, M. (2004). "Assessment of traffic–generated gaseous and particulate matter emissions and trends over Delhi (2000–2010). Sindhwani, R., Goyal, P. (2014)", "Numerical methods in weather prediction." Elsevier, G. Marchuk. 2012. "Estimation of Air Pollution in Delhi Using Machine Learning Techniques" Srivastava, Chavi and Singh, Shyamli and Singh, Amit (2018).

III. Dataset and Evaluation

We are using two dataset as follows-

A. Dellhi weather dataset

B. Delhi air quality data

Delhi weather dataset is a time-series hourly data that contains 100990 samples along with 20 features ranging from 1997-2016 Delhi air quality dataset is a time-series daily data that contains 8845 samples along with 13 features ranging from 1995-2015. We merged both the datasets based on hour by considering pollution data of that day as mean and then adding variance based on the data of previous twenty days to get a resultant dataset with 57561 samples combining both their features. The dataset was divided into training, validation and test set in the ratio 75:15:10 respectively. Upon close inspection we removed certain unimportant features like "agency", etc. We faced issues due to NA values in some columns. So we removed those features which contained more than 15,000 NA values. For the remaining features with NA values we replaced NA with the mean value of that particular feature. We created some new features such as "ni", "si", "spi", "rpi" based on "NO2", "SO2", "SPM", "RSPM" respectively in order to create feature AQI (Air Quality Index). Before sending the data to train we performed one-hot encoding on categorical features. So our final dataset contains 57561 samples along with 83 features. For our regression problem, we will use root mean square error and for our classification problem, we will use accuracy score. Since the data for extreme weather conditions is skewed, we will use precision, recall and roc as our evaluation metric.

Fig. 1. Heatmap for all the features

IV. Methodology

A. AQI Prediction:

In the first part of our case study, we aim to derive machine learning solutions for the regression problem of predicting and forecasting AQI. Now our starting goal was to get the base error rate using Linear Regression. Linear regression gave RMSE of around 42 for both training and testing. RMSE was relatively high indicating a need for enhanced model training. But from this we were assured that at least our training is not erroneous and we can only better the performance.

Since both train and test accuracies were similar, the case of overfitting was less likely to occur. But to confirm this we applied Ridge and Lasso regression on the data. Both gave RMSE around 42 verifying the absence of overfitting. But the case of underfitting can't be neglected since we inferred from the results that the data is not linearly separable. Thus, we moved ahead in applying more complex models like SVM and tree algorithms like random forest. On SVM, we selected the hyperparameters using GridSearchCV. SVM definitely reduced the error rate but was found to take a lot of time in training. This deferred from our goal of quick prediction, so we moved on to apply tree algorithms. As expected, the error rate dropped considerably. To increase the complexity of the training model we then started working on applying neural networks. We applied MLP classifier in which the important hyper-parameters were picked by hand-tuning it through trial and error experiments performed from inferring our data knowledge. This resulted in MLP classifier performing the best on our dataset with testing RMSE going below 5 for hidden layer architectures of (4,4,4) and (5,5). For architectures more complex then these i.e. with more neurons or hidden layers, the variance of the model prediction started to shoot up indicating the emergence of overfitting. While models less complex that these gave results aligning near-linear models. Next, we introduced a new feature named "AQI predicted". For each sample, this feature contained mean values for AQI of the next day. Now, this feature formed the label to be predicted. So, our goal was to train models that accurately forecast the mean AQI levels for the future. Similar to how we approached the above problem, we went from applying linear regression all the way to complex neural nets. But the testing errors of every model were in no way near our expected standards. As RandomForests algorithm performed best with training RMSE around 25 and testing around 60. So, we were encountered with both overfitting and underfitting. Now we had to find a better solution to encounter this problem. So now we performed model analysis using data. Here, we segregated the training data to 10 days, 1, 6 months, 1, 2, 4 years. For each of these training data, the next 20 percent were taken as test data. We only applied our best models derived from above on this new data. Interesting observations were made from model evaluation of this data. It was seen that cross-validation test rmse rates halved for training data with fewer days (like 10days, 30 days). But it started to gradually increase as the number of days increased. From this, we derived that data in close proximity to the prediction generated the greatest importance. Next, we observed that after doing a one-hot encoding of categorical data, the number of features became. We were already aware that there were many unimportant features with respect to AQI prediction. Hand-picking and removing features could prove troublesome, thus using sklearn classes like ExtraTreeRegressor and Random Forest Regressor, we derived feature importance of each feature with respect to certain labels. From this, we selected only those set of features which were deemed important while the rest were removed from training. Then we applied all the above-used algorithms on this data for training. The output of this expedient was that cross-validation testing rmse was almost the same or even better in some cases than the models with all the features. Thus, we can safely remove the other unimportant features and so we reduced the dataset from features to 10 features. Then we performed another analysis where we separated the data month wise to 'Winter (Nov, Dec, Jan, Feb), Summer(Mar, Apr, May, Jun) and Autumn(July, Aug, Sep, Oct). And we tried to forecast AQI separately for each of these data. Testing rmse for Autumn was found around 32 while that of Summer and Winter were 42 and 46 respectively. From this, we can derive that in winter months the range is quite large thus proving difficult to predict as variance increases.

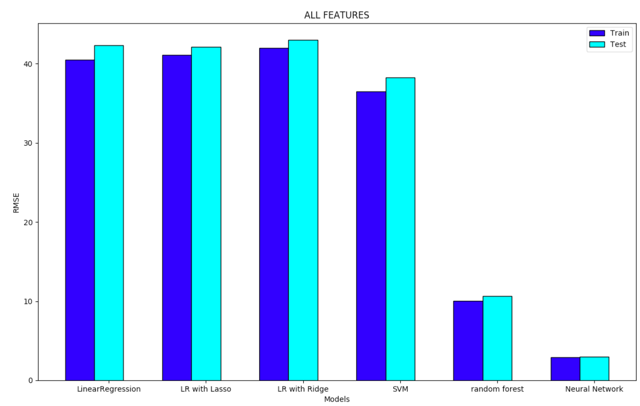

Fig. 2.

B.

In this part of our case study, we aim to correctly predict extreme weather conditions such as 'Thunderstorm', 'Heavy Smog', '. So, for this first, we prepared the data in the following manner. In the 'conds' column (Indicating weather conditions) we replaced those samples with extreme weather with 1 while the rest with 0. Now, this column will be our label column on which we will perform classification. Since this was skewed data consisting 4:1 ratio of normal weather to extreme weather, we used evaluation metrics mentioned in Dataset and evaluation. For base case Logistic regression training accuracy- 83.9, testing acc-83.3, test precision-75.9, test recall-41.2. As mentioned, before we gave more importance to precision and recall values rather than accuracy for model analysis and evaluation. Next, we performed variants of naïve Bayes which performed very poorly on this data. Then KNN algorithm was applied, it performed quite better comparatively. But since it is a non-parametric algo it took much more time than other algorithms. Next, we applied the decision tree algorithm, like most tree algorithms it gave 100 percent accuracy for training data. But for testing data, the difference was large indicating overfitting. Overfitting was also encountered in other algorithms like neural nets, random forests. So, we applied multiple techniques to encounter it. Like limiting max tree depth, limiting max iterations, applying regularization and also applying feature reduction through the feature importance derivation technique mentioned in regression problem reducing to top 10 features. Also, we performed hyper-parameter tuning using gridsearchCV. The best algorithm expectantly was found to be the random forest algorithm. Since if the decision tree performs well, the random forest is likely to perform well since it is an ensemble of multiple decision trees. Then we performed further analysis on the misclassified samples. After visualizing multiple plots, we found out that all the misclassified samples were from the data before the year 2003 and no sample was misclassified after that year. While the extreme weather data was spread overall uniformly. So next we sent we only the data before 2003 for training and testing on our best model. And precision, recall, and accuracy values didn't drop on this data. Thus, overall after this, the total number of misclassified samples dropped considerably. The data was skewed with a 4:1 ratio, so to perform better evaluation we performed stratified sampling by under-sampling the data to the consistent ratio. Also for each separated season (Summer, Winter, Autumn), we plotted the count extreme weather conditions corresponding to each season. At last, we performed our own user-defined bagging on the models. Here for each test sample, we performed majority voting from all the correctly trained models and generated the prediction values.

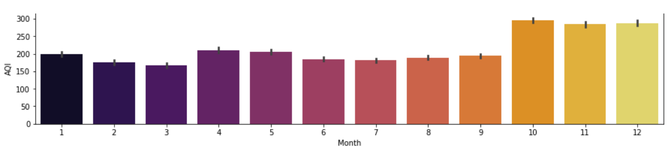

Fig. 3. Barplot of AQI corresponding to months

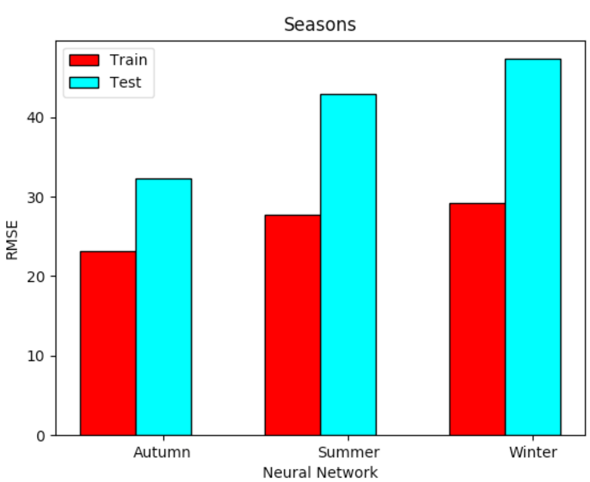

Fig. 4. Barplot of AQI corresponding to months

V. RESULTS

Below are all the model training and testing results obtained as described in the methodology section. Also the inferences

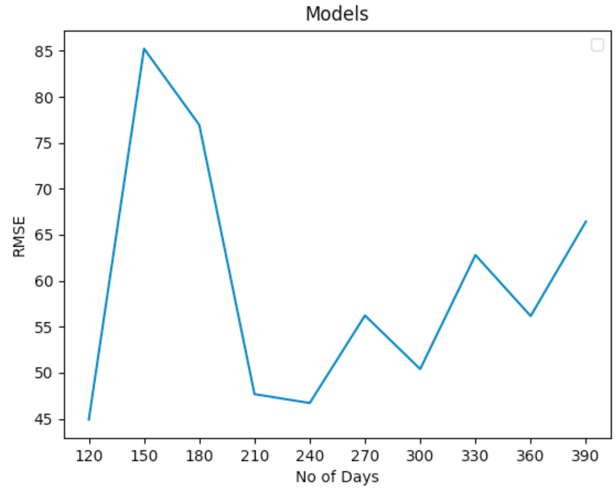

Fig. 5. Barplot of AQI corresponding to months

to the results are mentioned in that section.

	Autumn	Summer	Winter
Neural Network	Train=23.22 Test=32.35	Train=27.77 Test=42.99	Train=29.23 Test=47.34

Fig. 6.

Neural Network	All Features	Important Features	With Domain Knowledge	Without Other features of pollution
AQI	Train=3.78 Test= 3.0	Train=3.72 Test= 3.09	Train=53.5 Test=54.98	Train=65.78 Test=68.01
Future AQI	Train=28.45 Test=68.1	Train=27.56 Test=67.98	Train=103.23 Test=128.3	Train=122.2 Test=130.67

Fig. 7.

	Autumn	Summer	Winter
Neural Network	Train=23.22 Test=32.35	Train=27.77 Test=42.99	Train=29.23 Test=47.34

Fig. 8.

VI. CONTRIBUTIONS

A. Deliverables:

Gaurav was supposed to work on complex models on classification problems and Kaamraan ,Vedant were supposed to work on applying deep learning models on regression problems and producing results from prediction of extreme weather conditions. All the promised deliverables are deliver but not by specific member but by working as a team.

B. Individividual Contribution:

1. Gaurav has applied linear regression model and Random Forest model on predicting AQI,Future AQI,AQI by using domain knowledge .He also applied KNN and Logistic regression model on predicting extreme weather conditions.Apart from this he also apllied bagging. Functions

like- logisticExtremeWeatherConditions(data,y), logisticExtremeWeatherConditionsFeatureimportance(data,y), KNNExtremeweather(data,y),KNNExtremeFeatureimportance(data,y), bagging(data,y), AQIFutureRF(data, y.AQIpredicted),etc.

2. Kaamraan has applied linear regression with lasso model and Deep neural network model on predicting AQI,Future AQI,AQI by using domain knowledge . He also applied Decision tree and naïve bayes model on predicting extreme weather conditions. Apart from this he also predict the extreme weather conditions in different seasons of year. Functions like- DecisionTreeExtremeweather(data,y),naivebayesExtremeweather(data,y) ,naivebayesExtremeweatherFeatureimportance(data,y), DecisionTreeExtremeweatherFeatureimportance(data,y), countweatherclasses(),AQIFutureNN(data,y.AQIpredicted, layer,etc.

3. Vedant has applied Random forest and deep neural network model on predicting extreme weather conditions .He also has applied linear regression with ridge model and SVMon predicting AQI,Future AQI,AQI by using domain knowledge. Apart from this he also identified misclassified classes. Functions like- NNExtremeweather(data,y) NNExtremeweatherFeatureimportance(data,y), RFExtremeweather(data,y) RFExtremeweatherFeatureimportance(data,y), misclassification(),AQISVM(data),etc.

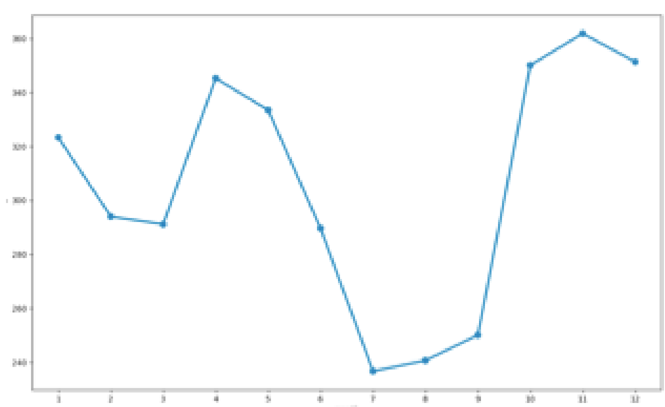

Fig. 9.

For prediction of important continuous variables like ‘temperature’ and ‘AQI’, the design choices we made were to adhere to our starting goal of getting the base error rate by applying simple learning algorithms like LinearRegression. So that we can slowly build up from the base error rate later. Then for column ‘conditions’ having 37 unique values, we have segregated those with extreme weather like ‘tornado’, ‘thunderstorm’, ‘hail’ to ‘1’ and the rest being normal weather as ‘0’. Again, on this we have applied simple classification algorithms like Logistic regression for prediction of extreme weather conditions to get base accuracy. Also, since it is a time-series data we have also applied ARIMA model on our data to predict SO2 levels in a future time duration.

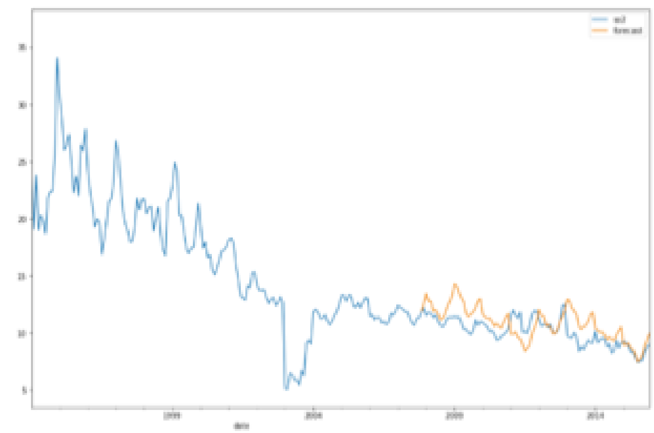

Fig. 10. SO2 prediction using ARIMA model

Despite applying feature selection with the help of EDA, while training we faced problems due to certain unimportant features which were only adding noise and confusing the models. Thus, using sklearn classes like ExtraTreeRegressor and Random Forest Regressor, we derived feature importance of each feature with respect to certain label. From this we selected only those set of features which were deemed important while the rest were removed from training. Along with the above-mentioned predictions our end goal is to develop a prediction model that forecasts all the weather conditions for the next 24 hours, next 3 days, next week alike other commercial weather forecasting services. So, we have performed similar base accuracy and error predictions on a number of labels like ‘wind-direction’, ‘wind-speed’, ‘humidity, etc. After completing the base case model training, we analysed the results to infer that the data is not linearly separable and so would need a more complex model to properly train it. Also viewing the high error rate of training and testing data, we have to come to a conclusion that the models are under-fitted. But we are optimistic about decreasing the error rates by more feature selection, training a few complex learning algorithms coupled with proper hyper-parameter tuning.

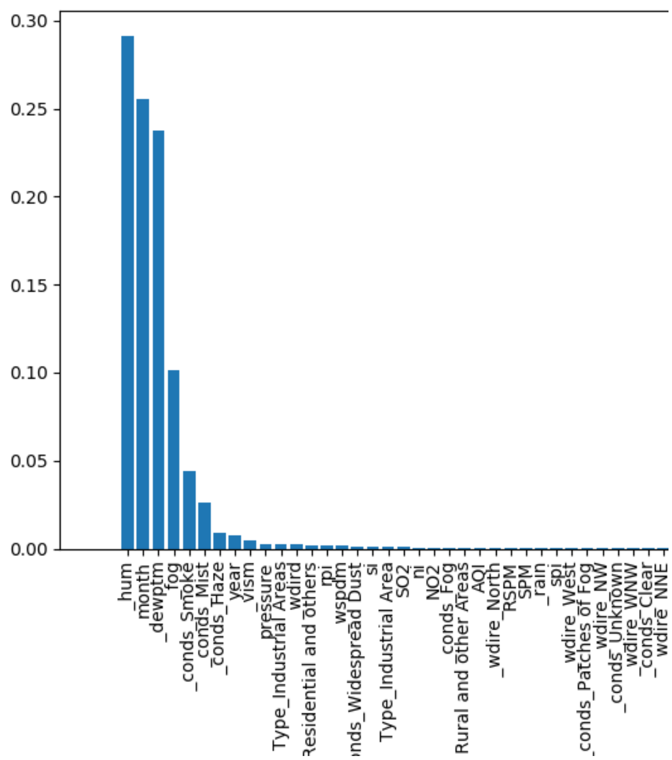

Fig. 11. Label Temperature:- Feature importance vs features