Classification and Prediction of Outdoor Qualities Based on Weather Parameters

Eric Liu (eliu7), Felix Merk (fmerk), Yijun Shao (yshao3), Ying Su (ysu1)

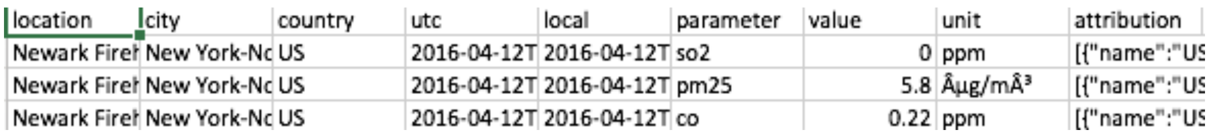
1. Abstract

Fine particulate pollution in the air is linked to severe cardiovascular and respiratory diseases. Air pollution is therefore closely linked to human and animal health. Modeling levels of air pollution can mitigate the adverse effects of high pollutant levels, but most current modeling only predicts at low resolution over long periods of time. Predicting air pollution in the immediate future allows people to minimize their contact to airborne pollutants. Thus, we utilized bulk air quality data and historical weather data to model how pollution levels correlate weather conditions.

We cleaned the collected air quality data from several cities, then merged it with historical weather data from Weather Underground. Using this data, we trained several machine learning classifiers: Logistic regression, linear svm, and random forest. From the resulting models, we found that NO_2 and CO were able to be best predicted, while the other pollutants were predictable to a lesser degree. The most important factors for predicting air pollution were found to be wind speed and humidity. These results suggest that there does exist correlation between air quality and weather parameters, and weather parameters can serve as a predictor for air pollution.

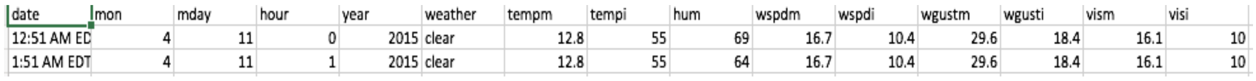
2. Data Collection and Integration

First, we collected the air quality data from https://openaq.org/#/sources and the OpenAQ api. Using the API, we obtained the historical air quality readings for San Francisco, Houston, Boston, and New York. We separated these files by pollutant (NO2, PM25, etc.) and removed duplicates from the data.



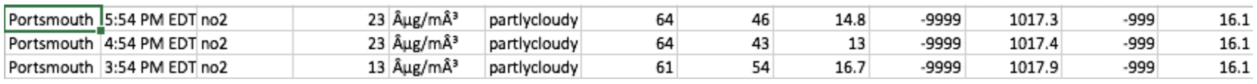
location	city	country	utc	local	parameter	value	unit	attribution
Newark Firet	New York-Nc	US	2016-04-12T	2016-04-12T	so2	0	ppm	[{"name":"US
Newark Firet	New York-Nc	US	2016-04-12T	2016-04-12T	pm25	5.8	Âµg/mÂ³	[{"name":"US
Newark Firet	New York-Nc	US	2016-04-12T	2016-04-12T	co	0.22	ppm	[{"name":"US

Then, we used the Weather Underground API (https://www.wunderground.com) to match up the collected air quality readings with historical weather data. The Weather Underground API returns the parameters weather condition, temperature, humidity, wind-speed, wind-gust, visibility, etc., which we saved into the same row as the pollution readings. "Getweather.py" is our script for getting data from weather API, and generating a csv file for weather data based on city.



date	mon	mday	hour	year	weather	tempm	tempi	hum	wspdm	wspdi	wgustm	wgusti	vism	visi
12:51 AM ED	4	11	0	2015	clear	12.8	55	69	16.7	10.4	29.6	18.4	16.1	10
1:51 AM EDT	4	11	1	2015	clear	12.8	55	64	16.7	10.4	29.6	18.4	16.1	10

After generating the first two files, we then used "merge.py" to merge the two datasets together and put them into a combined csv file based on city and air pollution parameters such as "bostonno2.csv".



Portsmouth	5:54 PM EDT	no2	23	Âµg/mÂ³	partlycloudy	64	46	14.8	-9999	1017.3	-999	16.1
Portsmouth	4:54 PM EDT	no2	23	Âµg/mÂ³	partlycloudy	64	43	13	-9999	1017.4	-999	16.1
Portsmouth	3:54 PM EDT	no2	13	Âµg/mÂ³	partlycloudy	61	54	16.7	-9999	1017.9	-999	16.1

For the primary machine learning component, this was the extent of the data cleaning necessary.

However, when we investigated the relationship between percent change and weather, we found that additional data processing was necessary. After generating the paired weather data, we then parsed the date for each row and sorted on date. We then found the change between each row and divided it by the set average to yield percent change.

3. Hypothesis

Even prior to obtaining the data, we predicted that there was a relationship between relationship between air quality and weather on a day-to-day basis. Our hypothesis was that long term averages of air pollution would primarily be determined by external environmental factors (Hochadel et al.), but deviations from that average on the time scale of a day would be primarily influenced by weather factors.

Before we start to build any model on the relationship between weather features and air qualities using machine learning methods, we first draw the scatterplot matrix to examine the correlation among those weather features as well as the possible correlation between air quality and weather features. Figure 1 shows the scatterplot matrix of correlation between PM2.5 measurements and weather parameter pairs.

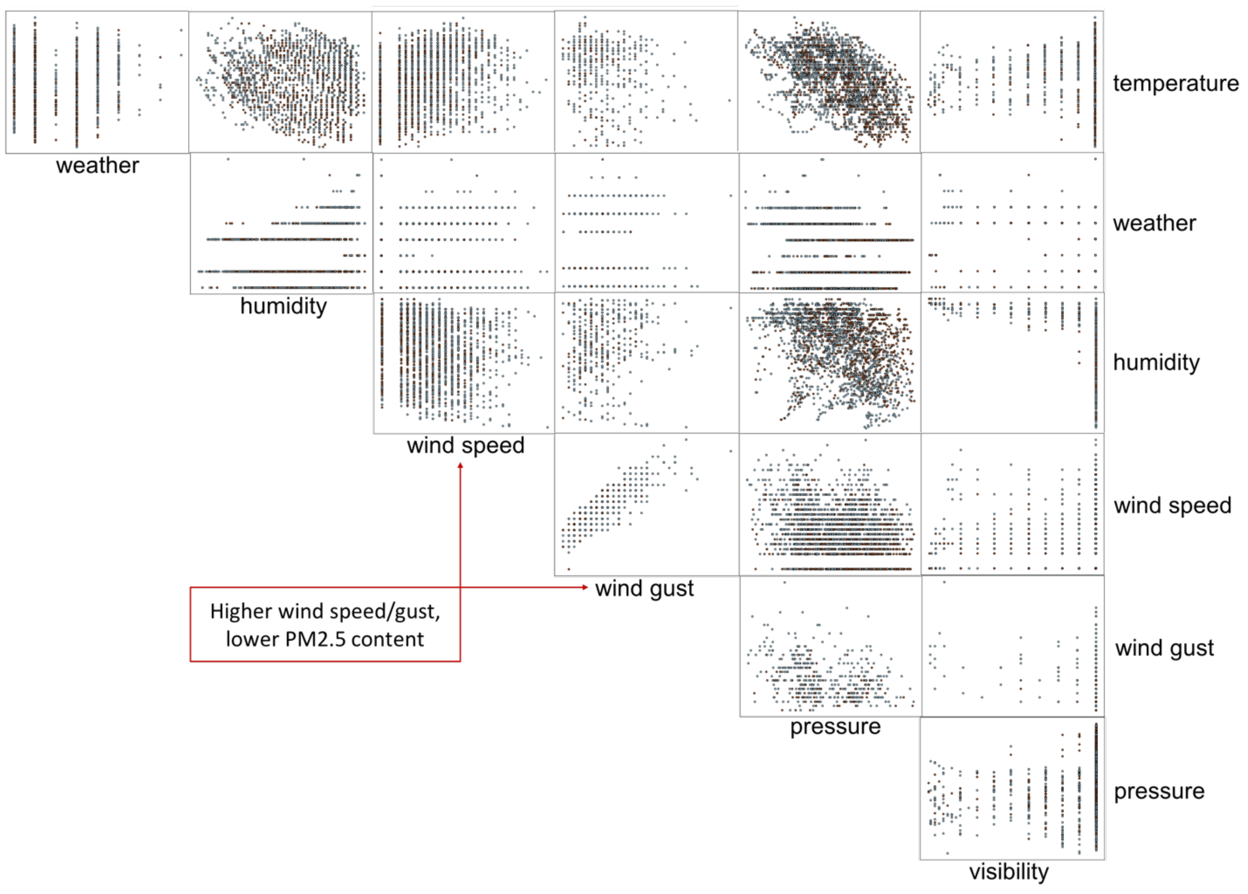


Figure 1. Scatterplot matrix of correlation between PM2.5 concentration and weather parameter pairs. Blue dots represent LOW (< 12 mg/m^3), red dots represent HIGH (> 12 mg/m^3).

It can be seen that except for the wind speed and wind gust pair, there is no obvious correlation between two weather features for most weather feature pairs, for which we can assume that those weather features are independent with each other. With respect to the correlation between PM2.5

measurements (with HIGH and LOW labels) and weather features, we found that the higher the wind speed/gust, the higher probability to find a LOW label (blue dots) than to find a HIGH label (red dots). There is no other particularly strong correlation between PM2.5 and other features. Therefore, we hypothesize that wind speed is an important feature influencing the PM2.5 content in air. Similar trend can also be found for other air qualities, e.g. CO, SO2, NO2, indicating that wind speed is an important feature for most air qualities.

Thus, we aim to verify the hypothesis that air pollution is correlated with weather and further examine the hypothesized strong link between wind speed and air quality. To do so, we employed various machine learning methods to determine the accuracy of prediction.

4. Methodology and Results

For the machine learning part of the project, we used four different methods to build models for the relationship between air quality and weather features: multiple regression, logistic regression, SVM and random forest.

4.1 Multiple Linear Regression

We performed multiple linear regression to model the relationship between a specific air quality (e.g. pm25, CO) and a series of weather variables (i.e. temperature, humidity, wind_speed, pressure, etc.). The L2 regularization and cross validation were used to prevent overfitting.

We first built the multiple linear regression model based on the linear predictor function, and performed the Stochastic Gradient Descent (SGD) to minimize the error term. The features are normalized into the 0~1 range by using the min-max normalization. With all seven features (weather condition, temperature, humidity, wind speed, wind gust, pressure and visibility) included in the model, the accuracy for CO prediction is 49% (with both prediction values and test y values rounded to 1 decimal place). Even with quadratic or cubic terms added to the model, the accuracy is not significantly improved (still below 55%). The accuracy of prediction on other air qualities (e.g. PM2.5, SO_2, NO_2) using multiple linear regression models are even lower, and for PM2.5 predictions, the accuracy is even lower than 5%.

Since it is possible that the air quality is more related to some of the weather features compared to others, we also tried Lasso regression model in sklearn package. Lasso is used to estimate sparse coefficients in cases where a solution with fewer features is preferred. L1 regularization is used in Lasso. With Lasso model, the accuracy for CO prediction (with cross validation number = 5) is below 37%.

The reason for the low accuracy of the multiple regression model might be that the distribution of the air quality values is broad and the variance is big, so it is hard to fit a single curve to those values. Therefore, it might be more feasible to perform classification on the air qualities by labeling each air quality with LOW or HIGH labels.

4.2 Logistic Regression

With the thought of performing classification on the air quality data, we first need to categorize our training data. Taking PM2.5 for example, according to US Environmental Protection Agency (EPA), when the content of pm2.5 in the air is larger than 12 mg/m^3, it is considered to be harmful for human health. So we divide the pm2.5 data into two categories: HIGH (> 12 mg/m^3) and LOW (<= 12 mg/m^3). The PM2.5 data set is based on the PM2.5 measurements in 5 months at Houston, and the base rate of HIGH PM2.5 is about 35%. For the logistic regression model, all weather features are normalized into the 0~1 range by using the min-max normalization.

We first used the SGD method with L2 regularization to implement the logistic regression, and after tuning the parameters (i.e. threshold, α, and λ), we were able to get a training accuracy at 68.3%, and the confusion matrix is as follows:

	True HIGH	True LOW
Predict HIGH	259	139
Predict LOW	525	1170

It can be seen that although the accuracy is a pretty decent number, the recall of this model is very low (about 33%). Since the number of LOW observations is much larger than that of HIGH observations, this data set is unbalanced, and it may cause the fact that the model tries to predict the label that accounts for the majority in the data set, which results in the low recall.

To solve the imbalance problem, we choose to use the LogisticRegressionCV classifier in sklearn.linear_model, and this classifier allows us to adjust weights inversely proportional to class frequencies in the data set, which balances the data. Still using L2 regularization, and setting the cross validation number to 5, the accuracy of the model with all 7 weather features is 63%, and the confusion matrix is as follows:

	True HIGH	True LOW
Predict HIGH	484	494
Predict LOW	300	815

It can be seen that although the accuracy is lower than the first model, the recall increases from 33% to 62%, and the current precision is about 50%.

To validate our original hypothesis that wind speed is an important weather feature for air quality prediction, we would like to compare how each single weather feature influences the air quality separately. Taking the PM2.5 predictions for example, first four models with single features (i.e. weather icon, wind speed, wind gust and temperature) were built, and the accuracies for the four models are: 0.53, 0.61, 0.46 and 0.58, respectively. The reason for the low accuracy of the wind gust model might be that a lot of data has NaN wind gust value, which were treated as 0 during data treatment. The receiver operating characteristic (ROC) plots (Figure 2) of the four models also show that the wind speed model has the largest area under the curve.

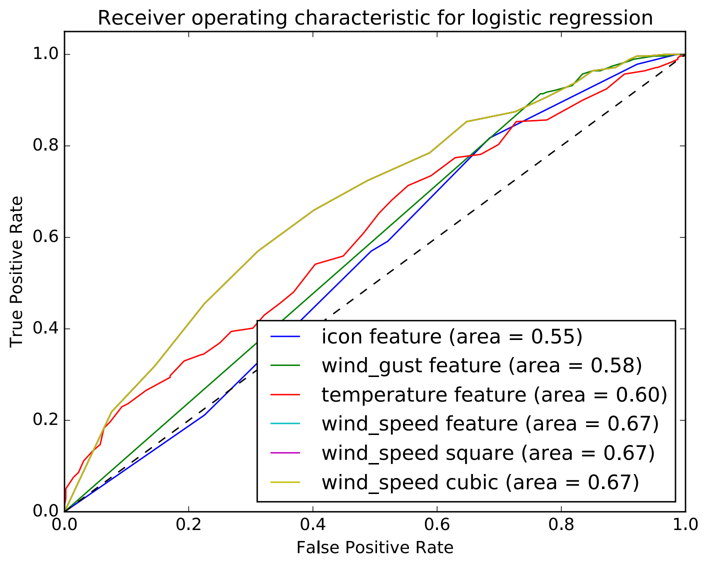


Figure 2. Comparison of ROC plots for models with single feature

To further explore with the wind speed feature using logistic regression, we introduced polynomial terms into the model. The accuracies of the models with quadratic wind speed term and cubic wind speed term are 0.61 and 0.64, respectively, and they are not dramatically improved compared to the model with linear wind speed term (0.61). Also it can be seen from Figure 2 that, the ROC plots of the polynomial models completely superimpose with the linear model, suggesting that the polynomial terms did not obviously improve the model.

The models with multiple (linear) features are also compared, and the accuracies for those models are: 0.63 for model with all 7 features, 0.58 for model with 6 features (7 features except for the wind speed feature), 0.62 for model with 3 features (weather condition, wind speed, wind gust) and 0.61 for model with 2 features (weather icon, wind speed). The ROC curves for those models are shown in Figure 3. The results again indicate that the wind speed is the most important feature among all weather features for the PM2.5 model, and the addition of other features just slightly improves the model accuracy and area under curve (from 0.67 to 0.68).

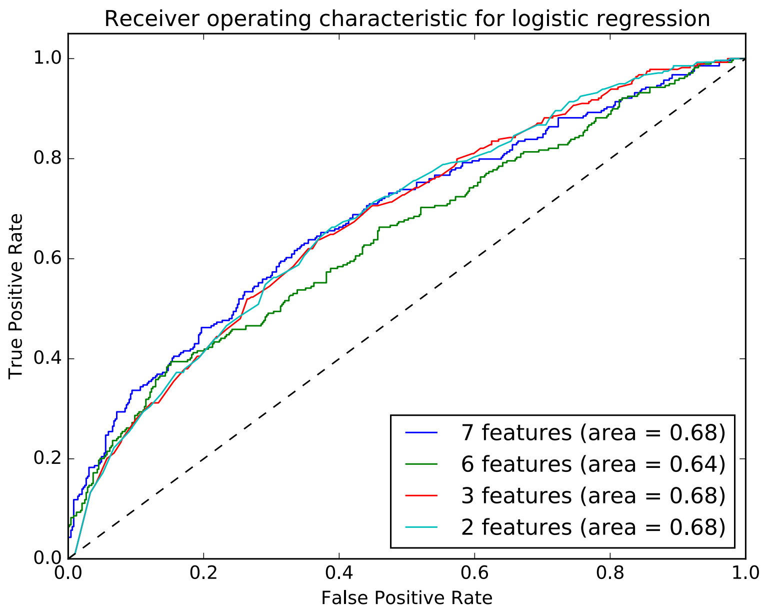
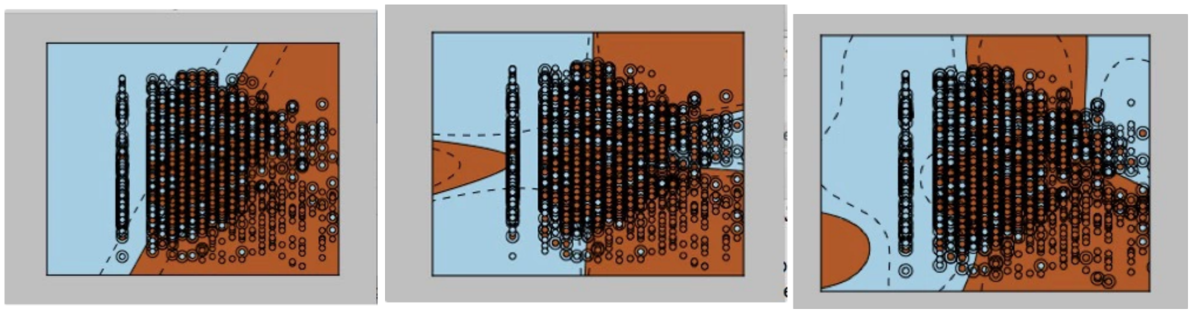


Figure 3. Comparison of ROC plots for models with different number of features

4.3 Support Vector Machine

First of all, we test our model on CO, because it is easy to be labeled. However, when we simply put data to train a multi labels SVM classifier, we cannot get a good classifier for the dataset as the accuracy is roughly 50%.

Therefore, we relabeled the parameters using average air quality value as a boundary. Also, we use kernel function to put features into a higher dimension to make the model more complex and accurate. We found radial basis SVM model gave the most accurate model (below is the boundary of the training data for 2 features)



Linear SVM model Polynomial SVM model Radial Basis SVM model

Figure 4. SVM boundaries of different SVM models

Then, in order to avoid overfitting, we use k-fold Cross-Validation method. In this case, we create 5 folds and test accuracy for each fold and for each pollution, and we choose the model with the highest accuracy among different left out fold. The result is pretty good for O3, NO2 and CO, but a little bit worse for PM25 and PM10. In general, the models seem pretty good. The accuracies of these models are between 70% to 85%. In order to find out if the those classifiers are truly good or not, we plot the ROC curves for different models based on different number of features(below is the ROC curve of O3 SVM model).

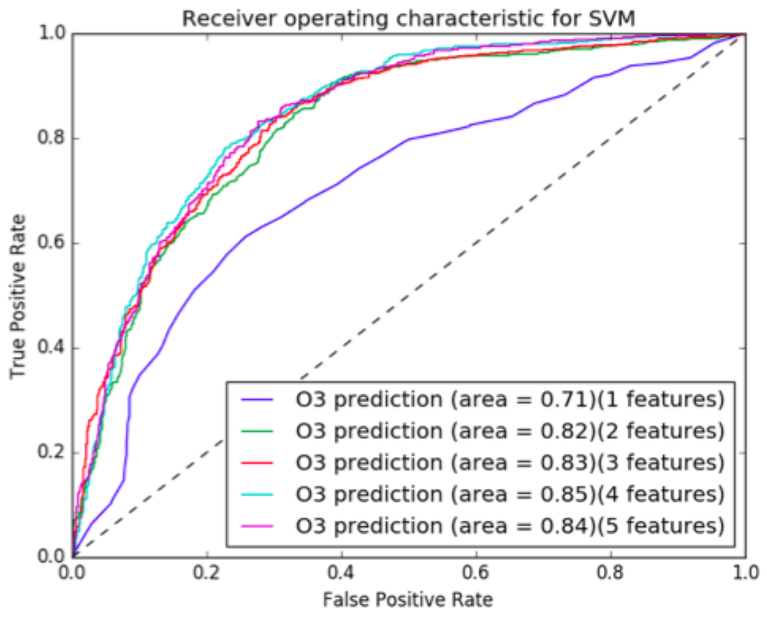


Figure 5. ROC plots for SVM models with different number of features

As we can see above, the model trained by radial basis SVM method is truly a good classifier. However, the biggest weakness for these models is that it can only predict if the value of air quality is above average or not. It cannot give the exact value of the air quality parameters. Therefore, we set SVM method as a backup plan.

4.4 Random Forest

Having gained some insight in the structure and nature of the data already, Regression Forest seemed to have great potential. Not only was the ability to predict actual values enticing, but the ability to handle dependence between features, durability under noise and the capacity to deal with features of different weights made Regression Forest a definite candidate.

Originally we assumed the features to be independent. This simplification makes life easier and is reasonable as we didn't see any excessively strong correlations between the features. However, weather features conceivably are correlated, so, instead of trying to identify these potential correlations directly, which is outside of the scope of our paper, we used the Random Forest model to make the question a non-issue.

Random Forest is more robust under noise than adaboost (Breiman 16) and generally handles noise very well. Weather in general is extremely noisy: most weather data (unlike climate data) is considered de-facto random, more precisely, even though the system is deterministic, small changes in initially input give vastly different outputs: the system is chaotic (Lorenz). Pollution levels also have high variances from our own tests. Clearly, this doesn't mean that there necessarily is a high level of noise for the correlation: weather and pollution could be so strongly correlated that random variations in one directly are reflected in the other, but the data we are dealing with has high propensity to noise so using a Regression algorithm that can deal with this makes sense.

Different weights between features is probably the most obvious concern in this case. From our previous algorithms and also simply from looking at the graphs, certain features, such as wind speed matter a lot while others, such as visibility, don't. Below is a pie chart showing how the Regression Forest deals with this for NO2:

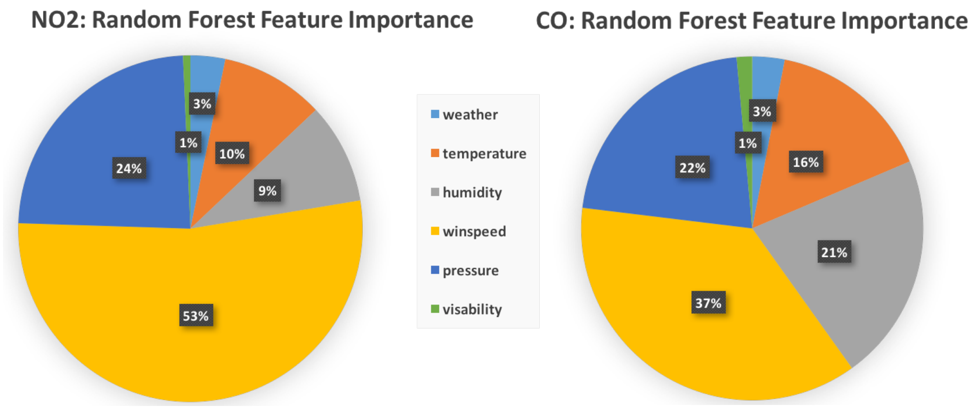


Figure 6. Importance of each weather parameter to the air qualities (NO2 and CO content in air)

Note how Windspeed carries almost 50% of the weight. Models such as Naive Bayes, clearly can't handle this well.

The final Regression Forest algorithm didn't need too much fine tuning, though it needed some, but a lot of work went into testing different Random Forests, cleaning the data, and interpreting results. Originally we used a modified Random Forest that split the data based on homogeneity. This is how one builds a standard classification tree. This generally is meant to be used for discrete predictions. As we had enough data this works fine, but we soon realized that a Regression Forest was the correct route. Regression Forests work by calculating the SSE (sum of squared errors) at each split point between predictions and actual values. This SSE is then minimized. This way we are making continuous predictions instead of discrete ones. Leafs return distributions for a continuous output variable.

The nice thing about Regression Forest is that it does a lot of fine tuning itself, but the not so nice part is that it is somewhat computationally intensive. Some optimization actually went into this. Firstly, we split the algorithm over multiple cores. Luckily the scikit ensemble library contained the necessary functions for this already, so this boiled down to telling the algorithm to use all 4 logical cores (2 physical cores hyperthreaded). Other then that, we ran the data on subsets and pre-cleaned to an extent so that when running the same forest multiple times we didn't spend the computation time cleaning every iteration. Computation time and memory became an even larger issue when we gained access to the entire pollution data set in the form of a single file, which was solved by taking subsets to be looked at.

Data cleaning for Random Forest had some unique aspects. Weather was converted to numerals: weather in the data was signified by strings such as "mostlycloudy" or "hazy" or "tstorms." These were converted into numbers using a purposely written excel function. Excel was used at this point because we were working with smallish data sets at this point (on average 1800 readings) and allowed for very quick turn around time for throwing up visualizations and calculating things like variance and mean in order to understand what was going on. The weakness of Random Forests of being mostly a black box was thereby mitigated through excel. This insight turned out to be crucial. Originally, having set up Random Forest, we got a Forest that perfectly predicted pollution levels. This of course seemed impossible, and it was. It turned out that the dataset contained duplicate readings. We had manually split the data into a training and testing set and it turned out that the original data set contained each reading on average three times. We still don't know why, but this was the case multiple times and was very easy to fix and wouldn't have been easy to find without excel. Other cleaning was fairly standard, such as cleaning up null values and splitting the features from the pollution values.

Interpreting results was also interesting with Random Forest. We hand wrote some simple algorithms that we threw at it first to see where we stood. We calculated things like variance, mean and even produced a tsv file with predictions so we could physically look at the numbers and get a feel for them. The R squared value is especially relevant. R squared signifies good a predictor is in relation to the variance in the data. Different pollutants have quite different variances, so R squared allows us to compare the effectiveness of Random Forests on the different pollutants. In the process we calculated mean-squared error and some other parameters.

Results wise Regression Forest exceedingly well with nitrogen dioxide and carbon monoxide while falling somewhat flat with PM25 and ozone. R^2 values were on average .95, .90, .56, .43 respectively. .56 and .43 aren't exactly bad values, clearly weather can predict pollution, but they are comparatively worse. The possible reasons for this are discussed in the conclusion section.

4.5 Percent Change

Due to the results of the other machine learning methods, we found that it was it was difficult to predict pm2.5, O3, and SO2. Thus, according to our hypothesis, we wanted to see if it would be able to predict pollution as a function of change instead. Furthermore, it made sense to calculate change as percent change against the average value of the set rather than percent change from the previous value because of the number of small values and zero values for pollution.

In order to accomplish this, the weather/air pollution data was combined on identical dates. Because some of the readings occurred on the same day, all of the pollution readings for one day were averaged together in order to make sure that the units of change/unit time were the same for each datapoint. Then, the entire file was sorted on the date which was parsed to a Python 'datetime' object, and each row was then given an additional 'change' parameter representing the difference between the current row and the previous row. Each of the change objects was then divided by the average of every pollution reading in the dataset in order to yield percent change.

We then performed machine learning on this data using a random forest with the same parameters as the previous model. Although this model was highly accurate for each individual dataset (R^2 ~ 0.97 for pm2.5) it was entirely inaccurate when we attempted to extend it to data from different cities. This was probably due in large part to the small size of the data, which was unavoidable because the majority of the air quality readings were restricted to the past few months. Though this was not as much of an issue for the first set of models due to the fact that multiple readings occurred per day, it caused the size of the data for the percent change calculation to be less than two hundred data points. This would definitely cause overfitting to occur, especially with the random forest, which explains why it was so inaccurate when it came to additional data.

5. Conclusions

Our results suggest that there is a correlation between air quality and weather, and that certain weather parameters can be used to predict air quality in the near future. Our results imply that the original hypothesis is true, and that the correlation we saw between wind speed and air quality also exists. The most important factor in predicting PM2.5, PM10, CO and NO2 according to the machine learning models is wind speed, while humidity was the most important factor for O3. These were the expected results, given the original correlation matrix. Ozone is most likely an outlier because of the close correlation between weather conditions tied to precipitation and ozone. Given that humidity is the parameter most closely tied to precipitation, it makes sense that humidity would be the strongest predictor for ozone levels.

6. Challenges and Future Direction

At the beginning, we had an issue with gathering the weather data. Due to restrictions of the Weather Underground API limiting the number of calls/minute, it would be extremely time consuming to pair all of the air quality readings with weather. However, we found that it was possible to avoid this issue temporarily by making a single API call for each day and accessing the per hour readings within the response JSON.

Another issue was the previously noted one of fitting a model for the pollutants pm2.5, O3, and SO2. We found that the models generated for these pollutants had a lesser degree of accuracy, and attempted to see if calculating change from day to day would be allow for a greater degree of accuracy in predicting the amount of each pollutant. However, it turned out that the collected data previously used to generate models was insufficient after being averaged per day. This was not enough data to generate an accurate model, and caused overfitting to occur.

We hoped to resolve this issue by using the entirety of the bulk air quality data from OpenAQ. We were able to obtain this data by contacting the OpenAQ team, who sent us the entirety of the collected historical air quality. However, due to the size of the file and time constraints, we were unable to calculate a model for percent change on the entire dataset.

However, given time in the future, we would definitely test both sets of machine learning models on the bulk dataset. Another future direction to explore is increasing the usability of the web app. Though we currently predict the air quality for a specific input location, we could also store our predictions and test them against the accuracy of actual air quality readings. Both of these areas would allow us to refine our model and increase the accuracy of prediction.

7. Deliverables

At this point, we have completed about 115% of the work proposed in our proposal, including the visualization of data, machine learning model building and analysis, data prediction and implementation of a webpage allowing interaction with user and providing air quality prediction in the next few days.

8. References:

Breiman, L. (2001). Random Forests. Berkeley Statistics Department, 16.

Hochadel, M., et al. (2006). Predicting long-term average concentrations of traffic-related air pollutants using GIS-based information.

Lorenz, E. (1962). Deterministic Nonperiodic Flow. MIT, AMS. Online 2010.

Petzoldt, K. et al. (1994). Correlation between stratospheric temperature, total ozone, and tropospheric weather systems.