**Brief synopsis of the problem you are trying to solve.**
Air pollution and air quality have been crucial topics to solve all around the world due to the increase in the use of fossil fuels and natural gasses. Exposure to air pollutants can cause health problems ranging from respiratory and cardiovascular diseases to cancer and premature deaths. Recognising patterns with regard to the air quality will help us target this issue.

**Brief description of dataset.**
This dataset contains air quality surveillance data from different locations in New York City. The data span across time, location, and type of air quality measurement. The different types of air quality measurements include nitrogen dioxide concentration, fine particulate concentration, ozone concentration, annual vehicle miles traveled and others.

**Link to dataset:**
CSV File:
https://data.cityofnewyork.us/api/views/c3uy-2p5r/rows.csv?accessType=DOWNLOAD
Data.gov link:
https://catalog.data.gov/dataset/air-quality

**1) Describe steps that you already took to clean and randomize the data.**
There are different types of measurements in the Name column though such as Ozone, Nitrogen Dioxide, Fine Particles, and even Asthma visits. We decided to focus on the rows that were only related to air quality measurements, and we split up the entire dataframe by the different measurements of air quality. The three we are analyzing are Ozone, Nitrogen Dioxide, and Fine Particles. We removed the Message column because they were all empty. We removed GeoPlaceName and GeoJoinID because we decided that the GeoTypeName was a more general classification of the location. We removed UniqueID because it was a unique number for every observation, which is not necessary for our analysis. We removed IndicatorID because it was redundant with the Name column. We removed TimePeriod because it was also redundant with the Start_Date column. All the NaN values were in the Message column, so removing the Message column removed all our NaN values, which is why our percentage of NaN values is zero.

**2) You may need to iterate on some of these steps. You may need to do more pre-processing of the data. For example, when using one of the ML apps, you may find that one or more of the features are not in the required format. For example, some algorithms allow categorical data, but some require only numbers. To address that, you may need to use one-hot encoding or binning. You may also need to normalize the data. State whether or not any of these pre-processing steps is necessary, and if so show the code that you used and the results.**
We realized that the Start_Date column was “unsuitable” for the regression analysis app. However, we found that the dates were all datetime objects. To overcome this, we decided to express time as the number of years since the earliest date. To find the earliest date, we used the min function:

```
>> min(air.Start_Date.Year)

ans =

        2005
```

We then made a new dataset which included only the rows that were in 2005. This is called only05. In this dataset, it was clear that the earliest date in 2005 was January 1, 2005.

```
>> only05 = air(find(air.Start_Date.Year == 2005), :)

  Start_Date      I
  ____________    -

  01-Jan-2005
  01-Jan-2005
  01-Jan-2005
  01-Jan-2005
  01-Jan-2005

       :

  01-Jan-2005
  01-Jan-2005
  01-Jan-2005
  01-Jan-2005
  01-Jan-2005
```

Now that we knew the earliest date, we added a new column to our dataset called “yrs_since_jan1_2005” and calculated the number of years using the datetime objects in the Start_Date columns.

```
air.yrs_since_jan1_2005=air.Start_Date.Year-2005 + (air.Start_Date.Month-1)/12+air.Start_Date.Day/30;
```

From there, we decided that the Start_Date column was no longer necessary. In addition, for each individual dataset, the Measure and MeasureInfo columns were exactly the same, so they would be unnecessary during the regression analysis. We deleted all these aforementioned columns. Even though the Name column is the same in each individual dataset, we kept it so we could distinguish the different types of measurements. So after these steps, here is our code and output:

Editor - finalc.m *

```
air = readtable('airqual.csv');
nanrows_bef = any(ismissing(air),2);
percentage_nan_bef = sum(nanrows_bef)/length(nanrows_bef)*100;
fprintf('Percentage of NaN rows before deleting: %%%.2f\n',percentage_nan_bef)

air = removevars(air, {'Message','GeoPlaceName', 'GeoJoinID', 'UniqueID','IndicatorID', 'TimePeriod'});

nanrows_aft = any(ismissing(air),2);
percentage_nan_aft = sum(nanrows_aft)/length(nanrows_aft)*100;
fprintf('Percentage of NaN rows after deleting: %%%.2f\n',percentage_nan_aft)

air.yrs_since_jan1_2005=air.Start_Date.Year-2005 + (air.Start_Date.Month-1)/12+air.Start_Date.Day/30;
air = removevars(air, {'Measure','Start_Date', 'MeasureInfo'});

perm = randperm(16218);
air = air(perm, :);
fine_part = air(find(categorical(air.Name) == 'Fine particles (PM 2.5)'), :);
nit_dio = air(find(categorical(air.Name) == 'Nitrogen dioxide (NO2)'), :);
ozone = air(find(categorical(air.Name) == 'Ozone (O3)'), :);
benzene = air(find(categorical(air.Name) == 'Outdoor Air Toxics - Benzene'), :);
formald = air(find(categorical(air.Name) == 'Outdoor Air Toxics - Formaldehyde'), :);
head(air)
head(fine_part)
```

```
Percentage of NaN rows before deleting: %100.00
Percentage of NaN rows after deleting: %0.00
```

| Name | GeoTypeName | DataValue | yrs_since_jan1_2005 |
|---|---|---|---|
| {'Fine particles (PM 2.5)'} | {'UHF34'} | 10.26 | 8.45 |
| {'Fine particles (PM 2.5)'} | {'UHF42'} | 8.84 | 9.95 |
| {'Fine particles (PM 2.5)'} | {'Borough'} | 10.92 | 4.45 |
| {'Annual vehicle miles traveled'} | {'UHF42'} | 39.6 | 0.033333 |
| {'Nitrogen dioxide (NO2)'} | {'UHF34'} | 22.68 | 10.95 |
| {'Fine particles (PM 2.5)'} | {'CD'} | 10.41 | 12.45 |
| {'Nitrogen dioxide (NO2)'} | {'UHF34'} | 13.06 | 14.45 |
| {'Fine particles (PM 2.5)'} | {'UHF42'} | 8.6 | 7.95 |

| Name | GeoTypeName | DataValue | yrs_since_jan1_2005 |
|---|---|---|---|
| {'Fine particles (PM 2.5)'} | {'UHF34' } | 10.26 | 8.45 |
| {'Fine particles (PM 2.5)'} | {'UHF42' } | 8.84 | 9.95 |
| {'Fine particles (PM 2.5)'} | {'Borough'} | 10.92 | 4.45 |
| {'Fine particles (PM 2.5)'} | {'CD' } | 10.41 | 12.45 |
| {'Fine particles (PM 2.5)'} | {'UHF42' } | 8.6 | 7.95 |
| {'Fine particles (PM 2.5)'} | {'CD' } | 10.25 | 10.45 |
| {'Fine particles (PM 2.5)'} | {'UHF42' } | 8.11 | 12.033 |
| {'Fine particles (PM 2.5)'} | {'UHF34' } | 11.5 | 3.95 |

To summarize, our predictor variables are GeoTypeName and yrs_since_jan1_2005. Our response variable is DataValue. Since we are doing this for multiple different types of air quality measurements, we decided to keep the Name as an identifier for convenience.

**3) Run some statistical analyses of the data, using functions such as mean and median. Do you have any outliers? If so, decide how to address that. Determine some correlation coefficients.**

For statistical analysis, we used the mean, median and std function for each of the three datasets. The following code shows how we displayed the three values of each of the datasets.

```
fine_part_mean = mean(fine_part.DataValue);
fine_part_median = median(fine_part.DataValue);
fine_part_std = std(fine_part.DataValue);
disp('For fine particles:')
fprintf('Mean: %f\nMedian: %f\nStandard Deviation: %f\n\n',fine_part_mean,fine_part_median,fine_part_std)

nit_dio_mean = mean(nit_dio.DataValue);
nit_dio_median = median(nit_dio.DataValue);
nit_dio_std = std(nit_dio.DataValue);
disp('For nitrogen dioxide:')
fprintf('Mean: %f\nMedian: %f\nStandard Deviation: %f\n\n',nit_dio_mean,nit_dio_median,nit_dio_std)

ozone_mean = mean(ozone.DataValue);
ozone_median = median(ozone.DataValue);
ozone_std = std(ozone.DataValue);
disp('For ozone:')
fprintf('Mean: %f\nMedian: %f\nStandard Deviation: %f\n\n',ozone_mean,ozone_median,ozone_std)
```

```
For fine particles:
Mean: 9.355528
Median: 9.080000
Standard Deviation: 2.041713

For nitrogen dioxide:
Mean: 20.960244
Median: 20.690000
Standard Deviation: 6.278526

For ozone:
Mean: 30.082700
Median: 30.340000
Standard Deviation: 3.244355
```

| | Mean | Median |
|---|---|---|
| **Fine Particles** | 9.3555 | 9.08 |
| **Nitrogen Dioxide** | 20.96 | 20.69 |
| **Ozone** | 30.083 | 30.34 |

Our three datasets all do have outliers. We decided to remove the outliers before running our regression. The following code shows how we did this for all three of our datasets.

```
fine_part = air(find(categorical(air.Name) == 'Fine particles (PM 2.5)'), :);
out_fine_part = isoutlier(fine_part.DataValue);

nit_dio = air(find(categorical(air.Name) == 'Nitrogen dioxide (NO2)'), :);
out_nit_dio = isoutlier(nit_dio.DataValue);

ozone = air(find(categorical(air.Name) == 'Ozone (O3)'), :);
out_ozone = isoutlier(ozone.DataValue);

fine_part = fine_part(~out_fine_part,  :);
nit_dio = nit_dio(~out_nit_dio, :);
ozone = ozone(out_ozone, :);
```

To confirm that each dataset had outliers, we used the sum function on the logical vectors created using the isoutlier function:

```
>> sum(out_fine_part)

ans =

    38

>> sum(out_nit_dio)

ans =

    22

>> sum(out_ozone)

ans =

    35
```

To find the correlation coefficient of the datasets, we used the corrcoef function. The coefficient of the Fine Particles dataset is negative, showing that there is a negative correlation between the years since January 1, 2005. This is seen for the dataset of Nitrogen Dioxide too. For both these datasets, we can thus say that the amount of particles decreases as the years pass by. However, for the ozone dataset, the correlation coefficient is almost equal to 0. This shows that the amount of ozone in the air remains constant and does not experience significant change as the years pass by. Additionally, the correlation coefficient for the Fine Particles dataset has an absolute value that is closest to 1, which indicates that the Fine Particles dataset was most closely related to time compared to the nitrogen dioxide and ozone concentrations;.

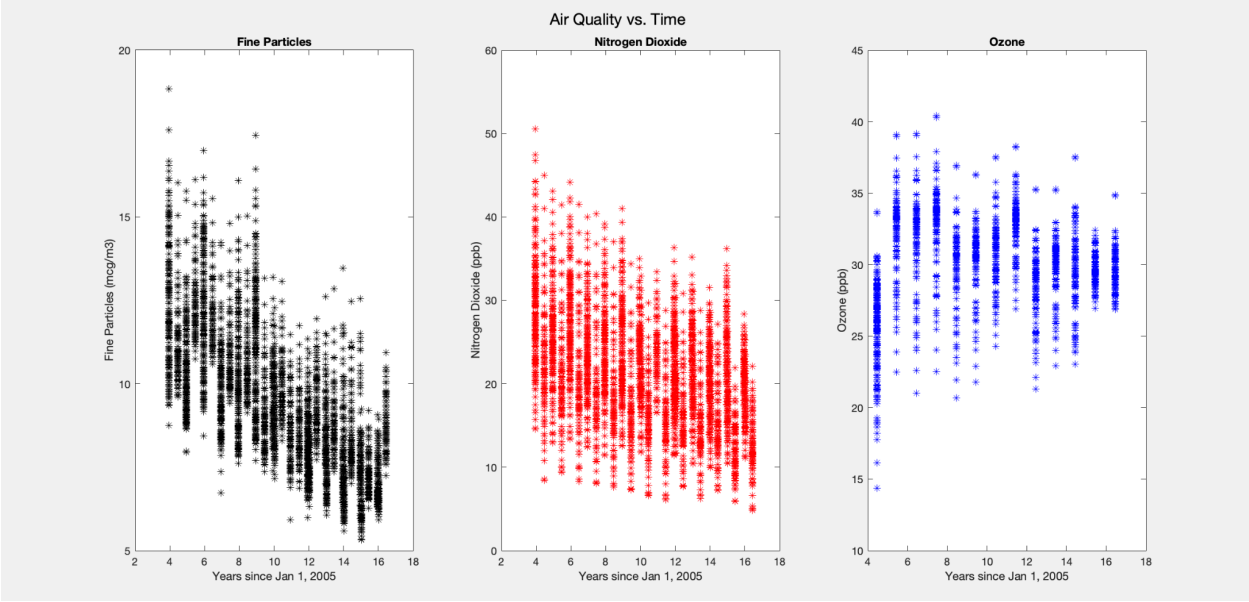
```
fine_part_corrcoef = corrcoef(fine_part.yrs_since_jan1_2005,fine_part.DataValue);
fine_part_corrcoef = fine_part_corrcoef(1,2)

nit_dio_corrcoef = corrcoef(nit_dio.yrs_since_jan1_2005,nit_dio.DataValue);
nit_dio_corrcoef = nit_dio_corrcoef(1,2)

ozone_corrcoef = corrcoef(ozone.yrs_since_jan1_2005,ozone.DataValue);
ozone_corrcoef = ozone_corrcoef(1,2)
```

```
fine_part_corrcoef =

   -0.7317


nit_dio_corrcoef =

   -0.4305


ozone_corrcoef =

    0.0957
```

**4) Produce at least two plots, relevant to the problem you are trying to solve. Make sure that they are well annotated.**

We produced 3 plots under the title - 'Air quality vs. Time'. It shows the relation of the amount of particles in the three datasets changes with time from January 1, 2005.

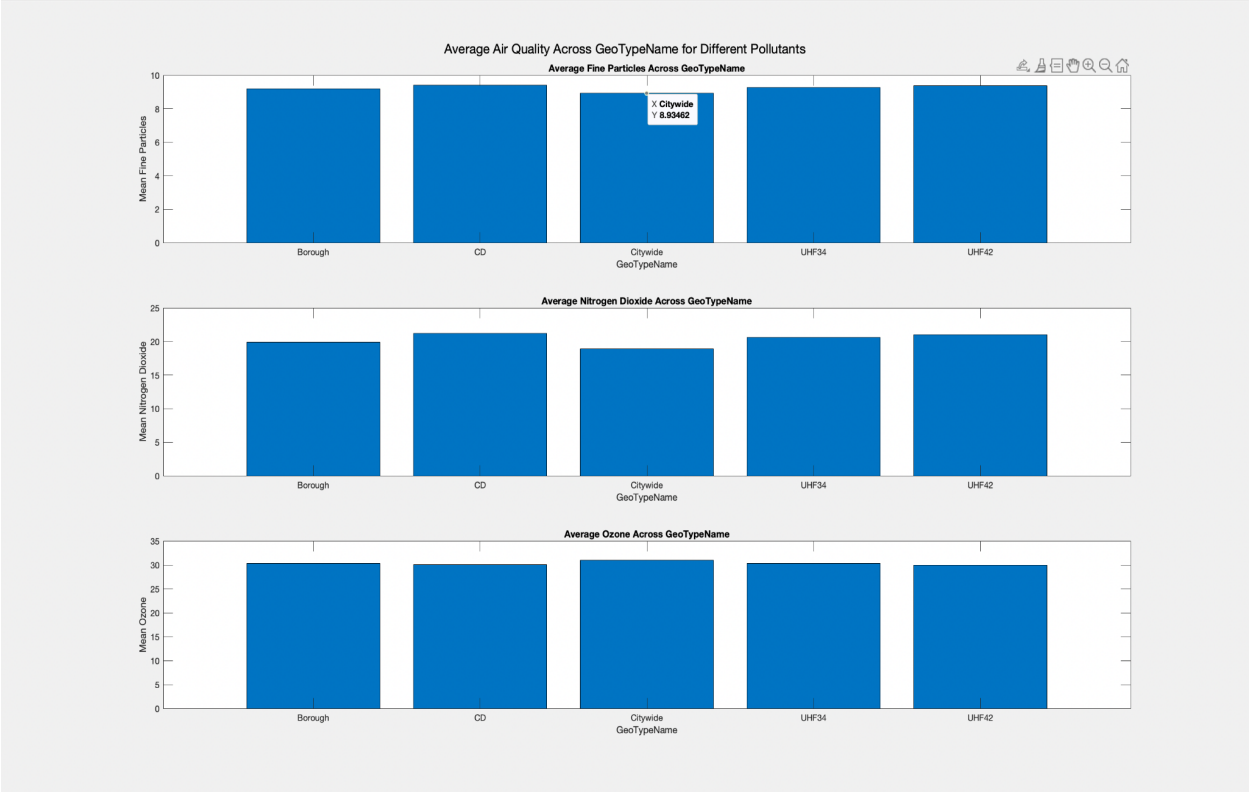
```
%3 subplots for dataset vs time
%For fine particles
subplot(1,3,1)
plot(fine_part.yrs_since_jan1_2005,fine_part.DataValue,'k*')
xlabel('Years since Jan 1, 2005')
ylabel('Fine Particles (mcg/m3)')
title('Fine Particles')
%For nitrogen dioxide
subplot(1,3,2)
plot(nit_dio.yrs_since_jan1_2005,nit_dio.DataValue,'r*')
xlabel('Years since Jan 1, 2005')
ylabel('Nitrogen Dioxide (ppb)')
title('Nitrogen Dioxide')
%For ozone
subplot(1,3,3)
plot(ozone.yrs_since_jan1_2005,ozone.DataValue,'b*')
xlabel('Years since Jan 1, 2005')
ylabel('Ozone (ppb)')
title('Ozone')
%Main heading
sgtitle('Air Quality vs. Time')
```




The graph also clearly shows that the data values for Fine Particles and Nitrogen Dioxide decreases as the years pass. In the case of the ozone dataset, the data values remain almost constant and decrease to a small extent. This was very shocking as we expected the data values to increase as the years pass by.

We also produced three bar graphs under ‘Average air quality across Geo TypeName for different pollutants’. It compares the average air quality across the five types for Fine Particles, Nitrogen Oxide and Ozone.

```
% Bar for Fine Particles
subplot(3, 1, 1);
bar(unique_geo_types,grouped_fine_part.mean_DataValue);
title('Average Fine Particles Across GeoTypeName');
xlabel('GeoTypeName');
ylabel('Mean Fine Particles');

% Bar for Nitrogen Dioxide
subplot(3, 1, 2);
bar(unique_geo_types,grouped_nit_dio.mean_DataValue);
title('Average Nitrogen Dioxide Across GeoTypeName');
xlabel('GeoTypeName');
ylabel('Mean Nitrogen Dioxide');

% Bar for Ozone
subplot(3, 1, 3);
bar(unique_geo_types,grouped_ozone.mean_DataValue);
title('Average Ozone Across GeoTypeName');
xlabel('GeoTypeName');
ylabel('Mean Ozone');
% Main header
sgtitle('Average Air Quality Across GeoTypeName for Different Pollutants')
```



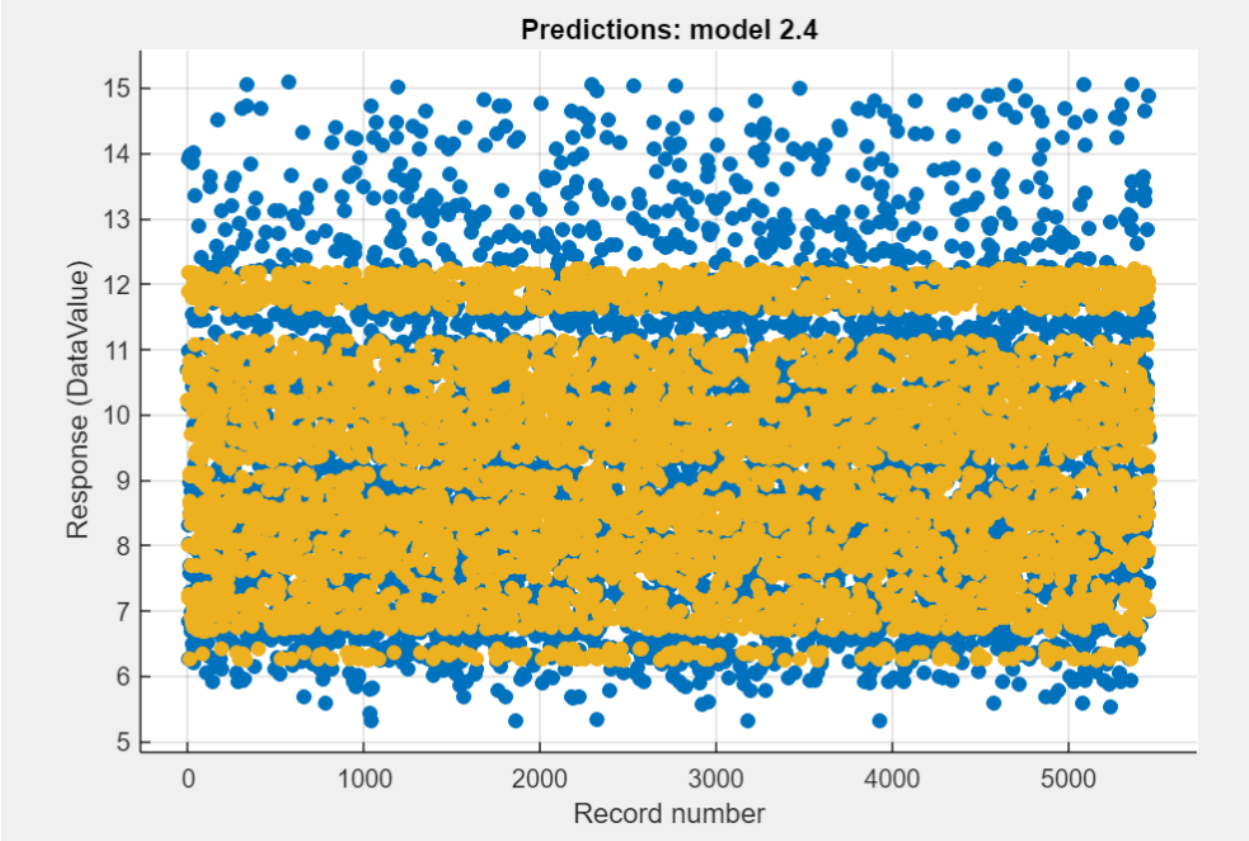
The graph shows that for both Nitrogen Dioxide and Fine Particles, Citywide has the lowest mean value whereas contrastingly it has the highest mean ozone content. Nitrogen Dioxide has the most varied data as shown by the largest gap in between the different bars which shows that nitrogen dioxide concentration is the most differing compared to Fine Particles and ozone. Lastly, ozone stays relatively the same (stable) which makes sense as ozone is pretty much the same concentration throughout the world.

**5) Get your randomized data set as a table variable in the base workspace, and then get into the ML Toolbox. In MATLAB, click on the APPS tab (not the HOME tab). You will see two supervised learning apps, "Classification Learner" and "Regression Learner". If you are trying to classify data, choose the former and if you are trying to predict a real number, choose the latter. Click on New Session, then From Workspace, and then choose your table variable. Some features will be the "predictors" and one will be the "response". You can change that. Start the session. You will see a lot of algorithms listed. If any are grayed out, it is because your data set is not in the correct format for that particular algorithm. Choosing "All Quick to Train" and then clicking on Train will run through all of the available algorithms.**

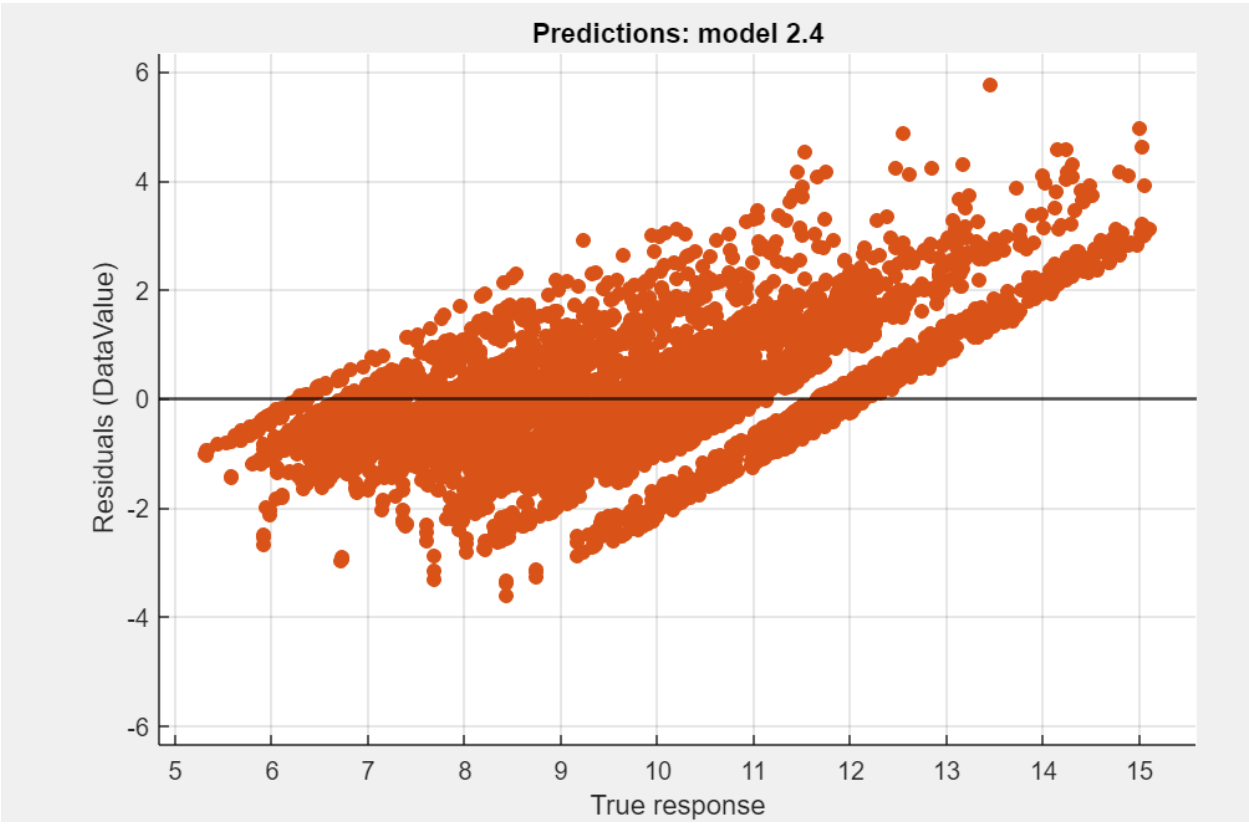
We have three datasets we want to analyze: Fine Particles, Nitrogen Dioxide, and Ozone concentrations. We began by training our data on Fine Particles using the regression learner. The model that had the lowest RMSE was the Coarse Tree model. This means that the Course Tree model was best at predicting the actual values in the dataset and had the lowest root mean squared error. Below we have a summary table of all the models and their RMSE, R squared, MSE, and MAE. The Coarse Tree model (Model 2.4) was better at predicting than the Fine and Medium Tree model. The difference among these three tree models is the number of binary decisions made, with the Coarse Tree making the fewest decisions.

| Model Num... | Model Type | Status | RMSE (Validation) | MSE (Validation) | RSquared (Validation) | MAE (Validation) |
|---|---|---|---|---|---|---|
| 1 | Tree | Trained | 1.14 | 1.2997 | 0.6662 | 0.86159 |
| 2.1 | Linear Regression | Trained | 1.3374 | 1.7887 | 0.5406 | 1.0509 |
| 2.2 | Tree | Trained | 1.14 | 1.2997 | 0.6662 | 0.86159 |
| 2.3 | Tree | Trained | 1.1378 | 1.2946 | 0.6675 | 0.85983 |
| 2.4 | Tree | Trained | 1.1335 | 1.2848 | 0.67003 | 0.85544 |
| 2.5 | Efficient Linear | Trained | 1.3376 | 1.789 | 0.54052 | 1.051 |
| 2.6 | Efficient Linear | Trained | 1.3472 | 1.8148 | 0.5339 | 1.042 |

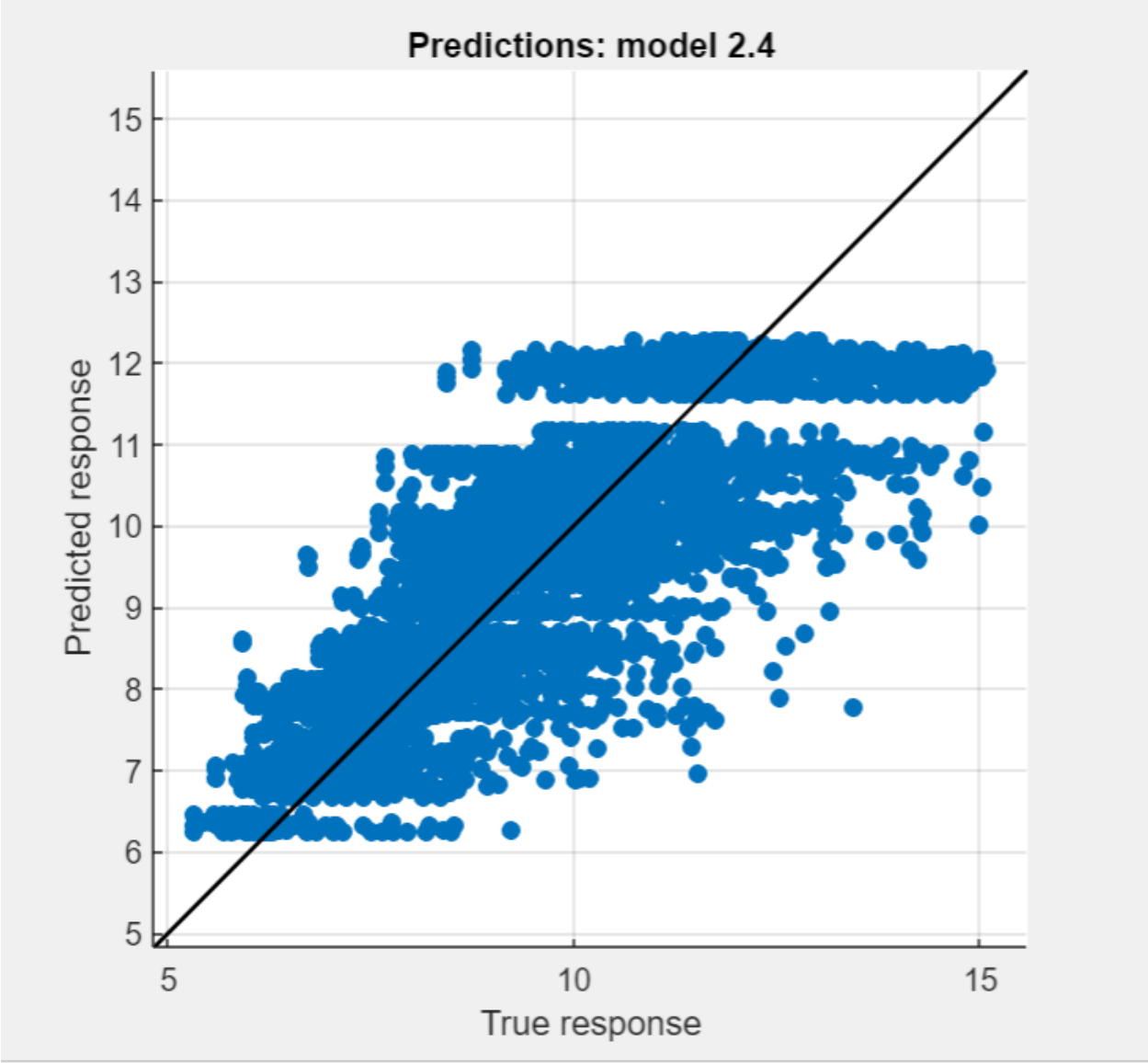
The following is the response plot for the Coarse Tree Model. The blue points are the actual values and the yellow points are the predicted values.




We used the residual plot to see how well the model predicts the data and the errors.

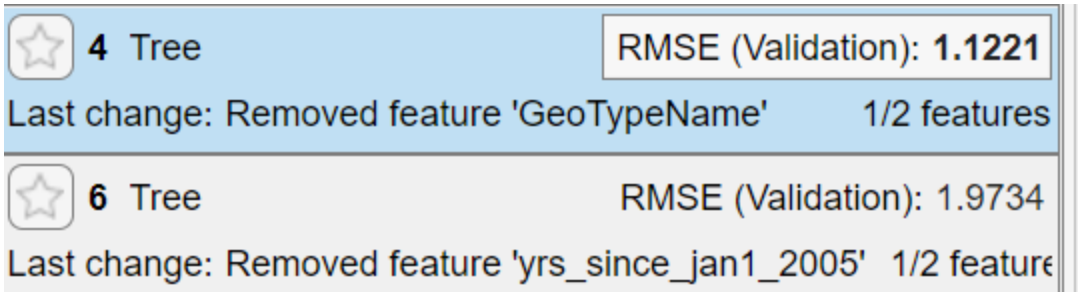


This residual plot does not have an even scatter above and below residual = 0 throughout the plot. To figure out why, we looked at the Predictions vs. Actual plot below.

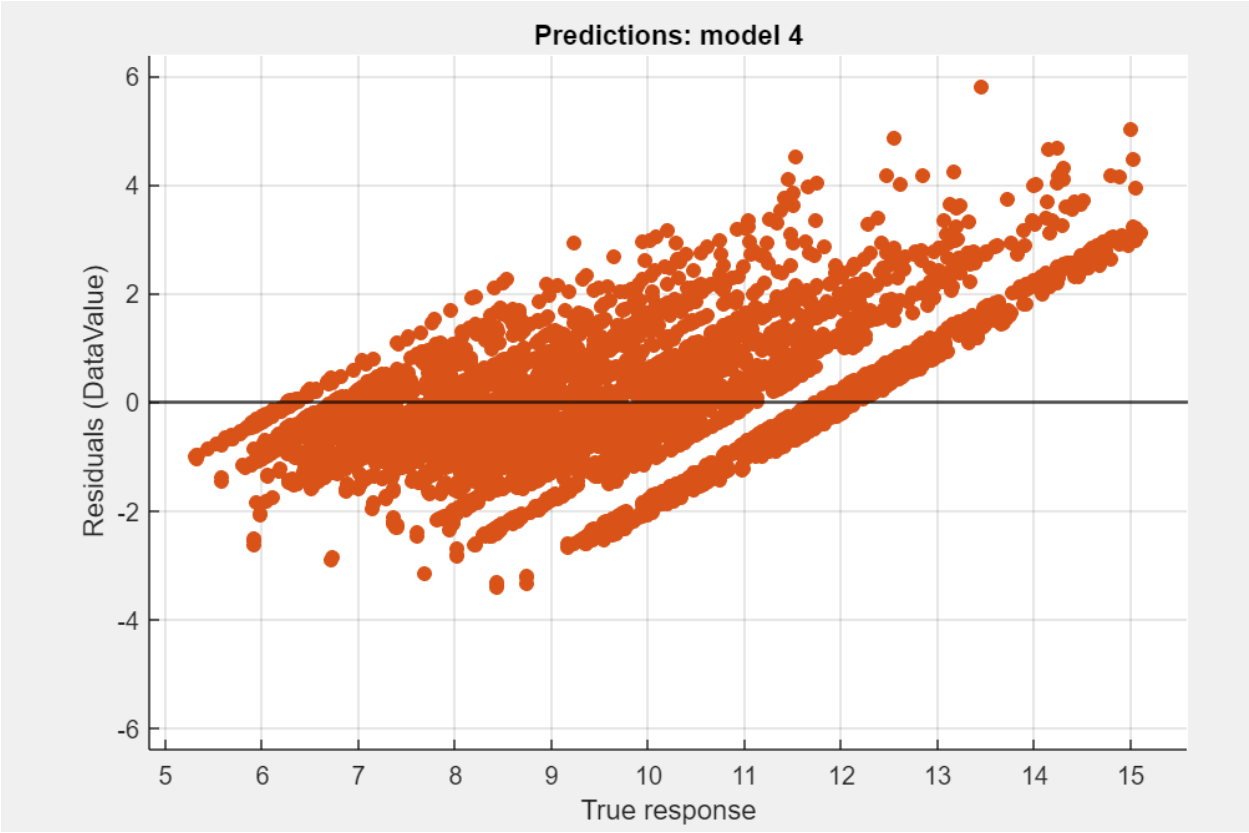



We noticed that there is a definitive minimum and maximum, which we realized was a result of removing our outliers, which caused the skew scene in the residual plot.

To improve the model for Fine Particles, we decided to see if removing one of our features would improve the model. So, we performed another Coarse Tree models, for Fine Particles Concentration vs. Years since Jan. 1, 2005. As seen below, the regression with the Years feature proved to have a lower RMSE value than the original model with both features. This could indicate that the location type does not really have a significant effect on the Fine Particles concentration. While the location type logically seems as it should affect the air quality, it is likely that the classification of geography type used in this dataset is not helpful to predict the air quality.

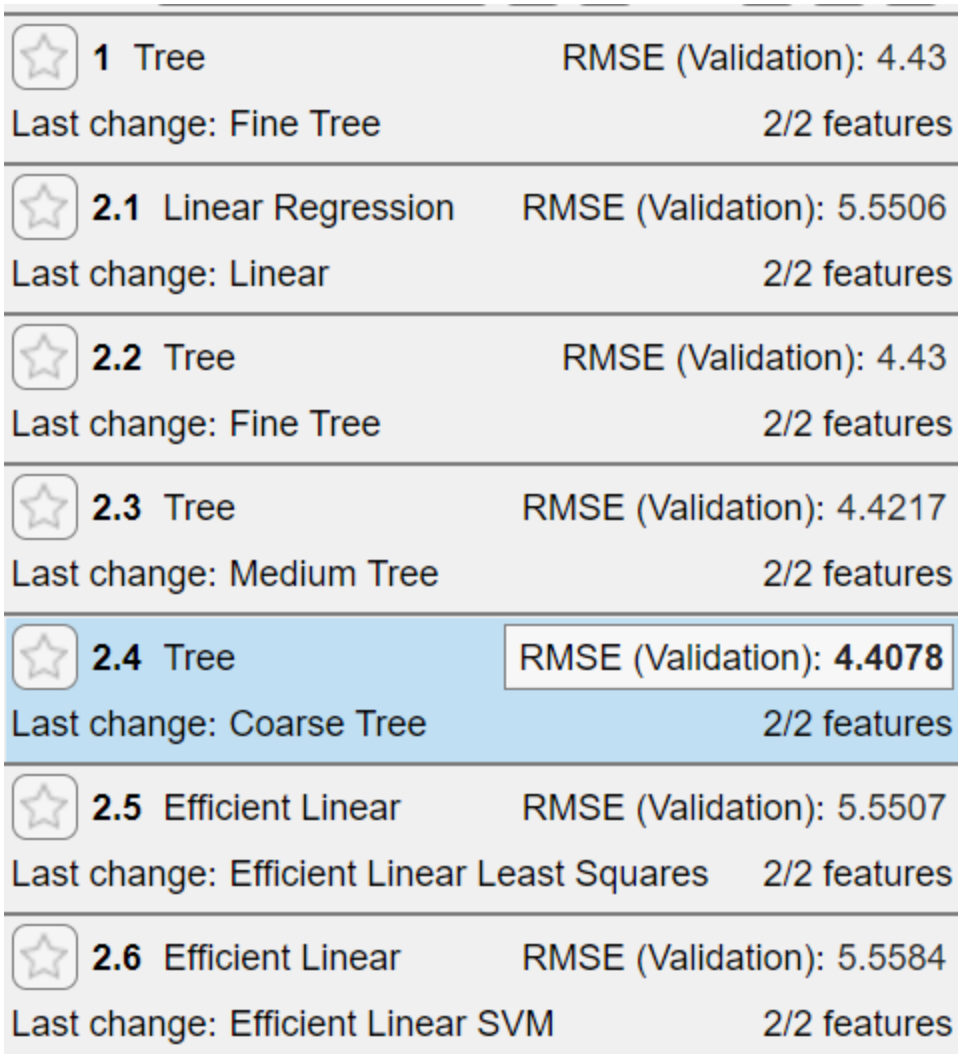



While Model 4 has the lowest RMSE of 1.1221, the residual plot still looks similar to the previous residual plot:

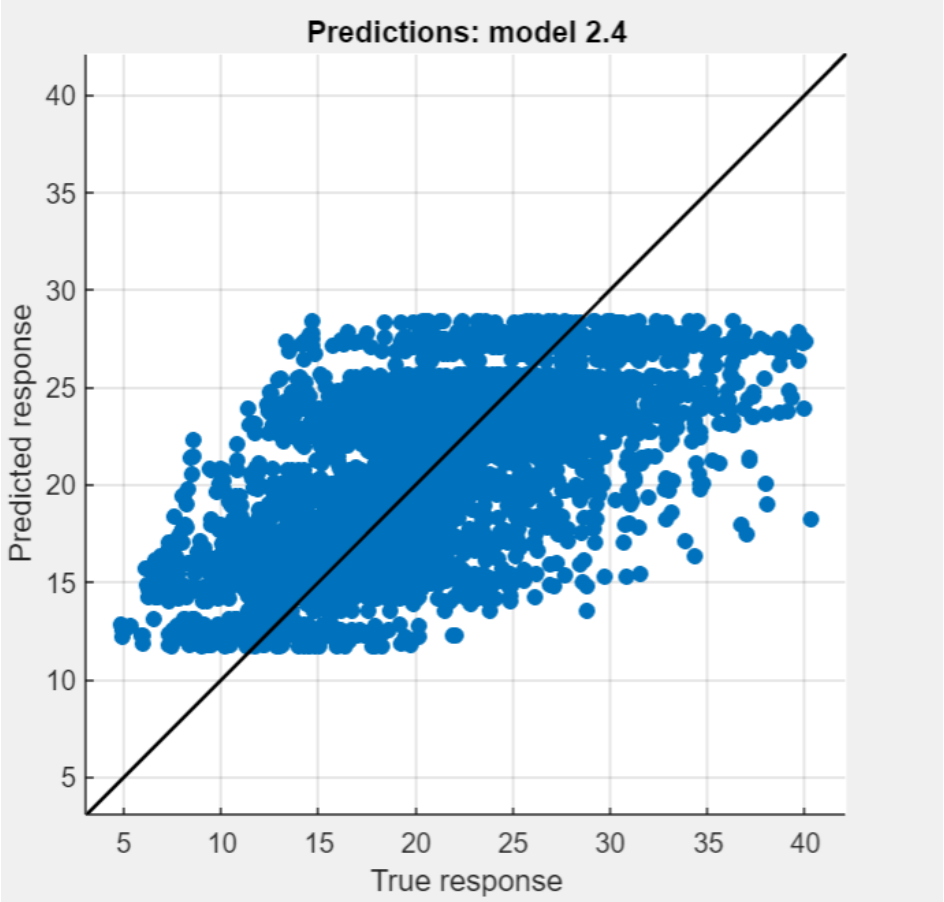


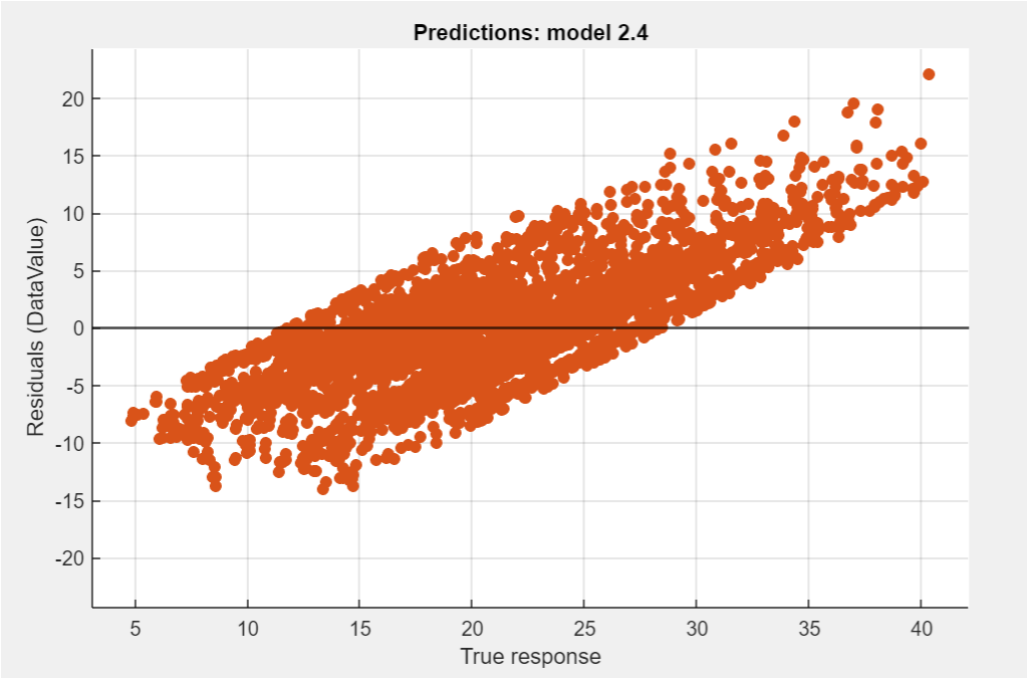

Regardless of the shape of the distribution of residuals, which was the result of removing all the outliers, the middle of the plot has a generally even scatter above and below residual = 0. We decided that the best model for the fine particles dataset was Model 4, in which only one explanatory variable was used, the number of years since January 1, 2005.

Next, we performed analyses on the Nitrogen Dioxide data. Again, the coarse tree model proved to be most effective, with the lowest RMSE.




This makes sense because both nitrogen dioxide and fine particle concentrations decrease with better air quality, so we would expect them to follow similar trends. The following is the Predictions vs. Actual plot for the Nitrogen Dioxide data. The outliers again show a clear cutoff where the data starts and ends. There is also a similar trend in the residual plot as we saw in the residual plot for the fine particle data.



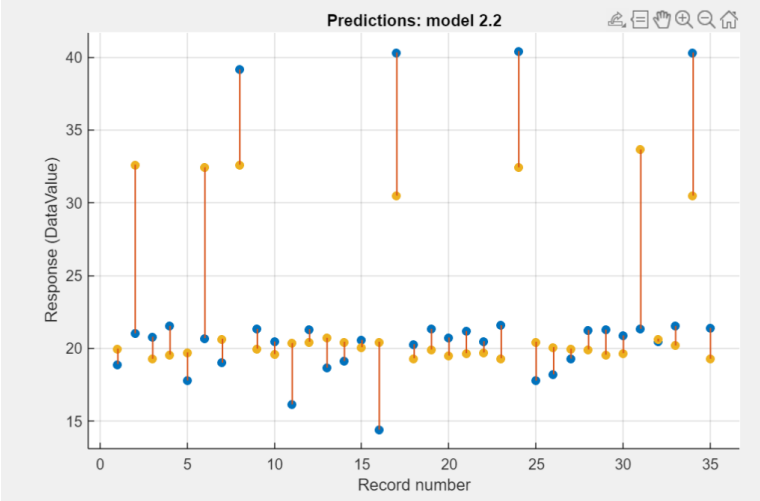




Our conclusions for the Nitrogen Dioxide data are very similar to the Fine Particles data since they have similar relationships to air quality.

Finally, we conducted an analysis on the ozone concentration data. As our plots in question #4 show, the correlation between ozone concentrations and time was very low, so we expected different results from the analyses of Nitrogen Dioxide and fine particles. The model with the lowest RMSE was the fine tree model.

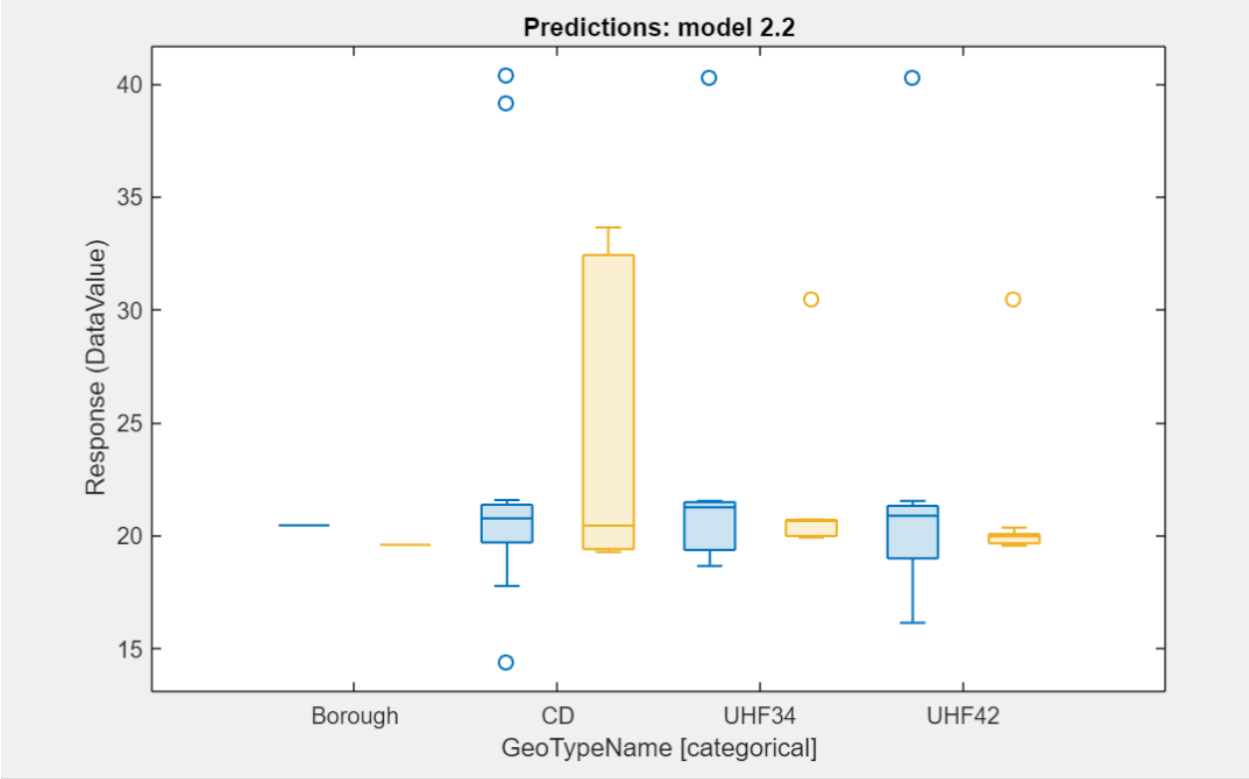
| | | |
|---|---|---|
| **2.1** Linear Regression | RMSE (Validation): 8.1748 | |
| Last change: Linear | 2/2 features | |
| **2.2** Tree | RMSE (Validation): **4.8909** | |
| Last change: Fine Tree | 2/2 features | |
| **2.3** Tree | RMSE (Validation): 6.9083 | |
| Last change: Medium Tree | 2/2 features | |
| **2.4** Tree | RMSE (Validation): 6.7451 | |
| Last change: Coarse Tree | 2/2 features | |
| **2.5** Efficient Linear | RMSE (Validation): 7.9369 | |
| Last change: Efficient Linear Least Squares | 2/2 features | |
| **2.6** Efficient Linear | RMSE (Validation): 9.4592 | |
| Last change: Efficient Linear SVM | 2/2 features | |

The response plot displays the actual values in blue and the predicted values in yellow. The response plot for the fine tree model is below.



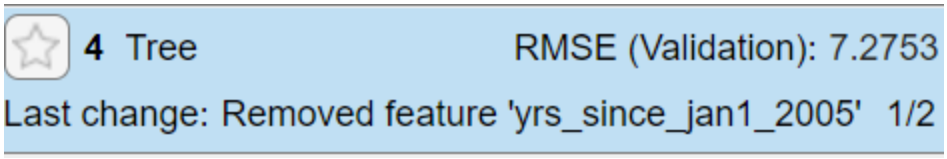
The errors for lower ozone concentrations were very sm. For the few higher ozone concentration data points, while the error is high, it is interesting that the model extrapolated so far relatively well. In the overall dataset, the amount of data on ozone concentrations was very low compared to the amount of data on Fine Particles and Nitrogen Dioxide. So, we expect that if more data was collected, this model could have been even more reliable and accurate.

Since we saw previously that the number of years since January 1, 2005 did not correlate strongly with the ozone concentration, we were curious to see what effect the GeoTypeName had on this model. So, using the same response plot as above, we changed the x-axis of the plot to sort the data based on GeoTypeName. Again, the predictions are yellow and the actual is blue. This plot is shown below.




We noticed that the variability among the different GeoTypeNames is very different for all five categories. Furthermore, the prediction values have a similar variability to the actual values, showing its importance in predicting the air quality. While there's not enough data to make an actual conclusion as to why this is the case, we know that variations in weather result in different ozone formation. The classifications in GeoTypeNames refer to the geographic location where the measurement is taken. (Boroughs refer to five boroughs of New York City: Bronx, Brooklyn, Manhattan, etc. CD, UHF34, and UHF42 are clusters of neighborhoods.)

We tried to train a fine tree model in which we used only the GeoTypeNames as the predictor variable, however this resulted in an RMSE of 7.2753 which was much greater than the previous model, so we decided it was more effective to use the fine tree model with both predictor variables.




Overall, our goal was to see if we could predict the air quality data using a statistic model so we could see what factors had a greater effect on the air quality. This could help politicians and scientists work together to target the issue. In these models, we used only two explanatory variables, however we could explore other factors such as time of day, weather conditions, natural gas emitted, and analyze how they affect air quality. Interestingly enough, for our analysis, we found that nitrogen dioxide and fine particle concentrations both general decreased over time, while ozone particles stayed the same. This is the opposite of what we expected, which could show that some policies restricted greenhouse gas emmissions might be working. If this data does not corroborate with other data though, it could indicate that the measurements or data collection was not properly conducted, which resulted in unexpected trends. Finally, we concluded that the GeoTypeName variable did not have as significant of an effect on the air quality data as we expected. This could mean that this classification of locations is not as effective when predicting air quality data and we could try a different way of classifying locations that might reveal a more interesting trend.

**Your deliverable must also include a statement saying what percentage of the work was done by each individual team member.**
Nishtha Ladi: 33.3%
Varsha Athreya: 33.3%
Rishi Kappor: 33.3%